

**DATABASE MANAGEMENT AND
BIOLOGICAL DATA BANKS-
MOLECULAR DESIGNING
(DIB03)
(PG DIPLOMA)**



ACHARYA NAGARJUNA UNIVERSITY

CENTRE FOR DISTANCE EDUCATION

NAGARJUNA NAGAR,

GUNTUR

ANDHRA PRADESH

Lesson 3.1.1**TOOLS IN BIOINFORMATICS****Objective****3.1.1.1 Biological Research on the Web****3.1.1.2. Sequence analysis tools****3.1.1.3 Web-Based Protein Structure Tools****Summary****Model Questions****References****Objective**

Sequence analysis refers to extraction of information from nucleic acid or protein primary sequences. This sequence information determination generally is of two kinds, degree of similarity and homology of the nucleic acid/ protein to other nucleic acid/ proteins. On obtaining this information it becomes easier to predict likely structure or function in the protein or coding regions in DNA.

The primary sequence of a protein is generally a translation of a DNA sequence. This means that if DNA sequences evolve, so do proteins, this infers protein sequences from closely related organisms are very similar and the comparison of sequences of the same protein from different organisms gives evolutionary relationships. Proteins that have evolved from a common ancestor are said to be homologous and intern protein homologies can be used to determine evolutionary relationships. In order to detect and quantify homology it is necessary to align and compare sequences. This is a huge task when you have to compare with several thousands of sequences. New sequences are usually compared against sequence databanks using powerful computers and tools. If two sequences show a high degree of similarity or identify then there is a good chance that they are homologous, conversely, a low degree of identity state homologous sequences widely separated in evolution can look very dissimilar. It can be said that unrelated sequence shows about 6% random identity and for long protein sequences 20% identity is usually taken to indicate homology. When homologous sequences are compared, if it shows some regions that are conserved this is an indicative of essential common function for those regions-in case of protein sequences active sites in enzymes are often located in this way.

For example, there are several approaches to the comparison of amino acids, though they generally fall into two categories:

1. Comparison on the basis of physical characteristics such as polarity, charge, vander waals volume etc. to generate a similarity matrix.
2. Comparison on the basis of likely replacement of an amino acid in a sequence by any other amino acid. This is based upon the observation of many protein sequences from homologous proteins and generates a Point Accepted Mutation Matrix (PAM).

Few questions to be asked while doing database search are –

What sequence (nucleotide or protein) has to be used to search for similarity ?

It we have a nucleotide sequence, should we have to search the DNA databases only ?

Or should we translate this into protein and search protein databases ?

The answers to above queries are –

When comparing DNA sequences, we get significantly more random matches than we get with protein sequences. Because of the fact that the DNA databases are much larger, and are growing at a faster pace when compared to protein databases. Bigger database means more random hits. For DNA we usually use identity matrices, for protein more sensitive matrices like PAM and BLOSUM, which allow for better search results. Proteins are conserved evolutionarily as they are rarely mutated. By translating DNA into amino acid sequence, we'll presumably lose information, because two or more codons can be translated to the same amino acid. As far as possible we should use proteins for database similarity searches.

3.1.1.1 Biological Research on the Web

The Internet has completely changed the way scientists search for and exchange information. Data that once had to be communicated on paper is now digitized and distributed from centralized databases. Journals are now published online. And nearly every research group has a web page offering everything from reprints to software downloads to data to automated data-processing services.

A simple web search for the word *bioinformatics* yields tens of thousands of results. The information you want may be number 345 in the list or it may not be found at all. Where can you go to find only the useful software and data, and scientific articles? You won't always get there by a simple web search. How can you judge which information is useful? Publication on the Web gives information an appearance of authority it may not merit. How can you judge if software will give the type of results you need and perform its function correctly?

In this session we examine the art of finding information on the Web. We cover search engines and searching, where to find scientific articles and software, and how to use the classic online information sources such as PubMed.

Using Search Engines

Scirus, Google, AltaVista, Lycos, HotBot, Northern Light, Dogpile, and dozens of other search engines exist to help you find your way around the billion or more pages that make up the Web. As a scientist, however, you're not looking for common web commodities such as places to order books on the Web or online news or porn sites. You're looking for perhaps a couple of needles in a large haystack.

Knowing how to structure a query to weed out the majority of the junk that will come up in a search is very useful, both in web searching and in keyword-based database searching.

Boolean Searching

Most web surfers approach searching haphazardly at best. Enter a few keywords into the little box, and look at whatever results come up. But each search engine makes different default assumptions, so if you enter *protein structure* into Excite's query field, you are asking for an entirely different search than if you enter *protein structure* into Google's query field. In order to search effectively, you need to use boolean logic, which is an extremely simple way of stating how a group of things should be divided or combined into sets.

Search engines all use some form of boolean logic, as do the query forms for most of the public biological databases. Boolean queries restrict the results that are returned

from a database by joining a series of search terms with the operators AND, OR, and NOT. The meaning of these operators is straightforward: joining two keywords with AND finds documents that contain only *keyword1* and *keyword2*; using OR finds documents that contain *either keyword1* or *keyword2* (or both); and using NOT finds documents that contain *keyword1* but not *keyword2*.

However, search engines differ in how they interpret a space or an implied operator. Some search engines consider a space an OR, so when you type *protein structure*, you're really asking for protein or structure. If you search for *protein structure* on Excite, which defaults to OR, you come up with a lot of advertisements for fad diets and protein supplements before you ever get to the scientific sites you're interested in. On the other hand, Google defaults to AND, so you'll find only references that contain protein and structure, which is probably what you intended to look for in the first place. Find out how the search engine you're using works before you formulate your query.

Boolean queries are read from left to right, just like text. Parentheses can structure more complex boolean queries. For instance, if you look for documents that contain *keyword1* and one of either *keyword2* or *keyword3*, but not *keyword4*, your query would look like this: (*keyword1* AND (*keyword2* OR *keyword3*)) NOT *keyword4*.

Finding Scientific Articles

Scientists have traditionally been able to trust the quality of papers in print journals because these journals are refereed. An editor sends each paper to a group of experts who are qualified to judge the quality of the research described. These reviewers comment on the manuscript, often requiring additions, corrections, and even further experiments before the paper is accepted for publication. Print journals in the sciences are, increasingly frequently, publishing their content in an electronic format in addition to hardcopy. Almost every major journal has a web site, most of which are accessible only to subscribers, although access to abstracts usually is free. Scientific articles in these web journals go through the same process of review as their print counterparts.

Another trend is e-journals, which have no print counterpart. These journals are usually refereed, and it shouldn't be too hard to find out by whom. For instance, the *journal of Molecular Modeling*, an electronic journal published by SpringerVerlag, has links to information about the journal's editorial policy prominently displayed on its home page.

An excellent resource for searching the scientific literature in the biological sciences is the free server sponsored by the National Center for Biotechnology Information (NCBI) at the National Library of Medicine. This server makes it possible for anyone with a web browser to search the Medline database. There are other literature databases of comparable quality available, but most of these are not free. Your institution may offer access to such sources as Lexis-Nexis or Cambridge Scientific Abstracts.

Outside of refereed resources, however, anyone can publish information on the Web. Often research groups make papers available as technical reports on their web sites. These technical reports may never be peer reviewed or published outside the research group's home organization, and your only clue to their quality is the reputation and

expertise of the authors. This isn't to say that you shouldn't trust or seek out these sources. Many government organizations and academic research groups have reference material of near-textbook quality on their web sites. For example, the University of Washington Genome Center has an excellent tutorial on genome sequencing, and NCBI has a good practical tutorial on use of the BLAST sequence alignment program and its variants.

Using PubMed Effectively

PubMed (<http://www.ncbi.nlm.nih.gov/entrez/query.fcgi>) is one of the most valuable web resources available to biologists. Over 4,000 journals are indexed in PubMed, including most of the well-regarded journals in cell and molecular biology, biochemistry, genetics, and related fields, as well as many clinical publications of interest to medical professionals.

PubMed uses a keyword-based search strategy and allows the boolean operators AND, OR, and NOT in query statements. Users can specify which database fields to check for each search term by following the search term with a field name enclosed in square brackets.

3.1.1.2. Sequence analysis tools

We now begin our tour of bioinformatics tools in earnest. In the next five chapters, we describe some of the software tools and applications you can expect to see in current research in computational biology. From gene sequences to the proteins they encode to the complicated biological networks they are involved in, computational methods are available to help you analyze data and formulate hypotheses. We have focused on commonly used software packages and packages we have used; to attempt to encompass every detail of every program out there, however, we'd need to turn every chapter in this book into a book of its own.

The first tools we describe are those that analyze protein and DNA sequence data. Sequence data is the most abundant type of biological data available electronically. While other databases may eventually rival them in size, the importance of sequence databases to biology remains central. Pairwise sequence comparison, which we discuss in this chapter, is the most essential technique in computational biology. It allows you to do everything from sequence-based database searching, to building evolutionary trees and identifying characteristic features of protein families, to creating homology models. But it's also the key to larger projects, limited only by your imagination—comparing genomes, exploring the sequence determinants of protein structure, connecting expression data to genomic information, and much more.

The types of analysis that you can do with sequence data are:

- Knowledge-based single sequence analysis for sequence characteristics.
- Pairwise sequence comparison and sequence-based searching
- Multiple sequence alignment
- Sequence motif discovery in multiple alignments
- Phylogenetic inference

Sequence databases are usually accessed over the Internet, although some can be obtained on CD-ROM or downloaded from internet. The most important databases are discussed in biological database chapter. The most common type of database search is a similarity search. There are many approaches to similarity searching, the two most widely used are:

- Blast-Basic Local Alignment Search Tool. This identifies sequences that have common blocks of local similarity.
- Fast A – This searches for sequences that display global similarity.

The most common approach is to carry out a Fast A search of a database then to use BLAST to locate and quantify blocks of local similarity. A basic understanding of sequence alignments is necessary to comprehend BLAST or any other sequence similarity search tool. Sequence alignments are used to find potential homologues, which are then used to predict function of the query sequence or in modeling its 3D structure.

The sequence alignment tools are classified as global and local alignment tools.

- The best overall alignment over the entire length of the specified is a global alignment. Entire sequence length alignment is obtained by introduction of gaps in two sequences.

- The optimal alignment between local regions in a specified sequence is a local alignment. The main advantage of global alignment is its sequence optimization, which share high degree of similarity. This is very useful in 3D structure prediction based on sequence similarity. And the local alignment is best suited to find motifs, domains, and other conserved regions within the sequences. Local alignment can be used for shorter regional segments that show high degree of similarity.

BLAST

BLAST (Basic Local Alignment Search Tool) is a popular user-friendly tool for searching all the major sequence databases. BLAST is a heuristic method to find the highest scoring locally optimal alignments between a query sequence and a database sequence. Blast programs were designed for fast database searching, with minimal sacrifice of sensitivity to distant related sequences. BLAST is used to find sequence homologs to predict the identity, function, 3D structure of the query sequence. BLAST shows better results for protein sequences than nucleotide sequences. The default database is the nr(non-redundant) database, and the user still has the option to select any database of their choice from the list. BLAST is based on statistical theory developed by Samuel Karlin and Steven Altschul (PNAS 87: 2284-2268. 1990). This original theory was later extended to cover multiple weak matches between query and databases entry. A group at the National Center Biotechnology Information (NCBI), USA supports the BLAST server.

Some of the solient features of BLAST are

Local alignments : BLAST tries to find patches of regional similarity rather than trying for global fit between the query and the database sequence. But multiple hits to the same sequence is allowed.

Ungapped alignments: BLAST programs work on statistics of ungapped sequence alignments, but theoretically this reduces sensitivity of search. However, output shows multiple local alignments between query and a database sequence that can be used to anticipate the gaps between them. Only identities and conservative replacements are taken into account.

The use of filters reduces problems of contamination with numerous artifacts in the databases. By using filters we can find out the excluded true positive hits from the initial run. For this we have to first search selecting the filter to reduce false positive hits, while other search is done without selecting filter to maximize sensitivity. The outputs from these two are contrasted to find the possible true positive hits that were excluded from the initial filter search. The statistical theory only covers the likelihood of finding a match by chance under particular assumptions; it does not guarantee any biological importance.

BLAST is extremely fast, the program can be run locally or queries can be E-mailed to the NCBI server. BLAST does not guarantee to find the best alignment between your query and the database; it may miss matches.

Because its strategy is expected to find most matches, and this way it sacrifices complete sensitivity in order to gain speed (Hewistic). In practice few biologically significant matches that are missed by BLAST can be found out by using other sequence search programs.

Blast searches the database in two phases. First it looks for short subsequences that are likely to have significant matches, and then it tries to extend these matched regions (subsequences) on both sides in order to obtain maximum sequence similarity. A substitution matrix is used during all phases of sequence searches. A substitution matrix is a scoring method used in the alignment of one residue (nucleotide / amino acid) against other Margaret Dayhoff and her co-workers developed the first substitution matrix used in the comparison of protein sequences for evolutionary terms. These matrices were derived from global alignments of closely related sequences. These matrices were extrapolated for less similar or evolutionary more distant sequences. These matrices are commonly called as Dayhoff, MDM or PAM matrices. The number (PAM40, PAM100) accompanying PAM matrices refers to the evolutionary distance between the sequences. Larger number represent greater evolutionary distant and smaller number signifies evolutionarily less distant sequences. In contrast to PAM matrices that are based on the concept of global alignments of closely related sequences, Steve Henikoff and his co-workers developed BLOSUM matrices and these are based on local alignments of distantly related sequences. In BLOSUM series the small number (BLOSUM 62, BLOSUM 80) indicates minimum percent identity i.e. Smaller numbers correspond to greater evolutionary distant sequences. PAM series can be suited better for global similarity searches, while BLOSUM series perform better in finding local similarity regions, depending on the usage the respective matrix could be selected for our sequence analysis. Substitution matrices greatly improve sensitivity.

The BLAST algorithm works in the following steps –

Preprocessing of the query

The first step is to quickly locate ungapped similarity regions between the query sequence and sequences from the database. Similarly, all the words of length w (tuples or words) of the query are compared with those of all the database sequences. Blast uses a more efficient manner to search the database, all the words of length w formed with the alphabet of the sequences are generated (for example, with amino acid sequences, if $w = 2$ there are $20^2 = 400$ possible words, and $20^3 = 8000$ if $w = 3$) and each word of the query is compared with each word of this exhaustive set and a threshold T for the similarity between words is set. Each position of the query sequence is associated with a list of words that score more than T when compared with the word of the query starting at this position. The similar words are also called neighbours.

Blastp : This program compares an amino acid query sequence against a protein sequence database.

Blastn : This program compares a nucleotide query sequence against a nucleotide sequence database. A newly sequenced can be queried against the database to identify the sequence or to find out the potential contamination of the query sequence.

Blastn : This program searches the six-frame translation products of a nucleotide sequence (both strands) against a protein database. Initially the nucleotide query sequence is translated into all of its six possible reading frames. This program is useful in finding errors in nucleotide sequencing by comparing the translated nucleotide query sequence to its best homologs in a protein sequence database. From the blastn output we can also identify unclear nucleotides.

Tblastn : This program searches a protein sequence against translated nucleotide sequences (all 6 reading frames) in the database. Initially the nucleotide sequence in the database are translated into each of its six possible reading frames and then compared with the protein query sequence to find the best nucleotide homolog. The output of tblastn can also help to clarify unclear amino acid residues in the query.

Tblastx : Compares the six-frame translations of a nucleotide query sequence against the six-frame translations of a nucleotide sequence database. This program is similar to blastx and tblastn program.

PSI - BLAST

Position specific Iterated-BLAST, this tool can be used when your BLAST searches results give you very few matches, the PSI-BLAST will re-iterate the BLAST searches creating a defined profile, upon re-iteration (you just click on the button to re-iterate) you may reveal alignment matches that are significant that you would not have found using BLAST alone. PSI-BLAST generates “on-the-fly” a scoring matrix specific to your BLAST search, and continues to specify this matrix upon each re-iteration.

PHI-BLAST

Pattern Hit Initiated-BLAST, this tool can be used to search for a specific pattern or motif in your sequence and in the databases the pattern designates the amino acid sequence you are searching for e.g. (RG) – (M) – (X) – (YWF) – 5(X) – (A); this submission pattern would yield a search for sequence patterns having “R” (Arginine) or “G” (glycine) at position 1 (not necessarily position 1 or the N-terminus of the amino acid sequences in the databases), followed by a “M” (Methionine), followed by any amino acid “X”, followed by any one of three AA: “Y” (Tyrosine) or “W” (Tryptophan) or “F” (Phenylalanine); followed by any 5 amino acids “X”, followed by an “A” (Alanine).

BLAST 2 sequences

This tool produces the alignment of two given sequences using BLAST engine for local alignment.

GENERAL GUIDELINES for analysis CUT-OVVS of BLAST OUTPUT

E – value <0.05

% Identity >25% over 100 base pairs or 35 amino acid residues.

BLAST 2.0 is a new version with new capabilities such as gapped-Blast and Psi-Blast. New versions allow gaps which make them much more powerful eg. GAPBLAST. Even more powerful versions build a sort of consensus sequence for initial matches and then use that for further searching eg. PISIBLAST. This new BLAST 2.0 server has been

redesigned to optimize speed and sensitivity. The filtering capabilities enable to find sequences with low complexity regions. GAPPED BLAST algorithm allows gaps for alignment runs three fold faster to original blast. That means that introductions of gaps to these sequences prevent the similar regions not to be broken into several less meaningful sequence fragments (as in the older versions). This method reflects biological relationships much better. Gaps are either deletions or insertions introduce into the sequence. This way the similar sequence regions are prevented from breaking down into segments. The conserved regions reflect active sites, binding sites. Although gapped extension is computationally intensive but highly sensitive.

PSI – BLAST (Position-Specific Iterated BLAST) provides a new automated “profile-like” search, available for protein-protein search, here the database is searched using a profile as a query rather than a simple sequence. The program first performs gapped BLAST search and the information of the significant alignments are then used by the program to construct a position specific score matrix. This matrix replaces the original query sequence and is used to find profiles in the next iterative database search runs. The program may be iterated until no new significant alignments are found. There is no guarantee that the group discovered is authentically indicates a functional group but we can deduce a relation between them.

Fast A

Fast A is another sequence analysis tool very much similar to BLAST, this was originally developed by W.R. Pearson & Hipman and this algorithm can be accessed from EBI site. Fast A gives better results for nucleotide sequences than protein sequences. The Fast A programs searches the database files to find a number of related sequences to the query sequence and displays a pairwise alignment between them.

Some salient features of Fast A

Fast A is used for nucleic acids and Fast P for proteins. Find regions of similarity by first breaking the sequence into short subsequences (“words or Ktups”) then searching for diagonals (picture a dot matrix alignment) with highest density of words that match.

The alignment in these diagonals is then rdefined.

Alignments near enough to each other are joined to create a longer alignment. Gaps can be allowed.

The program is quite fast but is not guaranteed to find the best alignment between your query and the database; it may miss matches. This is because it was a strategy, which is expected to find most matches, but sacrifices complete sensitivity in order to gain speed.

Fast A can be better than BLASTN for nucleic acid comparisons, but usually no better for proteins. So BLAST P is preferable because it is much faster.

The following are the programmes available in FAST A.

- * FASTA 3 : Scans a protein or DNA sequence library for similar sequences.
- * FAST x/y³ : Compares a translated DNA sequence in forward and reverse frame against a protein sequence database.
- * tFAST x/y³ ; Compares a query protein sequence to a translated DNA data bank.
- * FASTS3 : Compares linked peptides to a protein databank.
- * FAST f3 : Compares mixed peptides to a protein databank.

Parameters and terms used in Fast A

Ktup determines how many consecutive identities should be present for a word to match (word length). If the word length is 2, then the programme searches only those regions in the database sequence that have at least 2 adjacent identical residues. For DNA searches a ktup of 6 is the default and for protein searches the default is 2. The thumb rule is that the larger the word length the less sensitive, but faster will be the search. Word length is 1 is used for more sensitive searches. Here more number of comparisons would be required, hence more time is taken.

MATRIX : The default matrix for fast a is Blosum 62 you may choose any matrix from the available list (Eg. BLUSUM, PAM etc.), which covers various evolutionary constraints.

Gap Penalty : Gap open penalty for the first residue in the gap (-12 by default for proteins, -16 for DNA), gap extension penalty for additional residues in a gap (-2 by default for proteins, -4 for DNA).

Histogram: displays the search histogram of the expected frequency of chance occurrence of the database matches found.

E-VALUE : E(Expectation)-Values : is used for the evaluation of statistical significance. The E-value for a given alignment depends upon its score as well as the lengths of both the query sequence and the database searched sequences. Smaller E-value indicates more statistical significance of the match. The upper E-value limit for score and alignment display by defaults are 10.0 for FASTA with protein searches, 5.0 for translated DNA/protein comparisons, and 2.0 for DNA/DNA searches. In the lower E-value limit a value of 1×10^{-6} from being displayed. This allows the use of to focus on more distant relationships.

Scores

Init 1 score : The score for a pair of matching regions of the query and the database sequence that has highest degree of word matches.

Init n score : If any of the initial regions from difference. Non overlapping diagonals can be joined together to form an approximate alignment with gaps. This score for the joined regions is the score of the initial regions.

Opt score : The score for the alignment of the best segment of similarity between the query sequence and database sequence using variation of smith waterman algorithm.

Z-score : Evaluates the significance of the "opt score" by generating a score distribution from the alignment of many random pairs of sequences having the same length as the 2 compared sequences. From this distribution, the number of standard deviations from the mean for the alignment score of interest is calculated as the z-score. Better the match, higher the z-score.

Multiple Sequence Alignment

Multiple sequence alignment techniques are most commonly applied to protein sequences; ideally they are a statement of both evolutionary and structural similarity among the proteins encoded by each sequence in the alignment. We know that proteins with closely related functions are similar in both sequence and structure from organism to organism, and that sequence tends to change more rapidly than structure in the course of evolution. In multiple alignments generated from sequence data alone, regions that are similar in sequence are usually found to be superimposable in structure as well.

With a detailed knowledge of the biochemistry of a protein, you can create a multiple alignment by hand. This is a painstaking process, however. The challenge of automatic alignment is that it is hard to define exactly what an optimal multiple alignment is, and impossible to set a standard for a single correct multiple alignment. In theory, there is one underlying evolutionary process and one evolutionarily correct alignment to be generated from any group of sequences. However, the differences between sequences can be so great in parts of an alignment that there isn't an apparent, unique solution to be found by an alignment algorithm. Those same divergent regions are often structurally unalignable as well. Most of the insight that we derive from multiple alignments comes from analyzing the regions of similarity, not from attempting to align the very diverged regions.

The dynamic programming algorithm used for pairwise sequence alignment can theoretically be extended to any number of sequences. However, the time and memory requirements of this algorithm increase exponentially with the number of sequences. Dynamic programming alignment of two sequences takes seconds. Alignment of four relatively short sequences takes a few hours. Beyond that, it becomes impractical to align sequences this way. The program MSA is an implementation of an algorithm that reduces the complexity of the dynamic programming problem for multiple sequences to some extent. It can align about seven relatively short (200-300) protein sequences in a reasonable amount of time. However, MSA is of little use when comparing large numbers of sequences.

Multiple Alignment with ClustalW

One commonly used program for progressive multiple sequence alignment is ClustalW. The heuristic used in ClustalW is based on phylogenetic analysis. First, a pairwise distance matrix for all the sequences to be aligned is generated, and a guide tree is created using the neighbor-joining algorithm. Then, each of the most closely related pairs of sequences—the outermost branches of the tree—are aligned to each other using dynamic programming. Next, each new alignment is analyzed to build a sequence profile. Finally, alignment profiles are aligned to each other or to other sequences (depending on the topology of the tree) until a full alignment is built.

This strategy produces reasonable alignments under a range of conditions. It's not foolproof; for distantly related sequences, it can build on the inaccuracies of pairwise alignment and phylogenetic analysis. But for sequence sets with some recognizably related pairs, it builds on the strengths of these methods. Pairwise sequence alignment by dynamic programming is very accurate for closely related sequences regardless of which scoring matrix or penalty values are used. Phylogenetic analysis is relatively unambiguous for closely related sequences. Using multiple sequences to create profiles increases the accuracy of pairwise alignment for more distantly related sequences.

3.1.1.3 Web-Based Protein Structure Tools

Now that we've reviewed the basics of protein chemistry, let's turn our attention to the tools. The *most* important *source* of information about protein structure is the POB. In addition to being an entry point to the structural data itself, the POB website (<http://www.rcsb.org/pdb>) contains links to many tools database you can apply to individual protein structures as you search the database. Information from the

database is made available through the Protein Structure Explorer interface. For each protein, you can view the molecular structure using 3D display tools such as RasMol and the Java QuickPDB viewer. PDB files and file headers can be viewed as HTML and downloaded in a variety of formats. Links to the protein structure classification databases CATH, FSSP, and SCOP are provided, along with the tools CE and VAST, which search for structures based on structural alignment. Average geometric properties, including dihedral angles, bond angles, and bond lengths can be displayed in tabular format with extremes and deviations noted. Sequences can be viewed and labeled according to secondary structure, and sequence information downloaded in FASTA format.

You can go directly to the page for a particular protein of interest by entering that protein's four-letter PDB code in the Explore box on the PDB's main page. The PDB can also be searched using two different search tools, SearchLite and SearchFields. SearchLite is a simple search tool that allows you to enter one or more search terms separated by boolean operators into a single search field. SearchFields is a tool for advanced searches that provides a customizable search form that allows you to use separate keywords to search each PDB header field. You can modify the form by selecting checkboxes at the bottom of the form and regenerating the form. SearchFields supports options for searching a dozen of the most important fields in the PDB header, as well as crystallographic information. SearchFields also allows the database to be searched using FASTA for sequence comparison, as well as secondary structure features or short sequence features.

From the individual protein page generated by the Structure Explorer, the PDB provides a menu of links through which to connect to other tools. These features are still evolving rapidly. Table 9-2' provides a brief overview of the PDB protein page. We also encourage you to explore the PDB site regularly if you are interested in tools for protein structure analysis.

Structure Visualization

One of the first tools developed for structure analysis and one of the first analyses you will probably want to do is simply structure visualization. Protein structure data is stored as collections of x , y , z coordinates, but proteins can't be visualized simply by plotting those points. The connectivity between atoms in proteins has to be taken into account, and for the visualization to be effective, a virtual 3D environment, which provides the illusion of depth, needs to be created. Fortunately, all this was worked out in the 1970s and 1980s, and there are now a variety of free and commercial structure visualization tools available for every operating system.

Even with virtual 3D representation, protein structures are so complex that they are difficult to interpret visually. The human eye can interpret 3D solids, but has a difficult time with topologically complex 3D data sets. There are a number of conventional simplified representations of protein structure that allow you to see the overall topology of the protein without the confusion of atomic detail. In order to be useful, a protein structure visualization program needs to, at minimum, be able to display user-selected subsets of atoms with correct connectivity, draw standard cartoon representations of proteins such as ribbons and cylinders, and recolor subsets of a molecule according to

a specified parameter.

Molecular Structure Viewers

One type of molecular structure viewers are lightweight applications that can be set up to work with your web browser. When properly configured, they will display molecular data as you access it on the Web. RasMol and CnD3 are two of the most popular viewers.

RasMol

One of the most popular molecular structure visualization program tools is RasMol. It is available for a wide range of operating systems, and it reads molecular structure files in the standard PDB format. RasMol 2.7.1, the most up-to-date version, can be downloaded from Bernstein and Sons (<http://www.ben1stein-plus-sons.com>). Either source code or precompiled binary distributions can be downloaded.

RasMol comes in three display depths: 8-, 16-, and 32-bit. Eight-bit is the default, but if you have a high-resolution monitor, you may have to experiment and find out which executable is right for your system. You'll know you have a problem when you try to nm RasMol and it complains that no appropriate display has been detected. Start with the 8-bit version, and work your way up.

If you plan to compile RasMol yourself, you need to get into the *src* directory and edit the *Makefile* to produce the appropriate version. To do this, open the *Makefile* with an ASCII text editor such as *vim* or Emacs and search for the variable *DEPliDEF*. You should find something like this:

```
# DEPTHDEF = - DTHIRTYIWOBIT
DEPTHDEF = -DSIXTEENBIT
# DEPTHDEF = -DEIGHTBIT
```

In this example, *DEPliDEF* has been defined as 16-bit.

The # character at the beginning of a line marks that line as a comment, which isn't read by the *make* program when it scans the *Makefile*. Lines of code can be skipped over by being commented out; that is, marked as a comment. Remove the # character in front of the depth definition you need to use, and add it to comment out the others. Comment characters vary from programming language to programming language, but the notion of a comment line is common to all standard languages.

You may also need to edit the *rasmol.h* file, according to the install instructions.

Once you have the proper RasMol executable, whether you download it or compile it yourself, you need to copy it into */usr/local/bin* and copy the file *rasmol.hlp* into the directory */usr/local/lib/rasmol*. Then, in your web browser's preferences, you need to add RasMol as an application. If you're using Netscape, the default browser on most Linux systems, go to the Preferences_Navigator_Applications menu, select New, and enter the following values into the dialog box:

```
Description:Brookhaven PDB
MIMEType:chemical/x-pdb
```

Suffixes: .pdb

Application: /usr/local/bin/rasrnol

You may also want to create a second entry for the MIME type *chemicallx-ras*.

When run from the command line, RasMol opens a single graphics display window with a black background. The molecule can be rotated in this window either directly with the mouse, or with the sliders on the bottom and right side of the window. This window has five pulldown menus. The File menu contains commands for opening molecular structure files. The Display menu contains commands for changing the molecular display style to formats including ball and stick, cartoons, and spacefill. These display commands execute quickly, so you can try each of them out to see the different standard molecular display formats. The Colours menu allows you to change the color scheme of the entire molecule, and the Options menu changes the *di_play* style, allowing you to display the molecule in stereo, turn the display of heteroatom groups or labels on and off, etc. The Export menu allows you to write the displayed image in common electronic image formats such as GIF, PostScript, and PPM, which can be edited later using standard image manipulation programs that come with most Linux distributions, such as GIMP.

When you import or save files in RasMol, you do it from the RasMol command line. In the shell window from which you start RasMol, the command prompt changes to *RasMol>*. Enter *help commands* at this command prompt to see the full range of RasMol commands, including commands for selecting subsets of atoms. If RasMol complains that it can't find its help file, create a symbolic link to */usr/local*

lib/rasmol/rasmol.hlp in the directory in which you installed RasMol and/or the directory in which you are running it. Help commands allow you to create your own combinations of colors and structure display formats, including some not available from the menus; create interatomic distance monitors; and display some intermolecular interactions, such as hydrogen bonds and disulfide bridges.

Cn3D

Cn3D is an application from NCBI that can view protein structure files in NCBI ASN.1 format. If you use the NCBI databases frequently, you will also want to install this tool and set it up to work as an application in your browser.

To install Cn3D on a Linux workstation and set it up as a browser application, you simply need to download the Cn3D archive from NCBI, make a *Cn3D* directory on your own machine, move the archive into that directory, and extract it.

Then, in your web browser's application preferences, make the following new entry:

Description:NCBI ASN.1

MIMKtype:chernical/ncbi-asnl-binary

Suffixes: .prt

Application:/usr/local/cn3d/Cn3D

Cn3D opens two windows: a color structure viewer, in which a molecule can be rotated, colored according to different properties, and rendered in different display formats; a sequence viewer, which allows you to view sequences and alignments

corresponding to the displayed protein and to add graphics to the sequence display to highlight the location of secondary structure features.

Structural Alignment

Recently, there have been many attempts to make protein-structure classification an automatic and quantitative process, rather than an expert-curated process. Overlaying and comparing structures is a 3D problem that is much more resource-intensive than comparing ID sequence data. The automated structure comparison tools that exist, therefore, are available primarily as online tools for searching precomputed databases of structure comparisons.

Comparing Two Protein Structures

The most common parameter that expresses the difference between two protein structures is RMSD, or root mean squared deviation, in atomic positions between the two structures. RMSD can be computed as a function of all the atoms in a protein or as a function of some subset of the atoms, such as the protein backbone or the alpha-carbon positions only. Using a subset of the protein atoms is common, because it is likely that, when two protein structures are compared, they will not be identical to each other in sequence, and therefore the only atoms between which one-to-one comparisons in position can be made will be the backbone atoms.

This is the first context we've discussed in which the *orientation* of a molecular structure becomes important. Because protein structures are generally described in Cartesian coordinates, they essentially exist within a virtual space, and they come with a built-in orientation with respect to that space. RMSD is a function of the distance between atoms in one structure and the same atoms in another structure. Thus, if one molecule starts out in a different position with respect to the reference coordinate system, the other molecule-the RMSD between the two proteins-will be large whether they are similar or not.

In order to compute meaningful RMSDs, the two structures under consideration must first be superimposed, insofar as that is possible. Superimposition of protein structures usually starts with a sequence comparison. The sequence comparison establishes the one-to-one relationships between pairs of atoms from which the RMSD is computed. Atom-to-atom relationships, for the purpose of structure comparison, may actually occur between residues that aren't in the same relative position in the amino acid sequence. Sequence insertions and deletions can push two sequences out of register with each other, while the *core architecture* of the two structures remains similar.

Once atom-to-atom relationships between two structures are established, the task of a superposition program is to achieve an optimal superposition between the two programs-that is, the superposition with the smallest possible RMSD. Because protein scaffolds, or cores, can be similar in topology without being identical, it isn't usually possible to achieve perfect overlap in all pairs of atoms in two structures that are being compared. Overlaying one pair of atoms perfectly may push another pair of atoms further apart. Superposition algorithms optimize the orientation and spatial position of the two molecules with respect to each other.

Once optimal superpositions of all pairs of structures have been made, the RMSD values that are computed as a result can be compared with each other, because the structures have been moved to the same frame of reference before making the RMSD calculations.

ProFit

Usage: *profit reference.pdb mobile.pdb*

ProFit, developed by Andrew Martin at the University of Reading, United Kingdom, is an easy-to-use program for superimposing two protein structures. One protein is assigned by the user to be the reference structure, and the other protein is mobile with respect to the reference. ProFit outputs RMSD and can also write out coordinates for the superimposed proteins. ProFit allows the option of superimposing only selected regions of each protein so that domains can be examined independently. ProFit compiles and runs on any Unix workstation. ProFit may be downloaded from Andrew's web site (<http://www.bioinforg.uk/>).

DALI Domain Dictionary

The DALI Domain Dictionary (ODD) at the EEl is based on an automatic classification of protein domains by sequence identity. Rather than using a human-designed classification scheme, ODD is constructed by clustering protein neighbors within an abstract fold space. Instead of working with whole proteins, ODD classifies structures based on compact, recurring structures (called *domains*) that may repeat themselves within, and among, different protein structures. The content of ODD may also be familiar to you as FSSP, the "Fold classification based on StructureStructure alignment of Proteins" database.

ODD can be searched based on text keywords; it can also be viewed as a tree or a clickable graphical representation of fold space. Views of sequence data for conserved domains are available through the ODD interface, as well as connections to structural neighbors.

The superposition program (SUPPOS) that produces the structural alignments in DALI/FSSP is available within the WHAT IF software package of protein structure analysis tools, which is discussed in the later section "WHAT IF/WHAT CHECK."

CE and CL

The Combinatorial Extension of the Optimal Path (CE) is a sophisticated automatic structure alignment algorithm that uses characteristics of local geometry to "seed" structural alignments and then joins these regions of local similarity into an optimal path for the full alignment. Dynamic programming can then optimize the alignment.

CE is available either as a web server or as source code from the San Diego Super-computer Center. The web server allows you to upload files for pairwise comparison to each other or to proteins in the PDB, to compare a structure to all structures in the PDB, to compare a structure to a list of representative chains, and review alignments for specific protein families. CE also is fully integrated with the PDB's web site, and CE searches can be initiated directly from the web page generated for any protein you identify in a sequence search. Along with the source code, you can download a current, precomputed pairwise comparison database containing all structures in the PDB. If you're doing only a few comparisons, however, you probably won't even want to do this.

When using the CE server to compute similarities, there are several parameters that you can set, including cutoffs for percent sequence identity, percent of the alignment spanned by gaps, and percent length difference between two chains. You can also set an RMSD cutoff and a Z-score cutoff. The Z-score is a measure of the significance of an alignment relative to a random alignment, analogous to a BLAST E-value. A Z-score of 3.5 or above from CE usually indicates that two proteins have a similar fold.

Along with CE, the SDSC offers the Compound Likeness (CL) server, a suite of tools for probabilistic comparison between protein structures. In CL, you select either an entire protein structure or a structure fragment to use as a probe for searching the PDB. Search features include bond length and angle parameters, surface polarity and accessibility, dihedral angles, secondary structure, shape, and predicted alpha helix and beta sheet coefficients. CL allows you to ask the question "what else is chemically similar to this protein (or fragment) that is of interest to me" and to define chemical similarity very broadly. A full tutorial on CL is available at the CL web site (<http://cl.sdsc.edu/cll.html/>).

VAST

VAST is a pairwise structural alignment tool offered by NCB!. VAST reports slightly different parameters about structural comparison than CE does, and the underlying algorithm differs in significant respects. However, the results tend to be quite similar. VAST searches automatically allow you to view your superimposed protein structures in the Cn3D browser plug-in, with aligned sequences displayed in Cn3D as well. For practical purposes, either CE or VAST is sufficient to give you an idea of how two structures match up; if you are concerned about the algorithmic differences, both groups provide access to detailed explanations at their sites. Unlike CE, the VAST software doesn't appear to be available to download, so if you want to perform a large number of comparisons on your own server, CE may be preferable.

Summary

Popular sequence databases, such as GenBank and EMBL, have been growing at exponential rates. This deluge of information has necessitated the careful storage, organization and indexing of sequence information. Information science has been applied to biology to produce the field called Bioinformatics. Several tools have been developed to retrieve meaningful information from these databanks while sequence analysis tools help in working out the biological meaning of the raw sequences deposited in the databanks.

Model Questions

1. Give the description of various tools used in sequence analysis?
2. Give an account of the web based protein structure tools?

References

1. Developing Bioinformatics Computer Skills by Cynthia Gibas, Per Jambeck
2. Bioinformatics - Sequence and Genome Analysis- David W. Mount.
3. www.ebi.ac.uk/Tools

AUTHOR:

B.M.REDDY M.Tech. (HBTI, Kanpur)

Lecturer, Centre for Biotechnology

Acharya Nagarjuna University.

Lesson 3.1.2

BIOLOGICAL DATABASES

Objective

3.1.2.1 Introduction

3.1.2.2 What Is a Biological Database?

3.1.2.3 Different types of databases

3.1.2.4 Accession codes vs identifiers

3.1.2.5 Sequence Databases

3.1.2.6 Nucleotide sequence databases

3.1.2.7 Protein sequence databases

3.1.2.8 Macromolecular 3D structure databases

3.1.2.9 Other relevant databases

3.1.2.10 Non-redundant database

Summary

Model Questions

References

Objective

To know about the biological databases and their classification basing on different characteristics.

3.1.2.1 Introduction

As much as possible of a particular type of information should be available in one single place (book, site, and database). Published data may be difficult to find or access and collecting it from the literature is very time-consuming. And not all data is actually published in an article. Since analysis of biological data almost always involves computers, having the data in computer-readable form (rather than printed on paper) is a necessary first step.

The computer became the storage medium of choice as soon as it was accessible to ordinary scientists. Databases were distributed on tape, and later on various kinds of disks. When universities and academic institutes were connected to the Internet or its precursors (national computer networks), it is easy to understand why it became the medium of choice. And it is even easier to see why the World Wide Web (WWW, based on the Internet protocol HTTP) since the beginning of the 1990s is the standard method of communication and access for nearly all biological databases.

As biology has increasingly turned into a data-rich science, the need for storing and communicating large datasets has grown tremendously. The obvious examples are the nucleotide sequences, the protein sequences, and the 3D structural data produced by X-ray crystallography and macromolecular NMR. When Sanger first discovered the method to sequence proteins, there was a lot of excitement in the field of Molecular Biology. Initial interest in Bioinformatics was propelled by the necessity to create databases of biological sequences. Biological databases can be broadly classified in to

sequence and structure databases. Sequence databases is applicable to both nucleic acid sequences and protein sequences, whereas structure database is applicable to only Proteins. The first database was created within a short period after the Insulin protein sequence was made available in 1956. Incidentally, Insulin is the first protein to be sequenced. The sequence of Insulin consisted of just 51 residues (analogous to alphabets in a sentence) which characterize the sequence. Around mid nineteen sixties, the first nucleic acid sequence of Yeast tRNA with 77 bases (individual units of nucleic acids) was found out. During this period, three dimensional structure of proteins were studied and the well known Protein Data Bank was developed as the first protein structure database with only 10 entries in 1972.

One of the first biological sequence databases was probably the book "Atlas of Protein Sequences and Structures" by Margaret Dayhoff and colleagues, first published in 1965. It contained the protein sequences determined at the time, and new editions of the book were published well into the 1970s. Its data became the foundation for the PIR database. Other types of data that are or will soon be available in databases are metabolic pathways, gene expression data (microarrays) and other types of data relating to biological function and processes.

One very important issue is the frequency and type of errors that the entries in a database have. Naturally, this depends strongly on the type of data, and whether the **database is curated** (added, deleted, or modified by a defined group of people) or not. For the sequence databases, the errors may be either in the sequence itself (misprint, wrong on entry, genuine experimental error...) or in the annotation (mistaken features, errors in references,...). In the 3D structure database (PDB), structures have been deposited which were later discovered to contain severe errors. The error handling policy differs considerably between databases. If one needs to use any particular database heavily, then the implications of its particular policy need to be considered.

3.1.2.2 What Is a Biological Database?

A database is any collection of meaningful data relevant to some aspects of the real world and datamining can be defined as extraction of hidden predictive information from large databases. Biology today no longer remains merely the study of living organisms, there is huge amount of data (sequence, structural etc..) being generated by large number of scientists across the globe and this data is growing exponentially day by day . Biological research is becoming increasingly database driven, motivated partly by the advent of large-scale functional genomics and proteomics experiments of gene expression. These provide wealth of information on each of the thousands of proteins encoded by a genome. Data in biology are becoming very diverse and abundant with the increase in biological knowledge, computer – based databases have become very essential for this task. Databases are essential for managing similar kind of data and developing a network to access them across the globe eg:SGD *saccharomyces* genome database - this database contains complete genomic, proteomic data of *saccharomyces*.

“A biological database is a large, organized body of persistent data, usually associated with computerized software designed to update, query, and retrieve components of the data stored within the system”.

A simple database might be a single file containing many records, each of which includes the same set of information. For example, a record associated with a

nucleotide sequence database typically contains information such as contact name, the input sequence with a description of the type of molecule, the scientific name of the source organism from which it was isolated, and often, literature citations associated with the sequence.

For researchers to benefit from the data stored in a database, two additional requirements must be met:

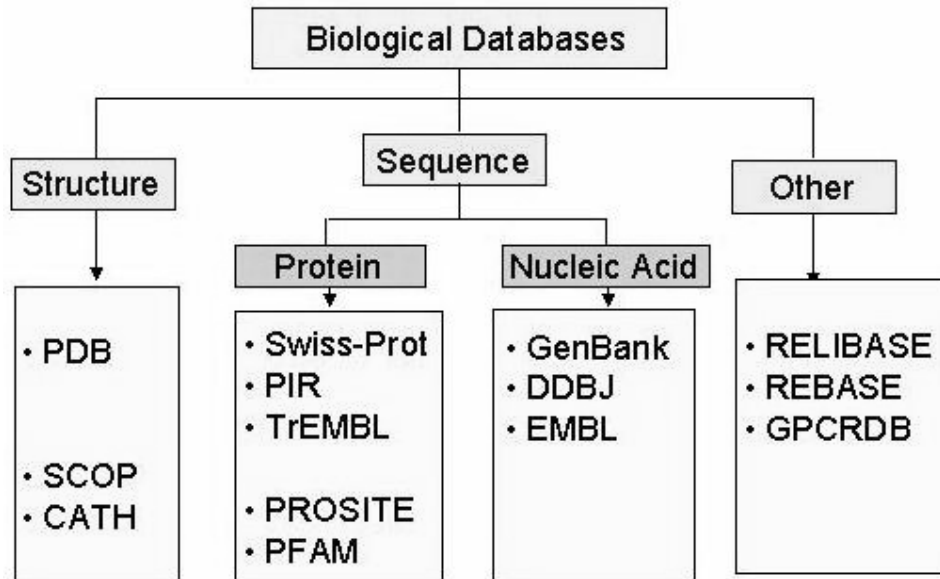
- easy access to the information
- a method for extracting only that information needed to answer a specific biological question

3.1.2.3 Different types of databases

One may characterize the available biological databases by several different properties:

- Type of data
 - nucleotide sequences
 - protein sequences
 - proteins sequence patterns or motifs
 - macromolecular 3D structure
 - gene expression data
 - metabolic pathways
- Data entry and quality control
 - Scientists (teams) deposit data directly (Non-curated)
 - Appointed curators add and update data (Curated)
 - Are erroneous data removed or marked?
 - Type and degree of error checking
 - Consistency, redundancy, conflicts, updates
- Primary or derived data
 - Primary databases: experimental results directly into database (ex. GenBank,
 - Secondary databases: results of analysis of primary databases
 - Aggregate of many databases (Composite)
 - Links to other data items
 - Combination of data
 - Consolidation of data
- Maintainer status
 - Large, public institution funded by government (EMBL, NCBI)
 - Quasi-academic institute (Swiss Institute of Bioinformatics, TIGR)
 - Academic group or scientist
 - Commercial company

Broad classification of Biological Databases



3.1.2.4 Accession codes vs identifiers

Many databases in bioinformatics (SWISS-PROT, EMBL, GenBank, Pfam) use a system where an entry can be identified in two different ways; essentially it has two names:

- Identifier
- Accession code (or number)

The question how to deal with changed, updated and deleted entries in databases is a very tricky problem, and the policies for how accession codes and identifiers are changed or kept constant are not completely consistent between databases or even over time for one single database.

The exact definition of what the identifier and accession code are supposed to denote varies between the different databases, but the basic idea is the following:

Identifier

An identifier ("locus" in GenBank, "entry name" in SWISS-PROT) is a string of letters and digits that generally is interpretable in some meaningful way by a human, for instance as a recognizable abbreviation of the full protein or gene name.

SWISS-PROT uses a system where the entry name consists of two parts: the first denotes the protein and the second part denotes the species it is found in. For example, KRAF_HUMAN is the entry name for the Raf-1 oncogene from Homo sapiens.

An identifier can change. For example, the database curators may decide that the identifier for an entry no longer is appropriate. This does not happen very often.

Accession code (number)

An accession code (or number) is a number (possibly with a few characters in front) that uniquely identifies an entry. For example, the accession code for KRAF_HUMAN in SWISS-PROT is P04049.

The main conceptual difference from the identifier is that it is supposed to be stable: any given accession code will, as soon as it has been issued, always refer to that entry, or its ancestors. It is often called the primary key for the entry. The accession code,

once issued, must always be possible to find again, even after large changes have been made to the entry.

In the case where two entries are merged into one single, then the new entry will have both accession codes, where one will be the primary and the other the secondary accession code. When an entry is split into two, both new entries will get new accession codes, but will also have the old accession code as secondary codes.

3.1.2.5 Sequence Databases

Sequence databases are very common databases that almost all molecular biologists are familiar with. These mainly include three different types.

Annotated sequence databases

Here each record is annotated or the information in addition to the sequence is provided. By using these keywords the relevant information in the database can be searched successfully. Annotated sequence databases are typically used for identifying the function for unknown sequences by database similarity searching. If an unknown sequence matches a sequence from the database then its function can often be inferred from the annotations associated with the matching sequence. This way information retrieval is made easy and rapid. Eg. GenBank, EMBL and SWISS-PROT etc.

Low-annotation sequence databases

These databases mainly contain newly discovered sequences, generally with minimal annotations because many of these sequences are uncharacterized. These data bases can be a very useful source of new gene sequences. Eg: EST databases, high-throughput genome sequences.

3.1.2.6 Nucleotide sequence databases

Primary nucleotide sequence databases

The databases EMBL, GenBank, and DDBJ are the three primary nucleotide sequence databases: They include sequences submitted directly by scientists and genome sequencing group, and sequences taken from literature and patents. There is comparatively little error checking and there is a fair amount of redundancy.

The entries in the EMBL, GenBank and DDBJ databases are synchronized on a daily basis, and the accession numbers are managed in a consistent manner between these three centers.

The nucleotide databases have reached such large sizes that they are available in subdivisions that allow searches or downloads that are more limited, and hence less time-consuming. For example, GenBank has currently 17 divisions. There are no legal restrictions on the use of the data in these databases. However, there are patented sequences in the databases.

EMBL www.ebi.ac.uk/embl/

The EMBL (European Molecular Biology Laboratory) nucleotide sequence database is maintained by the European Bioinformatics Institute (EBI) in Hinxton, Cambridge, UK. As of 16 Jan 2001, it contained 10,378,022 records with a total of 11,302,156,937 bases; see the EMBL DB statistics page.

It can be accessed and searched through the SRS system at EBI, or one can download the entire database as flat files. An example of what an entry looks like is given for the human raf oncogene protein, ID: HSRFR.

GenBank www.ncbi.nlm.nih.gov/Genbank/

The GenBank nucleotide database is maintained by the National Center for Biotechnology Information (NCBI), which is part of the National Institute of Health (NIH), a federal agency of the US government.

It can be accessed and searched through the Entrez system at NCBI, or one can download the entire database as flat files. An example of what an entry looks like is given for the human raf oncogene protein, Locus: HSRAFR.

DDBJ www.ddbj.nig.ac.jp

The DNA Data Bank of Japan began as collaboration with EMBL and GenBank. It is run by the National Institute of Genetics. One can search for entries by accession number, FASTA/BLAST, keywords and regular expressions.

Other nucleotide sequence databases

The following databases contain subsets of the EMBL/GenBank databases. Some also contain more information or links than the primary ones, or have a different organization of the data to better suit some specific purpose. However, the nucleotide sequences themselves should always be available in the EMBL/GenBank databases. In this sense, the databases below are secondary databases.

UniGene www.ncbi.nlm.nih.gov/UniGene/

The UniGene system attempts to process the GenBank sequence data into a non-redundant set of gene-oriented clusters. Each UniGene cluster contains sequences that represent a unique gene, as well as related information such as the tissue types in which the gene has been expressed and map location.

SGD genome-www.stanford.edu/Saccharomyces/

The Saccharomyces Genome Database (SGD) is a scientific database of the molecular biology and genetics of the yeast *Saccharomyces cerevisiae*.

EBI Genomes www.ebi.ac.uk/genomes/

This web site provides access and statistics for the completed genomes, and information about ongoing projects.

Genome Biology www.ncbi.nlm.nih.gov/Genomes/

The Genome Biology site at NCBI contains information about the available complete genomes.

Ensembl www.ensembl.org

Ensembl is a joint project between EMBL-EBI and the Sanger Centre to develop a software system which produces and maintains automatic annotation of eukaryotic genomes.

3.1.2.7 Protein sequence databases

The two protein sequence databases SWISS-PROT and PIR are different from the nucleotide databases in that they are both curated. This means that groups of designated curators (scientists) prepare the entries from literature and/or contacts with external experts.

SWISS-PROT, TrEMBL www.expasy.ch/sprot/

SWISS-PROT is a protein sequence database which strives to provide a high level of annotations (such as the description of the function of a protein, its domains structure, post-translational modifications, variants, etc.), a minimal level of redundancy and high level of integration with other databases.

It was started in 1986 by Amos Bairoch in the Department of Medical Biochemistry at the University of Geneva. This database is generally considered one of the best protein sequence databases in terms of the quality of the annotation. Release 39.12 (11 Jan 2001) contained 92,211 entries.

TrEMBL is a computer-annotated supplement of SWISS-PROT that contains all the translations of EMBL nucleotide sequence entries not yet integrated in SWISS-PROT. The procedure that is used to produce it was developed by Rolf Apweiler. Release 15.14 (5 Jan 2001) contained 378,152 entries. The annotation of an entry in TrEMBL has not (yet) reached the standards required for inclusion into SWISS-PROT proper.

SWISS-PROT and TrEMBL are developed by the SWISS-PROT groups at Swiss Institute of Bioinformatics (SIB) and at EBI. The databases can be accessed and searched through the the SRS system at ExpASY, or one can download the entire database as one single flat file. An example of what an entry looks like is given for the human raf oncogene protein, ID KRAF_HUMAN.

The SWISS-PROT database has some legal restrictions: the entries themselves are copyrighted, but freely accessible and usable by academic researchers. Commercial companies must pay a license fee from SIB to use SWISS-PROT.

PIR pir.georgetown.edu

The Protein Information Resource (PIR) is a division of the National Biomedical Research Foundation (NBRF) in the US. It is involved in a collaboration with the Munich Information Center for Protein Sequences (MIPS) and the Japanese International Protein Sequence Database (JIPID). Release 67.00 (31 Dec 2000) contains 198,801 entries.

PIR grew out of Margaret Dayhoff's work in the middle of the 1960s. It strives to be comprehensive, well-organized, accurate, and consistently annotated. However, it is generally believed that it does not reach the level of completeness in the entry annotation as does SWISS-PROT. Although SWISS-PROT and PIR overlap extensively, there are still many sequences which can be found in only one of them.

One can search for entries or do sequence similarity searches at the PIR site. The database can also be downloaded as a set of files. An example of what an entry looks like is given for the human raf-1 oncogene protein, ID TVHUF6.

PIR also produces the NRL-3D, which is a database of sequences extracted from the three-dimensional structures in the Protein Databank (PDB) (see also the following page in this lecture. The NRL_3D database makes the sequence information in PDB available for similarity searches and retrieval and provides cross-reference information for use with the other PIR Protein Sequence Databases.

Sequence motif databases

Pfam www.sanger.ac.uk/Software/Pfam/, www.cgr.ki.se/Pfam/

Pfam is a database of protein families defined as domains (contiguous segments of entire protein sequences). For each domain, it contains a multiple alignment of a set of defining sequences (the seeds) and the other sequences in SWISS-PROT and TrEMBL that can be matched to that alignment.

The database was started in 1996 and is maintained by a consortium of scientists, among them Erik Sonnhammer (CGR, KI, Sweden), Sean Eddy (WashU, St Louis USA), Richard Durbin, Alan Bateman and Ewan Birney (Sanger Centre, UK). Release 5.5 (Sep 2000) contains 2478 families.

The alignments can be converted into hidden Markov models (HMM), which can be used to search for domains in a query protein sequence. The software HMMER (by Sean

Eddy) is the computational foundation for Pfam. The domain structure of protein sequences in SWISS-PROT and TrEMBL are available directly from the Pfam web sites, and it is also possible to search for domains in other sequences using servers at the web sites.

The Pfam database can be searched, or used to identify domains in a sequence, or downloaded from the websites above. An example of a multiple sequence alignment that defines a protein family (domain) is given for the Raf-like Ras-binding domain (Pfam name RBD, accession code PF02196).

The Pfam database is licensed under the GNU General Public License, which basically makes it available to anyone, but imposes the restriction that derivative works (new databases, modifications) must be made available in source form.

PROSITE www.expasy.ch/prosite/

PROSITE is a database of protein families and domains. It consists of biologically significant sites, patterns and profiles that help to reliably identify to which known protein family (if any) a new sequence belongs.

It was started by Amos Bairoch, is part of SWISS-PROT and is maintained in the same way as SWISS-PROT. The basis of it are regular expressions describing characteristic subsequences of specific protein families or domains. PROSITE has been extended to contain also some profiles, which can be described as probability patterns for specific protein sequence families.

The site above can be used to search by keyword or other text in the entries, to search for a pattern in a sequence, or to search for proteins in SWISS-PROT that match a pattern. An example of a PROSITE regular expression is given for the Ras GTPase-activating proteins signature pattern (RAS_GTPASE_ACTIV_1, accession code PS00509).

3.1.2.8 Macromolecular 3D structure databases

PDB www.rcsb.org/pdb/

The PDB is the main primary database for 3D structures of biological macromolecules determined by X-ray crystallography and NMR. Structural biologists usually deposit their structures in the PDB on publication, and some scientific journals require this before accepting a paper. It also accepts the experimental data used to determine the structures (X-ray structure factors and NMR restraints) and homology models. As of 16 Jan 2001 the PDB contained 14,109 entries, the majority of which (11,611) are X-ray structures.

The Protein Data Bank (PDB) was established in the 1970s at the Brookhaven Lab on Long Island, New York State, US. In 1999, the management was moved to the Research Collaboratory for Structural Bioinformatics (RCSB, a joint organisation between Rutgers University, San Diego Supercomputer Center and NIST).

The PDB entries contain the atomic coordinates, and some structural parameters connected with the atoms (B-factors, occupancies), or computed from the structures (secondary structure). The PDB entries contain some annotation, but it is not as comprehensive as in SWISS-PROT. Fortunately, there are cross-links between the databases in both file formats. Here is an example of an entry is the the Ras-binding domain of the human Raf-1 oncogene in the traditional PDB format and in the mmCIF format.

There are no legal restrictions on the use of the data in the PDB.

SCOP scop.mrc-lmb.cam.ac.uk/scop/

The SCOP (Structural Classification of Proteins) database was started by Alexey Murzin in 1994 (Lab of Molecular Biology, MRC, Cambridge, UK). Its purpose is to classify protein 3D structures in a hierarchical scheme of structural classes. It is maintained by experts (manually), and all protein structures in the PDB are classified, and it is updated as new structures are deposited in the PDB.

This is a typical secondary database; it is based on data in a primary database (in this case the PDB), but adds information through analysis and/or organisation, in this case the classification of protein 3D structures into a hierarchical scheme of folds, superfamilies and families.

CATH www.biochem.ucl.ac.uk/bsm/cath/

The CATH database (Class, Architecture, Topology, Homologous superfamily) is a hierarchical classification of protein domain structures, which clusters proteins at four major structural levels. Although the aim is very similar to SCOP, the scheme is different, and the philosophy and practical details of producing the classification differ considerably. For instance, a larger part of the classification of a new protein 3D structure is made automatically by software. It was started by Christine Orengo in Janet Thornton's lab (University College London) in 1996.

3.1.2.9 Other relevant databases

GeneCards bioinformatics.weizmann.ac.il/cards/

GeneCards is a database of human genes, their products and their involvement in diseases. It offers concise information about the functions of all human genes that have an approved symbol, as well as selected others. It is a typical example of a secondary database, which contains many links to other databases, and attempts to consolidate the information that is available for a specific class of entity, in this case human genes.

The web site can be used for free by academics, but companies must obtain a license.

KEGG www.genome.ad.jp/kegg/

The Kyoto Encyclopedia of Genes and Genomes (KEGG) is an effort to computerize current knowledge of molecular and cellular biology in terms of the information pathways that consist of interacting molecules or genes and to provide links from the gene catalogs produced by genome sequencing projects.

Although there seems to be no explicit license information, it appears that the web site can be used by anyone (including companies). However, downloading of the KEGG distribution requires a license agreement for non-academic users.

Amos' WWW links page www.expasy.ch/alinks.html

A page of many links to biological databases and/or web sites, maintained by Amos Bairoch.

3.1.2.10 Non-redundant database

Due to the presence of several sequence database, the main problem while searching for sequences are exploring every database and mostly getting same hits repeatedly across these databases and in this way chances of missing valuable data are more. To overcome this problem a non-redundant database having no duplicate entries are put together to form one common database for eg: NR-Nucleotide database containing sequences from Gen Bank + EMBL + DDBJ + PDB Dna and NR-from SWISS-PROT + Tr EMBL + Gen pept + PDB protein both are maintained at NCBI. Non-

redundant data bases contain only sequences the annotation is not a part of this database. But hyperlinks are provided to access annotations.

Systems for searching, indexing and cross-referencing

The usefulness of a database can be increased enormously if it is easy to find entries that satisfy certain search criteria. Some examples of searches that a scientist might want to do:

- All entries with the keyword "GTPase".
- The entries which have a given literature reference (by author or article).
- All proteins with the keyword "ribosomal" from human (organism).

The databases themselves may contain this information, but some software systems must be used to actually perform this kind of search. There are different ways of designing such systems, and two examples are mentioned here.

SRS

The Sequence Retrieval System (SRS) developed by Thure Etzold is a system for integrating heterogeneous databases. It is based on premade indexes of the items (words, entries, data fields, text,...) found in a set of documents (database files). Apart from the database files themselves, the indexing procedure requires a grammar (Icarus) that describes what different words in the data files mean, how they are to be indexed, and how they cross-reference to other items in other databases. SRS is a web-oriented system located on a server which is accessed through HTML pages and CGI scripts.

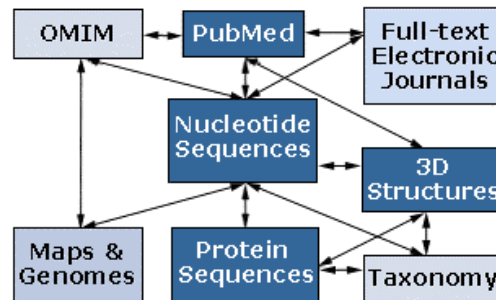
SRS started as an academic project, but it is now a commercial system developed and marketed by LION Bioscience AG. However, academic groups can license the SRS system free of charge and set it up at a server in their own lab.

EBI runs an SRS service which can be used by anyone. It indexes a large number of databases, and it also provides a well-defined web interface which allows programs or web sites to create links that query SRS at EBI.

Entrez

The Entrez system developed and accessible at the NCBI Entrez site. Similar to some extent to the SRS system, it provides search facilities for a large number of databases, and provides links between them. It provides a well-defined web interface which allows programs or web sites to define links that will query Entrez.

However, it appears that the Entrez system is not available to set up at one's own server. It is purely a system for accessing and searching the databases at NCBI.



Summary

A database is any collection of meaningful data relevant to some aspects of the real world and datamining can be defined as extraction of hidden predictive information from large databases. Biological databases can be broadly classified in to sequence and structure databases. Sequence databases is applicable to both nucleic acid sequences and protein sequences, whereas structure database is applicable to only Proteins.

Model Questions:

1. What is biological database? Classify them on different characteristics with examples?
2. Briefly write about the sequence databases?

References:

1. Introduction to Bioinformatics by Arthur M. Lesk.
2. Rolf Apweiler ,Protein sequence databases ,Current Opinion in Chemical Biology 2004, 8:76–80
3. Developing Bioinformatics Computer Skills by Cynthia Gibas, Per Jambeck.

AUTHOR:

B.M.REDDY M.Tech. (HBTI, Kanpur)

Lecturer, Centre for Biotechnology

Acharya Nagarjuna University.

Lesson 3.1.3

Searching Sequence Databases

Objective

3.1.3.2 Sequence database searching

3.1.3.3 Sequence Identification

3.1.3.4 Sequence Formats

3.1.3.5 Sequence search methods

3.1.3.5.1 BLAST

3.1.3.5.2 FASTA

3.1.3.6 Searching Swiss-Prot

3.1.3.7 Searching PIR

3.1.3.8 Searching sequence databases over the internet

Summary

Model Questions

References

Objective: The objective is to know about the sequence searching, sequence formats and methods using for sequence searching.

3.1.3.1 Introduction

Sequence database similarity searching is one of the most common computing techniques in modern biology. It allows the large repositories of DNA and protein sequence information to be queried using a sequence, with the goal of identifying database sequences homologous to this query sequence. This technique can be extremely useful in the study of gene and protein structure, function and evolution. Sequence searches tries to identify relevant features in a query sequence by comparing it against the body of available knowledge from other sequences. This usually takes the form of a sequence by sequence comparison between our query sequence and every single sequence known to date. For the less insecure, it most often shapes as a comparison with only relevant sequences, or selected subsets, and even to "concentrated" wisdom distilled from the existing knowledge base, resulting in faster and more accurate results.

The reasons of searching:

- Is my sequence **new** or not?
- Is there something **identical** to it in the database?
- Are there **similar** things?
- Can we recognise **subsequences of interest** by comparing it to other sequences?

What is expected to be found?

.The possibilities:

- We find something **identical** to all of *our* sequence - *oh well...* What can we learn *from* the annotation in the database? In what way is *our* information different from it?

- We find something **identical** to part of our sequence - *good news/What can we learn from the additional information in the database? What does our new sequence add to the picture?*
- We find something **similar** to our sequence – *good news! What do the sequences have in common? What are the differences? How much difference is there? Does it make sense? What part of sequence annotation can we transfer to our sequence?*

-

The focused approach:

- Goal: we search *for* homologues
- Method: we search by similarity
- Result: we find similar things
- Discussion: we have to figure *out* what is homologous and what is not

3.1.3.2 Sequence database searching

The most obvious first stage in the analysis of any new sequence is to perform comparisons with sequence databases to find homologues. These searches can now be performed just about anywhere and on just about any computer. In addition, there are numerous web servers for doing searches, where one can post or paste a sequence into the server and receive the results interactively:

There are many methods for sequence searching. By far the most well known are the BLAST suite of programs. One can easily obtain versions to run locally (either at NCBI or Washington University), and there are many web pages that permit one to compare a protein or DNA sequence against a multitude of gene and protein sequence databases. To name just a few:

- National Center for Biotechnology Information (USA) Searches
- European Bioinformatics Institute (UK) Searches
- BLAST search through SBASE (domain database; ICGEB, Trieste)
- and others too numerous to mention.

One of the most important advances in sequence comparison recently has been the development of both gapped BLAST and PSI-BLAST (position specific iterated BLAST). Both of these have made BLAST much more sensitive, and the latter is able to detect very remote homologues by taking the results of one search, constructing a *profile* and then using this to search the database again to find other homologues (the process can be repeated until no new sequences are found). It is essential that one compares any new protein sequence to the database with PSI-BLAST to see if known structures can be found prior to doing any of the other methods discussed in the next sections.

Other methods for comparing a single sequence to a database include:

- The FASTA suite (William Pearson, University of Virginia, USA)
- SCANPS (Geoff Barton, European Bioinformatics Institute, UK)
- BLITZ (Compugen's fast Smith Waterman search)
- and others.

It is also possible to use multiple sequence information to perform more sensitive searches. Essentially this involves building a *profile* from some kind of multiple sequence alignment. A profile essentially gives a score for each type of amino acid at each position in the sequence, and generally makes searches more sensitive. Tools for doing this include:

- PSI-BLAST (NCBI, Washington)
- ProfileScan Server (ISREC, Geneva)
- HMMER Hidden Markov Model searching (Sean Eddy, Washington University)
 - Wise package (Ewan Birney, Sanger Centre; this is for protein versus DNA comparisons)
 - and several others.

A different approach for incorporating multiple sequence information into a database search is to use a MOTIF. Instead of giving every amino acid some kind of score at every position in an alignment, a motif ignores all but the most invariant positions in an alignment, and just describes the key residues that are conserved and define the family. Sometimes this is called a "signature". For example, "H-[FW]-x-[LIVM]-x-G-x(5)-[LV]-H-x(3)-[DE]" describes a family of DNA binding proteins. It can be translated as "histidine, followed by either a phenylalanine or tryptophan, followed by an amino acid (x), followed by leucine, isoleucine, valine or methionine, followed by any amino acid (x), followed by glycine,... [etc.]".

PROSITE (ExPASy Geneva) contains a huge number of such patterns, and several sites allow you to search these data:

- ExPASy
- EBI

It is best to search a few different databases in order to find as many homologues as possible. A very important thing to do, and one which is sometimes overlooked, is to compare any new sequence to a database of sequences for which 3D structure information is available. Whether or not your sequence is homologous to a protein of known 3D structure is not obvious in the output from many searches of large sequence databases. Moreover, if the homology is weak, the similarity may not be apparent at all during the search through a larger database.

One last thing to remember is that one can save a lot of time by making use of pre-prepared protein alignments. Many of these alignments are hand edited by experts on the particular protein families, and thus represent probably the best alignment one can get given the data they contain (i.e. they are not always as up to date as the most recent sequence databases). These databases include:

- SMART (Oxford/EMBL)
- PFAM (Sanger Centre/Wash-U/Karolinska Intitutet)
- COGS (NCBI)
- PRINTS (UCL/Manchester)
- BLOCKS (Fred Hutchinson Cancer Research Centre, Seattle)
- SBASE (ICGEB, Trieste)

Generally one can compare a protein sequence to these databases via a variety of techniques. These can also be very useful for the domain assignment.

3.1.3.3 Sequence Identification

The two types of sequence identification numbers, **GI** and **VERSION**, have different formats and were implemented at different points in time.

1. **GI** number (sometimes written in lower case, "**gi**") is simply a series of digits that are assigned consecutively to each sequence record processed by NCBI. The GI number bears no resemblance to the Accession number of the sequence record.

- nucleotide sequence GI number is shown in the VERSION field of the database record
 - protein sequence GI number is shown in the CDS/db_xref field of a nucleotide database record, and the VERSION field of a protein database record
2. **VERSION** is made of the accession number of the database record followed by a dot and a version number (and is therefore sometimes referred to as the "**accession.version**")
- nucleotide sequence version contains two letters followed by six digits, a dot, and a version number (or for older nucleotide sequence records, the format is one letter followed by five digits, a dot, and a version number)
 - protein sequence version contains three letters followed by five digits, a dot, and a version number

The GI number has been used for many years by NCBI to track sequence histories in GenBank and the other sequence databases it maintains. The VERSION system of identifiers was adopted in February 1999 by the International Nucleotide Sequence Database Collaboration (GenBank, EMBL, and DDBJ).

The two systems of identifiers run in parallel to each other. That is, when any change is made to a sequence, it receives a new GI number AND an increase to its version number.

3.1.3.4 Sequence Formats

Sequence formats are simply the way in which the amino acid or DNA sequence is recorded in a computer file. Different programs expect different formats, so if you are to submit a job successfully, it is important to understand what the various formats look like.

In order to successfully submit a job it is important to understand what the various sequence formats used for describing biological sequences are and what their basic structure is. The job submission forms are fairly flexible but cannot cope with too much inconsistency.

You can submit sequence to the search and analysis programs in any of the formats mentioned in the options your chosen tool.

Plain sequence format

A sequence in plain format may contain only IUPAC characters and spaces (no numbers!).

Note: A file in plain sequence format may only contain **one** sequence, while most other formats accept several sequences in one file.

● An example sequence in plain format is:

```
acaagatgccattgtccccggcctctgctgctgctgctctccggggccacggccaccgctgcctgccctggagggtgg
ccccaccggccgagacagcgagcatatgcaggaagcggcaggaataaggaaaagcagcctcctgactttctcgttggtgtg
ttgagtggacctcccaggccagtgccgggcccctcataggagaggaagctcgggaggtggccaggcggcaggaaggcgcacc
ccccagcaatccgcgcgccgggacagaatgcctcgaggaacttctctggaagaccttctcctctgcaaataaacctcac
ccatgaatgctcacgcaagttaattacagacctgaa
```

EMBL format

A sequence file in EMBL format can contain several sequences. One sequence entry starts with an identifier line ("ID"), followed by further annotation

lines. The start of the sequence is marked by a line starting with "SQ" and the end of the sequence is marked by two slashes ("//").

An example sequence in EMBL format is:

```
ID AB000263 standard; RNA; PRI; 368 BP.
XX
AC AB000263;
XX
DE Homo sapiens mRNA for prepro cortistatin like peptide, complete cds.
XX
SQ Sequence 368 BP;
   acaagatgcc attgtcccc ggctctctgc tgetgctgct ctccggggcc acggccaccg    60
   ctgccctgcc cctggagggt ggccccaccg gccgagacag cgagcatatg caggaagcgg    120
   caggaataag gaaaagcagc ctctgactt tctctgcttg gtggtttgag tggacctccc    180
   aggccagtgc cgggccctc ataggagagg aagctcggga ggtggccagg cggcaggaag    240
   gcgaccccc ccagcaatcc gcgcgccggg acagaatgcc ctgcaggaac ttctctgga    300
   agacctctc ctctgcaaa taaacctca ccatgaatg ctcacgcaag ttaattaca    360
   gacctgaa                                368
//
```

FASTA format

A sequence file in FASTA format can contain several sequences. Each sequence in FASTA format begins with a single-line description, followed by lines of sequence data. The description line must begin with a greater-than (">") symbol in the first column.

An example sequence in FASTA format is:

```
>AB000263 |acc=AB000263|descr=Homo sapiens mRNA for prepro cortistatin like
peptide, complete cds. |len=368
acaagatgccattgtccccggcctctgctgctgctgctctccggggccacggccaccgctgccctgccctggagggtgg
ccccaccggccgagacagcagcatatgcaggaagcggcaggaataaggaaaagcagcctcctgacttctctgcttggtgt
ttgagtggacctccagccagtccgggccctcataggagaggaagctcgggaggtggccaggcggcaggaaggcgcacc
ccccagcaatccgcgcgccgggacagaatgccctgcaggaacttctctggaagaccttctctctgcaataaaacctcac
ccatgaatgctcacgcaagttaattacagacctgaa
```

GCG format

A sequence file in GCG format contains exactly one sequence, begins with annotation lines and the start of the sequence is marked by a line ending with two dot ("..") characters. This line also contains the sequence identifier, the sequence length and a checksum. This format should only be used if the file was created with the GCG package.

An example sequence in GCG format is:

```
ID AB000263 standard; RNA; PRI; 368 BP.
XX
AC AB000263;
XX
DE Homo sapiens mRNA for prepro cortistatin like peptide, complete cds.
XX
SQ Sequence 368 BP;
AB000263 Length: 368 Check: 4514 ..
```

```

1 acaagatgcc attgtcccc ggcctctgc tgctgctgct ctccggggcc acggccaccg
61 ctgcctgcc cctggagggt ggccccaccg gccgagacag cgagcatatg caggaagcgg
121 caggaataag gaaaagcagc ctctgactt tctcgcttg gtggttgag tggacctccc
181 aggccagtgc cgggccctc ataggagagg aagctcggga ggtggccagg cggcaggaag
241 gcgcaccccc ccagcaatcc gcgcgccggg acagaatgcc ctgcaggaac ttctctgga
301 agaccttctc ctctgcaaa taaaacctca ccatgaatg ctacgcaag ttaattaca
361 gacctgaa

```

GCG-RSF (rich sequence format)

The new GCG-RSF can contain several sequences in one file. This format should only be used if the file was created with the GCG package.

GenBank format

A sequence file in GenBank format can contain several sequences. One sequence in GenBank format starts with a line containing the word LOCUS and a number of annotation lines. The start of the sequence is marked by a line containing "ORIGIN" and the end of the sequence is marked by two slashes ("//").

An example sequence in GenBank format is:

```

LOCUS   AB000263           368 bp  mRNA  linear  PRI 05-FEB-1999
DEFINITION  Homo sapiens mRNA for prepro cortistatin like peptide, complete
            cds.

```

```

ACCESSION  AB000263

```

```

ORIGIN

```

```

1 acaagatgcc attgtcccc ggcctctgc tgctgctgct ctccggggcc acggccaccg
61 ctgcctgcc cctggagggt ggccccaccg gccgagacag cgagcatatg caggaagcgg
121 caggaataag gaaaagcagc ctctgactt tctcgcttg gtggttgag tggacctccc
181 aggccagtgc cgggccctc ataggagagg aagctcggga ggtggccagg cggcaggaag
241 gcgcaccccc ccagcaatcc gcgcgccggg acagaatgcc ctgcaggaac ttctctgga
301 agaccttctc ctctgcaaa taaaacctca ccatgaatg ctacgcaag ttaattaca
361 gacctgaa

```

```
//
```

IG format

A sequence file in IG format can contain several sequences, each consisting of a number of comment lines that must begin with a semicolon (";"), a line with the sequence name (it may not contain spaces!) and the sequence itself terminated with the termination character '1' for linear or '2' for circular sequences.

An example sequence in IG format is:

```
; comment
```

```
; comment
```

```
AB000263
```

```

acaagatgccattgtccccggcctctgctgctgctgctctccggggccacggccaccgctgcctgccctggagggtgg
ccccaccggccgagacagcgagcatatgcaggaagcggcaggaataaggaaaagcagcctctgactttctcgcttggtgt
ttgagtggacctccagccagtgcggggccctcataggagaggaagctcgggaggtggccaggcggcaggaaggcgcacc
ccccagcaatccgcgcggggacagaatgccctgcaggaacttctctggaagaccttctctctgcaataaaacctcac
ccatgaatgctcacgcaagttaattacagacctgaa1

```

IUPAC nucleic acid codes

To represent ambiguity in DNA sequences the following letters can be used (following the rules of the *International Union of Pure and Applied Chemistry (IUPAC)*):

A = adenine
C = cytosine
G = guanine
T = thymine
U = uracil
R = G A (purine)
Y = T C (pyrimidine)
K = G T (keto)
M = A C (amino)
S = G C
W = A T
B = G T C
D = G A T
H = A C T
V = G C A
N = A G C T (any)

3.1.3.5 Sequence search methods

Just like we can think of a sequence as a series of letters, although we know it is actually a series of chemical residues, the computer works with an abstraction where these letters are considered *symbols*. Under this assumption, the computer may apply a body of methods, known as *algorithms* to analyze these symbols.

Computer algorithms

Well known problems may be described in such a way as to have a formal solution that is guaranteed to provide a correct answer in a limited amount of time. However, overly complex problems may take too long to analyze using formal methods, to the point of being unpractical in real life. In these cases, approximate methods must be used to get *approximate* solutions which *do not guarantee to provide a meaningful answer, can mislead to incorrect conclusions or even not be able to find an existing answer at all*. Finally, there are those problems that are not yet totally known, where we don't have the knowledge or ability to define them, and which therefore can only be analyzed partially, to the extent of our current knowledge by any means. **The computer is no magic device**, *it can not find answers to problems we do not even know they exist, it can not answer questions without all the required data, it can not produce an answer out of the void*. It can however be taught how to learn, and be provided with senses we don't have to learn from existing data about things we can't perceive and tell us about them. But that's an altogether different story.

Comparing sequences

Applied to our problem, we want to know if our sequence has any resemblance to something already known. Or any significant difference from anything already known. This knowledge we can use as an additional evidence of the presence of some role, structure or function in our unknown sequence, or to discard its existence as well.

Up to this point the problem seems easy. It's when we consider the biological aspects that things get complicated: for one, we look for resemblances since we know related sequences are similar, probably derived from common roots, but we do not know for sure nor can formally express all the methods involved in sequence evolution. In other words, we do not know for sure everything that is involved in evolution, even those events we know, are sometimes very difficult to formalize, and even those we can express are complex enough to make their exhaustive analysis unfeasible and unreliable.

Search methods

First, there are situations where we are interested in such a simple query that it may be easily answered. One example is **looking for an exact match** (no mutations, no insertions/deletions, just plain coincidence) among a query sequence and anything on the database, like when we look for the presence of an oligonucleotide in a sequence database. This is one of those easy problems "satisfaction guaranteed".

A second class of problems can be easily formalized: when we look for **similarity allowing for mismatches** we may resort to well known and efficient methods that will take longer but will nevertheless find a correct answer, and even rank matches by a useful score.

Another class of problems may require analyzing too many possibilities and result in long computing times. One such example is **allowing for insertions/deletions** in the sequence. There are methods that will find and rank matches accommodating ambiguity and gaps in the comparison, but you will usually find these slow and unpractical unless you use a small search set or a specialized computer. In these cases you will most often be interested in approximate solutions, which although not guaranteeing to find the best matches work well enough in the majority of cases and are much faster allowing you an efficient work.

Finally, there are problems that are unfeasible to analyze using current technologies. An example is **allowing for translocations, inversions and other complex evolutive events**. These lead to an exponential explosion of possibilities, and are impractical to analyze even for only two sequences, much more for a high number of sequences as in a database. There is no way you can currently consider these possibilities, but yet, an approximate answer may be everything you need in most situations.

Which method should you use?

Since the most advanced methods can accommodate the simpler queries, you may reduce your choice to just two sets of options to simplify your life:

Formal methods that allow for complex queries including exact matches, search with mismatches, ambiguity and insertion/deletion events. These are normally based in one of two algorithms: **Needleman-Wunsch** or **Smith and Waterman** as implemented in MPSRCH or BLITZ. They are more reliable, but will take longer to execute unless you use specialized hardware or search a short database.

Approximated methods, that still allow for exact matches, mismatches, ambiguity and gaps, but are not guaranteed to find the best comparisons and may lose some significant matches. Their results will require more careful analysis and consideration before you accept them, but will be significantly faster and often yield good enough results. The most popular ones are FASTA and BLAST.

Since searching databases requires a high number of comparisons, and you probably won't have access to specialized hardware (like a massively parallel computer, a Field-Programmable Gate Array or a Bioccelerator) you will normally be left with **just one choice**: using the faster, approximate methods of FASTA and/or BLAST.

3.1.3.5.1 BLAST

BLAST is actually a family of programs performing various kinds of sequence searches. In some implementations, like that of GCG, there is a single command to be used, and this in turn deduces which actual command from the suite should be run considering the query sequence and the database selected.

The search

To speed things up, BLAST uses the database in a special, compressed format that takes less time to be read. While this makes it run faster, it also imposes an important limitation: **using BLAST you can only search against the full database**, i.e. you can not select a significant subset to further bound your search or avoid unwanted matches.

To compare the sequences, BLAST takes several residues at once (this is called a **word**). For each such "word" in both sequences, BLAST will compare them and assign that punctual match a score. This score is computed considering the degree of similarity or interchangeability among residues in each position (hence, a change I->L will contribute more than a change I->W to the global score of the word). In other words, **BLAST allows for mismatches and ambiguity in comparisons**.

BLAST then tries to join words and find the maximal segment of contiguous matching words. This is called a *maximal segment pair (MSP)* and represent a matching region **containing no gaps**. The scores of each word are added and a global score for the MSP is computed. BLAST will deal with each of these regions separately, i.e. **BLAST does not allow gaps inside matches**. This shouldn't be a problem since all matching regions will be analyzed, although they will be listed as separate items in the final report.

As it finds matches, BLAST takes decisions on how to align and consider them based on an statistical analysis of the sequences, discarding what it believes to be possibly a random or meaningless match. This speeds things up but has a side effect too: **you may lose matches on short or frequently appearing sequences**.

To avoid losing these matches you need to tell BLAST to use different statistical constrains. By default BLAST uses a *expectancy* value of 10. Decreasing this value will find more sequences, possibly uncovering biologically significant matches that might not be statistically significant enough.

An simple example will clarify this: by default BLAST will discard regions that do not contain at least 11 perfectly matching nucleotides, which is stringent enough in most cases, but might be a problem with lowly conserved sequences. For this reason it is usually advised to use FASTA for nucleotide searches or either reduce BLAST's word size to 6 or 7 nucleotides.

There is a second filter you may active during the search: even if the match looks like statistically unfrequent, **BLAST may reject repetitive sections or low complexity regions** to avoid what *might possibly* be biologically irrelevant matches. To activate this behaviour you need to explicitly ask BLAST about it with a command line option (-filter=x to filter short repeats and/or -filter=s for low complexity regions).

The search results

When the search has finished you will be presented with a report of the results. Each single matching region (scoring above the significance threshold) will have been evaluated, scored and ranked by its score. The output report will list the most significant matches for your commodity:

First comes an **introduction** listing the query sequence and database used, as well as some general information. This helps identify the file contents later on and reminds you of what it is all about.

```

BLASTP 2.0.5 [May-5-1998]                                BLAST HEADER

Reference: Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer,
Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997),
"Gapped BLAST and PSI-BLAST: a new generation of protein database search
programs", Nucleic Acids Res. 25:3389-3402.

Query= /usr/people/guest/jruser/work/seq/pep/atp6_rat.seq   Query sequence
      (226 letters)

Database: pir
      116,738 sequences; 37,460,341 total letters           Search set

Searching. . . . .done
  
```

```

Sequences producing significant alignments:                Score   E
                                                         (bits)  Value ..
  
```

Next thing is a **listing of sequences** containing matching regions. For each of them you get the score of the matches and the number of regions found. Since the number of sequences containing matches may possibly be too high, only a limited number of sequences is shown (250 by default, but you may change that) after sorting them out by the *score of the best match* contained and selecting the higher ranking ones. The listing will contain the sequence name, this score of the best match, the probability of finding a similar group of scores by chance and the number of matches contained by this sequence.

		Score	E	BLAST
Sequences producing significant alignments:		(bits)	Value	LISTING
PIR2:S04752	Begin: 1 End: 226			Sequence matched
!H+-transporting ATP synthase (EC 3.6.1.34) protein ...		377	e-104	Definition and scores
PIR1:PWMS6	Begin: 1 End: 226			
!H+-transporting ATP synthase (EC 3.6.1.34) protein 6...		359	3e-99	
PIR2:B25188	Begin: 1 End: 226			
!H+-transporting ATP synthase (EC 3.6.1.34) protein ...		326	4e-89	
PIR2:S26156	Begin: 1 End: 226			
!H+-transporting ATP synthase (EC 3.6.1.34) protein ...		321	1e-87	Probability of match being due to chance
PIR2:S41840	Begin: 1 End: 226			
!H+-transporting ATP synthase (EC 3.6.1.34) protein ...		320	2e-87	
PIR1:PW806	Begin: 1 End: 226			
!H+-transporting ATP synthase (EC 3.6.1.34) protein 6...		304	1e-82	
PIR1:PW806	Begin: 1 End: 226			
!H+-transporting ATP synthase (EC 3.6.1.34) protein 6...		294	9e-80	← 9 x 10 ⁻⁸⁰

Since scores are computed by statistical significance, which is *supposed* to be associated to biological significance, but need not be so, the program provides additional information to allow you make a decision on the validity of its findings: **the best scoring matches are shown aligned with your query sequence**. You should **never trust blindly the computer**, only a human expert can reliably assert possible biological meaningfulness of sequence alignments. That means *you!*

BLAST ALIGNMENT

```

\\End of List
>PIR2:S04752 H+-transporting ATP synthase (EC 3.6.1.34) protein 6 -
    rat mitochondrion (SGC1)           Details about matched sequence
    Length = 226

Score = 377 bits (958), Expect = e-104           Alignment scores
Identities = 193/226 (85%), Positives = 193/226 (85%)

Query: 1  MNENLFASFITPTMMGLPIVVTIIMFPSILFPSSERLISNRLHSFQHWXXXXXXXXXXXXX 60
          MNENLFASFITPTMMGLPIVVTIIMFPSILFPSSERLISNRLHSFQHW ← Consensus
Sbjct: 1  MNENLFASFITPTMMGLPIVVTIIMFPSILFPSSERLISNRLHSFQHWLIKLIKQMLLI 60
          HTPKGRTWALMIVSLIMFIGSTNLLGLLPHTFTPTTQLSMDLSMAIPLWAGAVILGFRHK 120
          HTPKGRTWALMIVSLIMFIGSTNLLGLLPHTFTPTTQLSMDLSMAIPLWAGAVILGFRHK
Sbjct: 61 HTPKGRTWALMIVSLIMFIGSTNLLGLLPHTFTPTTQLSMDLSMAIPLWAGAVILGFRHK 120
          HTPKGRTWALMIVSLIMFIGSTNLLGLLPHTFTPTTQLSMDLSMAIPLWAGAVILGFRHK 120
  
```

← Query seq.
← Consensus
← Matched seq.

Sbjct: 206 -PLMFPINLAGEFAKPTNISIRLFGNMFAGMVILGLLYKAAPVLI 249

BLAST SUMMARY

```

Database: pir
Posted date: May 25, 1999 5:23 PM
Number of letters in database: 37,460,341
Number of sequences in database: 116,738
Search set

Lambda      K      H
0.328      0.141  0.425
Parameters used

Gapped
Lambda      K      H
0.270      0.0470 0.230

Matrix: BLOSUM62
Gap Penalties: Existence: 11, Extension: 1
Comparison Matrix Used
Number of Hits to DB: 13093768
Observed results
Number of Sequences: 116738
Number of extensions: 433217
HSP: High Scoring Segment Pair
Number of successful extensions: 1497
Number of sequences better than 10: 98
Number of HSP's better than 10.0 without gapping: 85
Number of HSP's successfully gapped in prelim test: 13
Number of HSP's that attempted gapping in prelim test: 1312
Number of HSP's gapped (non-prelim): 105
length of query: 226
length of database: 37460341
effective HSP length: 48
effective length of query: 178
effective length of database: 31856917
effective search space: 5670531226
T: 11
A: 40
X1: 15 ( 7.1 bits)
X2: 38 (14.8 bits)
X3: 64 (24.9 bits)
S1: 40 (21.8 bits)
S2: 64 (29.3 bits)
  
```


BLAST Programs

BLASTP: Compares a protein query sequence against a protein database, allowing for gaps

BLASTN: Compares a DNA query sequence against a DNA database, allowing for gaps

BLASTX: Compares a DNA query sequence, translated into all six reading frames, against a protein database, allowing for gaps

TBLASTN: Compares a protein query sequence against a DNA database, translated into all six reading frames, allowing for gaps

TBLASTX: Compares a DNA query sequence, translated into all six reading frames, against a DNA sequence database, translated into all six reading frames. TBLASTX does not allow for gaps.

3.1.3.5.2 FASTA

FASTA also refers to a suite of programs, each intended for a given combination of query and database data types to be searched. It is based on a method developed by Pearson and Lipman and works on a different set of assumptions than BLAST and hence will provide different results. There will be occasions where FASTA will perform better or find matches that would not be apparent with BLAST, and situations where BLAST will perform better.

One first difference in some implementations (like GCG) is that you are allowed to select database subsets to search, hence speeding up the search, reducing the number of comparisons and restricting the screening to what *you know* to be actually relevant. This ability of previously reducing the search space often proves rather useful to remove worthless matches or get more easily to the matches you want.

The search method

FASTA speeds things up by comparing several residues at once. It **looks for exact matches** of this small number of residues (word). This is a first difference with BLAST: **FASTA does not consider ambiguity or approximate matches** in the comparison. Even if your sequence contains ambiguity codes, they are converted to what the program considers to be the most probable sequence before comparison.

Once all word matches have been found, FASTA tries to join them into regions, and rescores them again, this time allowing for conservative changes and matches smaller than a single word. This way it finds all matching regions just like BLAST did.

Next, FASTA introduces an important novelty: for each sequence it takes the 10 best matching regions and tries to join them into a bigger one even although they might be separated: **FASTA selects the similarity region accommodating gaps**, and computes an overall score for the match with the gaps.

Finally, FASTA sorts sequences by the best similarity region (after joining matches with gaps) found and generates a better quality alignment using the Smith & Waterman algorithm to calculate a new and more accurate score. If this score exceeds a given threshold depending on its length, the sequence is considered an acceptable match. This means that just like BLAST, **FASTA may reject some possibly biologically significant matches with a low statistical score** unless told not to apply this filter (with -noopt) or to use a different cutoff (with -opt=number).

To summarize, FASTA computes various scores:

- **init1** is the score of the best single similarity region

- **initn** is the sum of scores after extending this region with gaps as much as possible with additional, colateral, non-overlapping similarity regions
- **opt** is the score obtained after realigning the extended (gapped) similarity region using a better algorithm
- **z-score** is a linear regression estimate of the likelihood that this match might have appeared by chance

A recent improvement to FASTA also rejects those sequences whose z-score (statistical likelihood) exceeds a certain expectancy level. This is intended to further refine the results, but again, may result in the lose of some significant matches in low complexity or very short sequences/matches/motifs. You may select the expectancy level cutoff with a command line option (-exp=number).

Analyzing results

When FASTA finishes its work, it produces a report detailing its findings. The report contains various sections, each giving additional details on the search results.

The **heading** of the report tells which sequences and databases have been used for the search, and which parameters were selected to restrict FASTA's findings.

```
(Peptide) FASTA of: atp6_rat.seq from: 1 to: 226 January 25, 1999 18:04
ID ATP6_RAT STANDARD; PRT; 226 AA.
AC P05504;
DT 01-NOV-1988 (REL. 09, CREATED)
DT 01-OCT-1989 (REL. 12, LAST SEQUENCE UPDATE)
DT 01-OCT-1996 (REL. 34, LAST ANNOTATION UPDATE)
DE ATP SYNTHASE A CHAIN (EC 3.6.1.34) (PROTEIN 6)....

TO: SwissProt:* Sequences: 74,019 Symbols: 26,840,295 Word Size: 2

Databases searched:
SWISS-PROT, Release 36.0, Released on 21 Jul 1998, Formatted on 7 Oct 1998

Scoring matrix: GenRunData.blosum50.cmp
Variable pamfactor used
Gap creation penalty: 12 Gap extension penalty: 2
```

FASTA
HEADER

Query
sequence

Search
set

Search
parameters

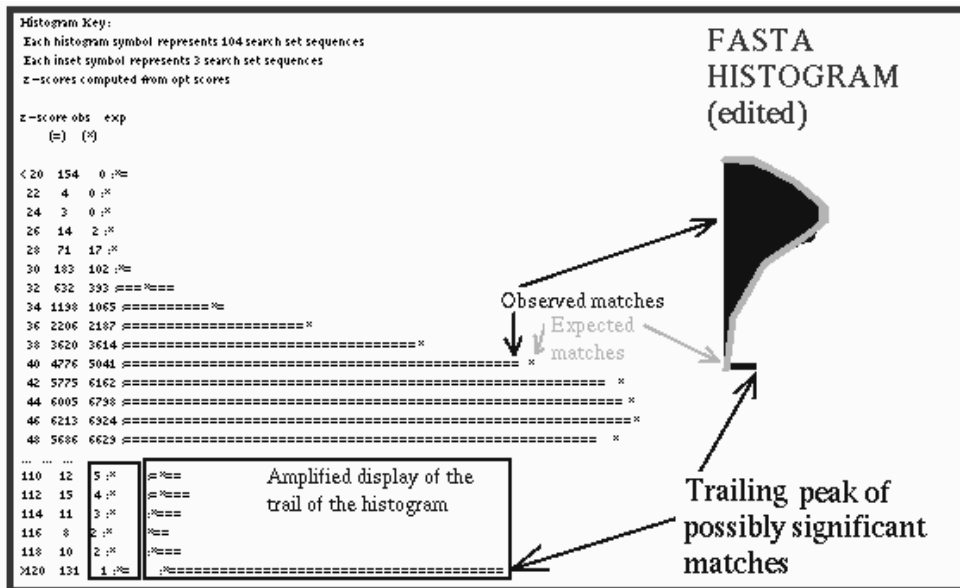
Histogram Key:

Each histogram symbol represents 104 search set sequences

Each inset symbol represents 3 search set sequences

z-scores computed from opt scores

After the introductory header comes a **histogram** of z-scores. It shows for each possible value of z-score how many sequences would have been expected to show it by chance with this dataset (marked with asterisks '*') and how many have been actually observed (marked with equal signs '='). This allows you to estimate the quality of the results. Normally you will be interested in sequences having a high z-score that is very unlikely to appear by chance (i.e. sequences with scores appearing in the trailing queue of the histogram). If the histogram doesn't have such a queue you may consider the results less reliable.



Next comes a listing of sequences that matched your query sorted by their scores. The listing includes both `init1` and `initn`, i.e. the scores of the biggest region with and without gaps, and is limited to a selected number of sequences.

Results sorted and z-values calculated from opt score
 1981 scores saved that exceeded 80
 57015 optimizations performed
 Joining threshold: 36, optimization threshold: 24, opt. width: 16

FASTA LISTING

The best scores are:

	init1	initn	opt	z-sc	E(73708)	
SW:ATP6_RAT Begin: 1 End: 226						Sequence matched
! P05504 rattus norvegicus (rat). atp...	1432	1432	1432	1921.4	0	Definition and scores
SW:ATP6_MOUSE Begin: 1 End: 226						
! P00848 mus musculus (mouse). atp sy...	1375	1375	1375	1845.3	0	
SW:ATP6_CRIGR Begin: 1 End: 226						
! P14413 cricetulus griseus (chinese ...	1244	1244	1244	1670.3	0	
SW:ATP6_EQURS Begin: 1 End: 226						
! P92480 equus asinus (donkey). atp s...	1220	1220	1220	1638.3	0	Probability of match due to chance

Score of best matching region Score of longest extended match (with gaps) Score of enhanced extended alignment Likelihood estimation

Finally you get a display of all the sequences selected aligned with your sequence. These are the high quality, Smith & Waterman alignments obtained in the last refinement step of the search, and normally will only include the matching region plus a short piece of adjacent sequence unless you have explicitly asked for full alignments on the command line (with `-showall`). Looking at these alignments you can further attest the biological significance of each matching sequence found.

```
! P52137 escherichia coli. putative a... 44 44 82 115.0 9.9
\\End of List
```

atp6_rat.seq		Query sequence		FASTA LISTING OF SEQUENCE ALIGNMENTS				
SW:ATP6_RAT		Matched sequence						
ID	ATP6_RAT	STANDARD;	PRT;	226 AA.				
AC	P05504;							
DT	01-NOV-1988 (REL. 09, CREATED)		Details of matched					
DT	01-OCT-1989 (REL. 12, LAST SEQUENCE UPDATE)		sequence					
DT	01-OCT-1996 (REL. 34, LAST ANNOTATION UPDATE)							
DE	ATP SYNTHASE A CHAIN (EC 3.6.1.34) (PROTEIN 6)							
SCORES Init1: 1432 Initn: 1432 Opt: 1432 z-score: 1921.4 E(): 0 Scores Smith-Waterman score: 1432; 100.0% identity in 226 aa overlap								
		10	20	30	40	50	60	
atp6_rat.seq	MNENLFASFITPTMMGLPIVVVTIIMFSPILFPPSSERLISHRLHSFQHWLIKLIKQMMLI	Query sequence						
ATP6_RAT	MNENLFASFITPTMMGLPIVVVTIIMFSPILFPPSSERLISHRLHSFQHWLIKLIKQMMLI	Matched sequence						
		70	80	90	100	110	120	
atp6_rat.seq	HTPKGRTWALMIIVSLIMFIGSTNLLGLLPHTFTPTTQLSMDLSMAIPLWAGAVILGFRHK							
ATP6_RAT	HTPKGRTWALMIIVSLIMFIGSTNLLGLLPHTFTPTTQLSMDLSMAIPLWAGAVILGFRHK							
		70	80	90	100	110	120	

FASTX, FASTY – compares a query DNA sequence to a protein sequence database, translating the DNA sequence in all six reading frames and allowing frameshifts.

TFASTX, TFASTY – Compares a protein sequence to a DNA sequence or DNA sequence library, such that the DNA sequence is translated in all six reading frames, and the protein query sequence is compared to each of the six derived protein sequences. The DNA sequence is translated from one end to the other; termination codons are translated into unknown amino acids.

LALIGN, LFASTA – Same as the FASTA program, except that multiple aligning regions may be reported for each sequence.

PLALIGN – dot plot algorithm available through the fasta suite

FAST-pat, FAST-swap: compares a sequence to a pattern database

3.1.3.6 Searching Swiss-Prot

Here is an example of how to search the Swiss-Prot database from start to finish. We suggest that, if possible, you open another window in your web browser so that you can follow along with the instructions while you are doing the search.

First we need a DNA sequence. Let's use this sequence:

```
ttctaactgc aacctttcga agcctttgct ctggcacaac aggtagtagg
cgacactgttctgtgtgtca acatgaccaa caagtgctct ctccaaattg ctctctgtt
gtgcttctccactacagctc ttcc
```

1. Now we need to translate the nucleotide sequence into a protein sequence. There is a handy tool on the WWW for automatic translation of nucleotide sequences into amino acid sequences. Copy the DNA sequence above and we will use this tool to translate that sequence.

2. You are offered a choice of translation tables. For now, we'll just use "Standard." Later, when you have a sequence from an organism that you've been working with, feel free to choose another. Now paste the DNA sequence into the input box. And then click on "**Generate Protein.**"

3. The output will now show your *translated* and then *transcribed* protein as your output. Select and copy your translated protein sequence.

4. Now go back to the main Tools page on the Swiss-Prot server and click on Blitz. **Blitz** is a search tool that allows you to search the databases for protein sequence.

(**Note:** you may also use BLAST for this. However, information about Blast searching is already available on our nucleic acid sequence page.)

(**Also note:** Blitz searching requires access to an email account. If you do not have access to an email account, skip down to our instructions for performing a BioSCAN similarity search with your protein sequence.)

5. Once you are in the Blitz entry screen, you will be asked to submit your email address and a title for your search (the title is for your own record keeping). You should end up with 2 email messages, the first being a confirmation of your job's submission and the second being the output of your job. Your output will contain dozens of proteins that resemble the sequence that you've given. At the bottom of your output will be the name and accession number of the protein that most closely resembles the sequence that you've submitted.

6. Congratulations! Your most closely resembled output should read:

```
RESULT 1   Score 104; Match 0.0%; Predicted No. 5.55e-17;
ID      INB_HUMAN          STANDARD;          PRT;          187 AA.
DE INTERFERON BETA PRECURSOR (FIBROBLAST).
```

(Note: there will be many results, but the first result is the most closely-related result.)

7. BioSCAN

Alternate instructions for similarity searches using BioSCAN: Go to the BioSCAN input page and paste your copied protein sequence in the input area. All of the other settings should default to normal (which is all we need for now), so now choose "**Submit Request.**"

8. Congratulations! Your results should read

9. Db|Acc|Name Description

10. sw|P01574|INB_HUMAN INTERFERON BETA PRECURSOR (FIBROBLAST)...

This means that your sequence most closely matched this precursor sequence in humans. All of the entries in BioSCAN are generated in hypertext, so you can easily find more information about your results.

3.1.3.7 Searching PIR

The most difficult part of searching the PIR database is finding the search engine on the WWW. Although many sites have been put together to search multiple databases at once, very few concentrate specifically on PIR. As a matter of fact, there is not even a link to the WWW search site from the PIR home page.

Because PIR is really a series of databases covering different topics and differing amounts of information, the PIR WWW search engine allows the user to select which database to search. However, because the engine is formatted to be a specific information retrieval site, you cannot search on a protein sequence. Rather, you may search the database to retrieve protein sequences of specific characteristics that you determine.

PIR is a good starting point to find information about sequence availability and detection. It is a very thoroughly cross-linked report tool that will help you get to more detailed information that you may be seeking. However, if you need to identify a protein sequence that you do not know the history of, you should try the search engines that are affiliated with Swiss-Prot.

3.1.3.8 Searching sequence databases over the internet

Protein Identification using MS-Fit

Due to the rapid growth of gene and protein databases, peptide fingerprint mass mapping is widely used for initial identification of proteins separated by gel electrophoresis. This method involves the enzymatic in-gel digestion of proteins to generate peptides, followed by mass measurement of the cleaved peptides. The experimental mass values are then compared with theoretical values from protein databases, calculated using the cleavage specificity of the enzyme. By using an appropriate scoring algorithm, the closest match or matches can be identified.

With delayed extraction in reflectron mode, the mass resolution (FWHM) can reach more than 10,000 for analytes that have mass-to-charge ratios of 1000 ~ 6000 kDa, and mass accuracies in the parts per million range can be achieved even at low levels of gel-separated proteins (Jensen *et al.*, 1996). Also, nanoESI mass spectrometers have typical resolutions higher than 10,000.

The advantage of the improved mass accuracy is the increased "search specificity" of each peptide by up to two orders of magnitude, leading to a much better discrimination of true hits against false positive search results. This increased search specificity even allows for the analysis of simple protein mixtures (Jensen *et al.*, 1996). Finally, peptide mass mapping is not only useful for identification of proteins existing in the databases, but also for cross-species identification. This approach relies on sequence conservation between two proteolytic cleavage sites (a stretch of ~ 10-15 amino acids) in homologous proteins of different organisms (Yates, 1998).

Peptide mass mapping using MS-Fit

(<http://prospector.ucsf.edu>)

First, sort out all specific peaks from non-specific ones (trypsin-autolysis products, human keratin contaminants). Series of peaks with constant mass differences can be attributed to polymers, which may come from tubes, or be added during preparation procedure (Jungblut and Thiede, 1997).

If in a database hit, some major peaks are left unassignable, perform a second database search using the unmatched masses, since more than one protein may be present in the gel spot due to co-migration during electrophoresis (Jensen *et al.*, 1996).

Also the success of peptide mass mapping can be compromised by errors in the sequence database, and the observation of too few peptides (less than 5 or so) in the MS map of a given protein. In those cases, partial amino acid sequence determined using post-source decay analysis or tandem mass spectrometry are needed to confirm a protein identification by MS-Fit.

1. Input the peptide masses obtained from mass spectrometer into the program. Monoisotopic peptide masses from MALDI-TOF are preferred. Peptide masses of multiply charged ions (M/z) from electrospray mass spectrometer along with charge state (z) are submitted at the same time.

It is advised to input these peptide masses into any spread sheet, then copy and paste the values to the blank in the program. Average masses may also be used for the search, however, more false positive results will be obtained due to low mass accuracy.

2. Set mass tolerance, which should be no less than the mass accuracy obtained experimentally using mass spectrometry. Usually, 100~300 ppm is used for MALDI-DE-TOF, and 100 ppm for electrospray orthogonal TOF when external calibration is used. 5~20 ppm can be used using internal calibration. 0.5 Da is set when regular MALDI-TOF is used.

Measuring masses as accurately as possible is the single most important thing one can do to achieve the highest certainty of protein identification in a peptide mass fingerprinting experiment. The more accurate the masses measured, the lower mass tolerance set, the more unambiguous results obtained.

3. Choose a minimum number of peptides required for matching a particular protein in the database to generate a hit. The default value is 4, which can be changed depending on your search output and the total number of peptides submitted. If the search result contains too many hits, increase this number to eliminate the number of false positive results.

4. Choose a protein database to start the search. Users can choose any databases listed in the program, such as NCBI nr, Genpept, Swiss Prot, Owl and dbEST. We suggest to use NCBI nr database as the first choice for the search, because NCBI nr combines most of the public domain protein databases (dbEST database not included) together to generate the largest non-redundant protein database, which has been updated most frequently. Swiss Prot may also be used since it is the best annotated database.

When there is no match in the protein databases, this may suggest that the gene encoding the protein of interest has not yet been cloned. However, it may be known as an expressed sequence tag (EST) that contains part of the protein. Therefore, it may be useful to screen a dbEST database. However, search times will typically be longer because of the multi-frame translations combined with the fact that the dbEST file is >3x larger than the NCBI nr file. When an EST database is selected, the number for DNA frame translation needs to be set. A user should select frame mode 6 unless it is known that the database being searched contains sequences exclusively cloned in one direction or contains known genes with sequences already in frame. However, it takes longer search time when you choose frame mode 6 instead of 3. Users can generate their own database for searching, the instructions are in our home page.

5. Select the species (optional). If you know the source of the protein of interest, you may perform a species-limited search. Alternatively, the species can be designated as "all".

In general, it is useful to designate the species as "all", because proteins in different species with high homology can be identified when proteins of all species are included in the search. Moreover, the species pre-filtering is imperfect because of the poor usage of taxonomy (standard species naming conventions) in the databases. If you don't know the Latin taxonomic name for the species of interest, see: NCBI Taxonomy Browser.

6. Select the enzyme specificity and the amount of missed cleavage sites. MS-Fit program will search the submitted peptide masses against calculated masses of peptides cleaved by the specific enzyme of choice. The termini of the matched peptides will be consistent with the cleavage specificity of the enzyme used to generate the peptide. Increasing the number of maximum number of missed cleavages allowed, enables matching to sequences with uncleaved sites internal to the peptide.

If the digestion time is fairly long (>6hrs), it is possible to find peptides derived from non-specific cleavages. The option for the non-existing enzyme Slymotrypsin was created as a means for allowing Chymotryptic cleavages in tryptic digests. When using this choice it is important to increase the missed cleavages allowed to 3 or above. It is possible to combine the rules for two or more enzymes by adding options to the Enzyme item on the HTML form.

7. Select the molecular weight and pI range (optional). First, perform the search with no molecular weight and pI restriction. Adjust the range according to the search output.

Generally, the molecular weights from SDS-PAGE are not accurate, therefore, the range needs to be set as MW +/-10 kDa. Moreover, if the analyzed protein is a degradation product of another larger protein, the molecular weight estimated from the gel would not be useful for the search.

8. Choose the types of modified residues. Users may also specify desired modifications for cysteine residues, thus generating theoretical masses to match those cysteine containing peptides which have been reduced, then alkylated with iodoacetic acid (carboxymethylation), iodoacetamide (carbamidomethylation) or 4-vinylpyridine (pyridylethylation). Some modifications of proteins can be generated during polyacrylamide gel electrophoresis, *i.e.*, oxidation of methionine and acrylamide-modified cysteine residues. The program can also show peptide mass changes due to known postranslational modifications, namely, N-terminal glutamine to pyro-glutamic acid, and phosphorylation of serine, threonine and tyrosine, when these modifications have been specified. Furthermore, for any database entry with a methionine at the N-terminus, the N-terminal peptide is considered either in its original form or in a form where the methionine is removed and the next amino acid is acetylated. Moreover, if the database curators have removed the N-terminal methionine from the sequence, then MS-Fit will not apply the acetylation modification.

9. Searching result Two scoring system are shown here, MOWSE Score[7] and MS-Fit rank. The higher the score, the higher number of peptide masses matched and the more likely the "hit" is. In the output, the number of peptide masses matched, corresponding sequence, protein molecular weight, pI, accession number, species and protein name are listed. Clicking on the MS-digest index number displays the coverage from the matched peptide of certain. Clicking on the accession number links the NCBI Entrez.

Currently, MS-Fit has only a simple ranking system. The results are sorted so that if multiple database entries are matched, sequences are more likely listed higher in the rank. All database entries matching the input data and parameters are ranked on the following basis: 1) Database entries with the least number of unmatched masses are ranked higher; 2) Among equivalent match (those with the same rank) the results are sorted in order of increasing index number.

For an unambigious protein idenitification, in our opinion, it is necessary to have a partial peptide sequence to confirm the MS-Fit search result, unless you know what the protein could possibly be from other experiments, or the peptide masses obtained with

high mass accuracy (<10ppm) and more than 60% peptide masses submitted are matched.

10. Alternative search: Homology mode

Selecting the box for that allows for a single amino acid substitution per peptide puts MS-Fit into homology mode searching. In this mode, the MS-Fit option for possible modifications are bypassed. In practice, the homology mode should only be used when one or more of the following conditions applies: the peptide mass data have excellent mass accuracy (+/- 10 ppm or better); a narrow intact protein MW filter is used; the hits will be saved and searched via MS-Tag. Min. # matches with NO amino acid substitution controls the particular protein entry in the database for homology search. Database sequences passing the homology widened peptide mass filter are then passed through a mutation matrix to try and find a single amino acid substitution which would transform the calculated mass of the database sequence to the experimentally determined mass. The output displays the necessary substitution and the corresponding sequence consistent with the experimental peptide mass data (not the sequence present in the database).

11. Other features. One ProteinProspector search program can serve as a pre-filter for another search program. To accomplish this, the Hits (index numbers for matching database entries) from the first program are saved to a user-specified file. This file is then retrieved by the second program, and only those matching database entries are searched by the second program. Saved hits from MS-Fit can be used by other programs such as MS-Tag and MS-Edman.

Summary:

Sequence database similarity searching is one of the most common computing techniques in modern biology. The most obvious first stage in the analysis of any new sequence is to perform comparisons with sequence databases to find homologues. There are many methods for sequence searching. By far the most well known are the BLAST suite of programs. The two types of sequence identification numbers, **GI** and **VERSION**, have different formats and were implemented at different points in time. Sequence formats are simply the way in which the amino acid or DNA sequence is recorded in a computer file.

Model Questions:

1. Briefly explain the sequence databases search and the possible information from them?
2. Write about the different sequence formats and search methods?

References:

1. Introduction to Bioinformatics by Arthur M. Lesk.
2. Shevchenko, A, Wilm, M. and Mann, M. 1997. Peptide Sequencing by Mass Spectrometry for Homology searches and Cloning of Genes. *J. of Prot. Chem.* 16(5):481-490.
3. Bioinformatics - Sequence and Genome Analysis- David W. Mount.
4. Developing Bioinformatics Computer Skills by Cynthia Gibas, Per Jambeck

AUTHOR:

B.M.REDDY M.Tech. (HBTI, Kanpur)

Lecturer, Centre for Biotechnology

Acharya Nagarjuna University.

Lesson 3.1.4

INFORMATION PROCESSING CHALLENGES

Objective

3.1.4.1 Introduction

3.1.4.2 Biological Data & Its Properties

3.1.4.3 Database and its concept

3.1.4.4 New Informatics Framework for Pharma Research

3.1.4.5 Intelligent Information Integration

Summary

Model Questions

References

Objective

The objective is to know the properties of biological data and need of new informatics framework for processing the biotechnology information

3.1.4.1 Introduction

The use of informatics to organize, manage, and analyze genomic data (the genetic material of an organism) has become an important element of biology and medical research. A new IT discipline— *bioinformatics*—fuses computing, mathematics, and biology to meet the many computational challenges in modern molecular biology and medical research. The two major themes in bioinformatics—data management and knowledge discovery—rely on effectively adopting techniques developed in IT for biological data, with IT scientists playing an essential role.

Information technologies produced the necessary speedup for collaborative research efforts in biology, helping genome researchers complete their projects on time. Many genomes have already been completely sequenced, and genome research has migrated from raw data generation to scientific knowledge discovery. Likewise, informatics has shifted from managing and integrating sequence databases to discovering knowledge from such biological data. Informatics' role in biological research has increased and it will certainly become increasingly important in extending our future understanding of biological life.

The basic data has so far usually been sequence data (nucleotide or protein), but other types of data (microarray, functional analysis, interactions) are now rapidly coming into focus.

Bioinformatics deals with the issues created by the massive amounts of new types of data obtained through novel biological experiments. Informatics has helped launch molecular biology into the genome era. In the 1990s, the Human Genome Project and other genome sequencing efforts generated large quantities of DNA sequence data. Informatics projects in algorithms, software, and databases were crucial in the automated assembly and analysis of the genomic data. The Internet also played a critical role: the World Wide Web let researchers throughout the world instantaneously share and access biological data captured in online community databases.

3.1.4.2 Biological Data & Its Properties

Some of the key properties of biological data are described in the paper 'Issues in incorporation semantic integrity in molecular biological object-oriented databases' by S Schweigert, P Herde & R Sibbald in *CABIOS* (now called *Bioinformatics*). The authors conclude that Biological data present, if not unique problems, a unique combination of problems with respect to ensuring semantic integrity.

Bear in mind that one of the main reasons for managing data using a DBMS is that it gives you the ability to automatically check whether (or to what extent) the data is correct - both *syntactically* correct and *semantically* correct. The properties of biological data identified below make this particularly challenging.

What is the difference between syntax and semantics? The following definitions are taken from the *Collins English Dictionary*:

syntax (*n.*) a systematic statement of the *rules* governing the properly formed formulas of a logical system.

Semantic (*adj.*) of or relating to meaning or arising from distinctions between the meanings of different words or symbols.

The difference between syntactic and semantic integrity can be illustrated by the following example: The number of legs that a human can have might be syntactically constrained to be an integer. Semantically, it might be loosely constrained as a positive integer < 5.

General properties of Biological Data

. Complexity:

Biological data are complex. They are arguably the most complex data known. Apart from mere numerical richness, e.g. many different species, different populations, etc., there are also many similar but not identical entities that are challenging to model.

Example:

We might think of encapsulating information about protein sequences as follows:
Source organism: *Homo sapiens*

Sequence: AVGHRTATGPA... (*i.e. composed of 20 naturally occurring amino acids*)

But this takes no account of the following: engineered proteins; non-standard amino-acid types; experimentally unresolved amino-acid types; ligands; post-translational modifications; mutations; etc.,

. Exceptions:

Biological data are exception-ridden. The question is how to accommodate them and maintain

the semantic checking.

Example:

What is the largest 'permissible' resolution for a protein structure solved by X-ray crystallography? PDB 1qgc has a resolution of 30Å. Is this an error? How should we handle this in a database? Exclude PDB 1qgc? Include it in the same way as any other structure?

. Missing Data :

Biological data are very often incomplete. This sometimes occurs because some biological objects are large and complete descriptions take time to obtain...It also occurs because of limited resources or because attempts to collect the data failed.

Examples:

Most of the genomes in GenBank are incomplete (many are small fragments); parts of the full GenBank sequence for a protein are frequently missing from its PDB structure; etc.

. Changing Models and Data :

Many biological concepts are incompletely understood. Our model of a given biological concept is likely to become more sophisticated over time.

Example:

It was originally assumed that a single gene produces (at most) a single product. It is now

known that the same gene can be translated in multiple ways producing multiple products. A

one-to-one mapping between gene and product is therefore an over-simplification.

A more frequent problem is that data items are modified after they have been committed to the database... If the data change, the issue of how checks are to 'propagate' over related objects arises. For example, a protein sequence is in a database and is part of a multiple sequence alignment. Alterations to the sequence may change the alignment.

. Holism:

The biological domain is highly integrated. The information tends to form nets and hierarchies with many facts having bearing on the truth of other facts. If the database can alter itself as a result of applying a semantic rule, then the result of the change must be checked. The question is, where to stop?

. Interoperability :

There is an increasing number of molecular biological data collections, and using them together is becoming a pressing issue. Interoperability is required since it is not practical to build a single monolithic database. .

Example:

The BBSRC / EPSRC Bioinformatics Initiative is funding a project to set up a subset of the data in the MSD database (a cleaned-up version of the PDB managed by an Oracle DBMS) within the Dept. of Biochemistry & Molecular Biology at DCL, and integrate it with the CATH database.

. Concept Mismatch and Nomenclature:

Biology is full of ambiguous definitions and conceptual confusion. Different branches of

biology (e.g. structural biologists, geneticists) use the same term for subtly different concepts. Many aspects of living systems are only partially understood.

Example:

The word colony is used in zoology to mean a group of animals of the same species that live together and depend on each other. In microbiology a colony is a group of microorganisms that are considered to have developed from a single parent cell.

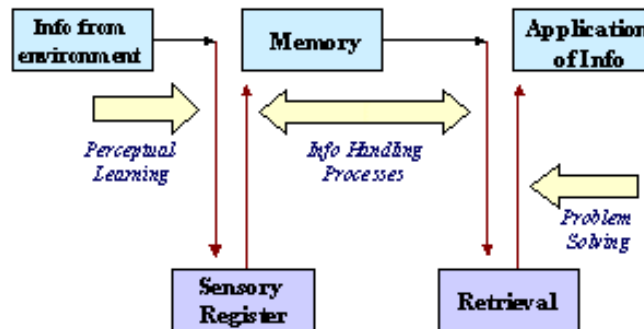
. Political Problems:

In a public data collection, there may be problems in labelling data as incorrect.

Obviously, these are general properties some or all of which may not apply to a particular

biological database application.

INFORMATION PROCESSING MODEL



Nicole A. Sage Copyright © 2000 All Rights Reserved

3.1.4.3 Database and its concept

A database is simply an organized collection of related data, typically stored on disk, and accessible by possibly many concurrent users.

A database is a collection of non-redundant data, which can be shared by different application systems. A database implies separation of physical storage from use of the data by an application program to achieve program / data independence. Using a database system, the user or programmer or application specialist need not know the details of how the data is stored and such details are "transparent to the user".

Databases are generally separated into **application** areas. For example, one database may contain Human Resource (employee and payroll) data; another may contain sales data; another may contain accounting data; and so on. Databases are managed by a **DBMS**.

The Database Concept

Was an attempt to solve the problems with file systems

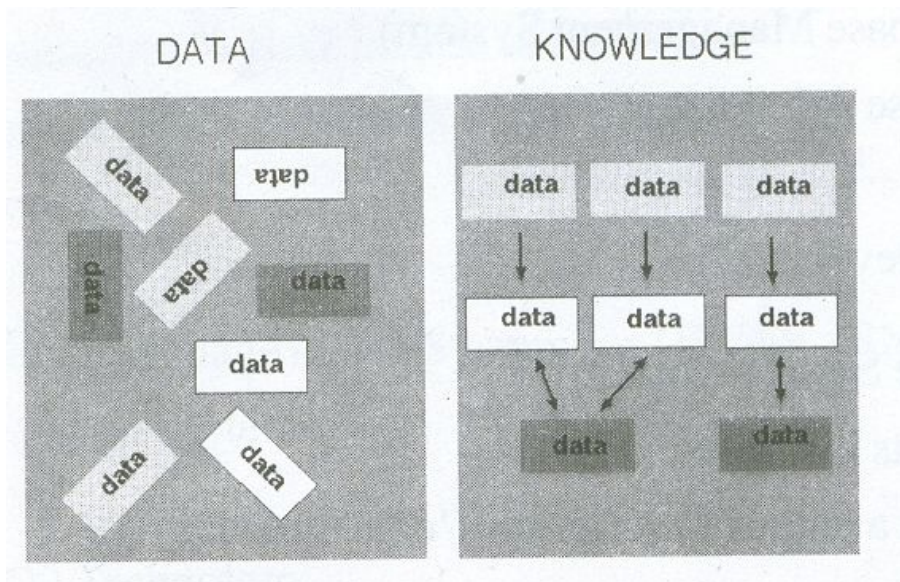
- First design the data then write the application programs
- Manage relationships between items of data independently of the application programs
- Database must also contain descriptions of the data, the schemas
- Requires a Database Management System (DBMS)

Two Primary Goals of Database

- Minimise Data Redundancy
 - Duplication of data
 - Stored same data in multiple files
- Achieve Data Independence
 - Ability to make changes in data without making changes to the programs that process the data.

Thus Databases are Needed

- . To collect and preserve data
- . To make data easy to find and search . To standardise data representation
- . To organise data into knowledge



3.1.4.4 New Informatics Framework for Pharma Research

Most pharma company's information resources are currently spread out and managed by different groups. Some of these will be made more widely available on a corporate Intranet, but others will remain only on locally accessible machines. The range of interfaces will be varied, with some having their original interfaces as supplied by the vendors or authors, and others having been incorporated into an HTML page or CGI forms based interface. In some cases, perhaps where two solutions have been bought from a single vendor, or where a special project has been undertaken, data may be shared between one or more of the application interfaces.

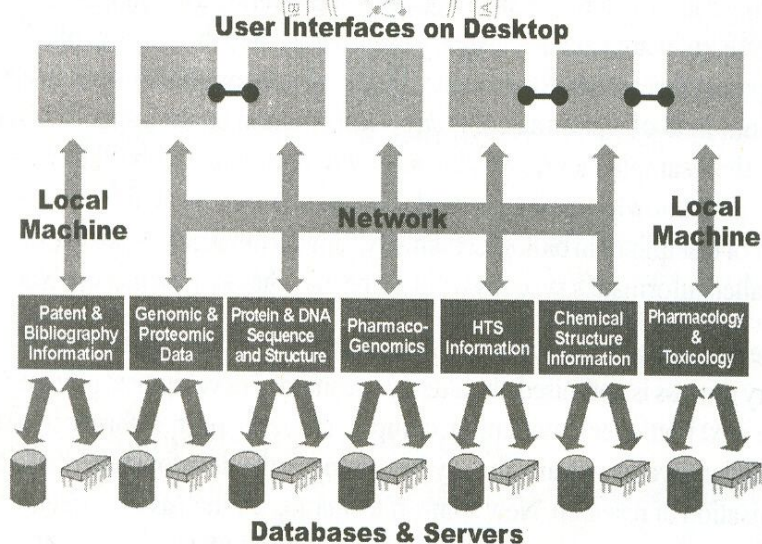
This collection of resources falls short of the system required by pharma companies. The systems are diverse and therefore complex to access and difficult to use. They share information only poorly, and can perform very slowly. Queries accessing data from multiple databases may be almost impossible to correlate as data are held in different types of databases.

To fulfill the requirements being demanded for improved R&D productivity, pharmacompanies need a new informatics framework to build around. The framework needs to support intelligent data integration and effective data dissemination. In order for any such framework to be successful in the real world it must fulfil a number of criteria which are as follows:

- It must encompass the widest possible range of information and other resources
- It must be flexible and allow new technologies and resources to be added
- It must be built on robust, scalable components
- It must seek to augment existing staff and systems, not threaten or replace them
- It must make full use of the optimised processes and systems already in

place

- It must be configurable to operate with different environments and processes
- It must be customisable to allow tailoring to meet new requirements
- It must perform as quickly as the existing resources.



These criteria cannot be met by systems that aim to suck data from existing databases into a single database environment. The information losses, optimisation problems, and required changes in expertise amongst the users alone make this type of wholesale change unlikely to succeed. In order to be able to provide the type of information infrastructure needed, a layer must be added on top of the existing systems, without disturbing the work and expertise that has gone into creating the component resources, but allowing them to be used more effectively together.

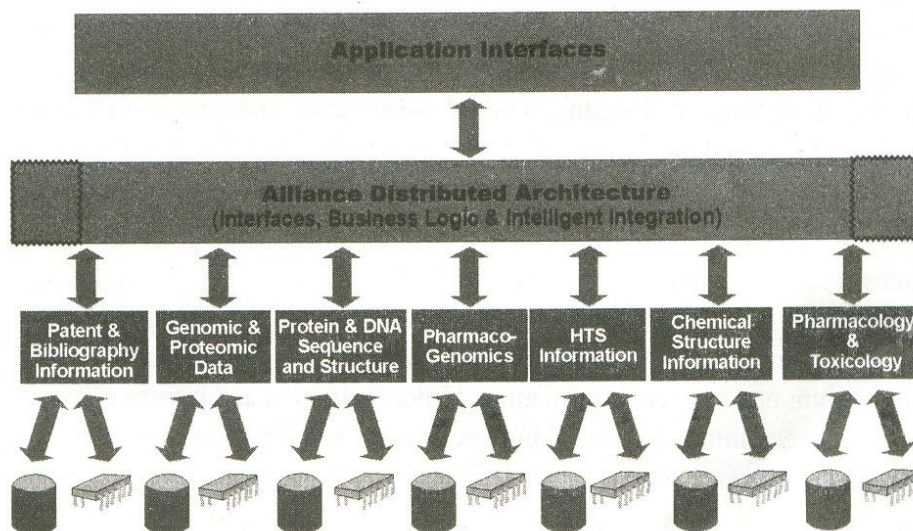
3.1.4.5 Intelligent Information Integration

There are various methods for achieving levels of integration between resources, ranging from simple anarchy where all systems are independent and have minimal integration capabilities to more cooperative systems.

The current status of most companies ranges from anarchy to detente, with a dose of dictatorship thrown in from competitive commercial systems. Recently attempts have begun to be made by the Object Management Group's Life Science Research DSIG to achieve a level of democracy by defining standards for interfaces in life sciences applications. These standard interfaces, expressed in Interface Definition Language (IDL) would define a "contract" between applications such that however they are written and whatever they do, they should support the passing of specified objects in a defined manner.

What pharcos really need is a mechanism of supporting their existing applications and their existing staff's expertise, whilst at the same time providing a robust and flexible framework for them to build upon. This is the federated model with systems

from multiple vendors being overlaid with a framework layer that handles data access, business logic and information integration.



This integration framework would have a number of components:

- Information access routines
- Information transport system
- Data cleaning and warehousing strategies
- Data mining routines
- Data presentation and analysis systems

The information access routines would provide a way of using the existing databases, files, and software programs to continue to organise, store and retrieve data in exactly the way they are already doing. Mediators would maintain a list of the available resources and the type of data stored in them, candidates there would have to be at least a corresponding thirty-fold increase in projects undertaken. This means that existing staff will be involved in multiple projects at anyone time, and managing the complexities of these interactions and interdependencies would require management tools for research staff and their resources.

The increased number of projects also implies that the projects must be shorter, and that stop/go decisions would have to be taken quickly and accurately. In order to make accurate decisions in a short space of time, all the available data must be correlated and presented coherently' to the project manager and the Head of Research. Results management tools and individual electronic notebooks capturing all relevant results would play an important role in ensuring that decisions can be taken accurately on the basis of all available information about a project.

Because all these tools and the infrastructure itself would be built of components they could be tailored or extended to react to new situations. New data sources could be incorporated, new user interfaces could be designed and built, and new automatic processes could be created, all simply by linking together the relevant components. As well as providing powerful customisation capabilities, this would also allow convenient

automation of commonly performed functions, and the capture of expert knowledge. In addition to the integration of data, data must be disseminated and presented in functional, interactive interfaces on a user's desktop.

Summary

In the last few decades, advances in molecular biology and the equipment available for research in this field have allowed the increasingly rapid sequencing of large portions of the genomes of several species. In fact, to date, several bacterial genomes, as well as those of some simple eukaryotes (e.g., *Saccharomyces cerevisiae*, or baker's yeast) have been sequenced in full. The Human Genome Project, designed to sequence all 24 of the human chromosomes, is also progressing. Popular sequence databases, such as GenBank and EMBL, have been growing at exponential rates. This deluge of information has necessitated the careful storage, organization and indexing of sequence information. Information science has been applied to biology to produce the field called Bioinformatics. Bioinformatics, an emerging area has applications and challenges in all fields of biology with special reference to Genomics, Proteomics, Drug Discovery, Pharmacogenomics and Biological sequence analysis at genome scale.

Model Questions

1. Write a note on biological data and its properties?
2. Briefly explain the information processing challenges in biotechnology/

Reference books

1. Bioinformatics, Concepts, Skills, and Applications by S.C.Rastogi & NAMita MEndiratta.
2. Bioinformatics - Sequence and Genome Analysis- David W. Mount.
3. Bioinformatics Computing- by Bryan Bergeron.
4. Introduction to Bioinformatics by Arthur M. Lesk.

AUTHOR:

B.M.REDDY M.Tech. (HBTI, Kanpur)

Lecturer, Centre for Biotechnology

Acharya Nagarjuna University.

Lesson 3.2.1

Structural Databanks

Structure

3.2.1.1 Introduction

3.2.1.2 Overview

3.2.1.3 Protein Data Bank

3.2.1.4 Molecular Modeling Database

3.2.1.5 BLAST Against PDB Sequences

3.2.1.6 VAST: Structure Similarity by 3-D Shape

3.2.1.7 Structure File Formats

3.2.1.8 Database structure Views

Summary

Model Questions

References

Objective: This chapter introduces the bioinformatics aspects of biomolecular structures, with special emphasis on the sequences that are contained in three-dimensional structures. It attempts to inform the reader about the contents of structure database records and how they are treated, and sometimes mistreated, by popular software programs.

3.2.1.1 Introduction

The imagery of protein and nucleic acid structures has become a common feature of biochemistry textbooks and research articles. This imagery can be beautiful and intriguing enough to blind us to the experimental details an image represents—the underlying biophysical methods and the effort of hard-working X-ray crystallographers and NMR spectroscopists. The data stored in structure database records represents a practical summary of the experimental data. It is not the data gathered directly by instruments, nor is it a simple mathematical transform of that data. Each structure database record carries assumptions and biases that change as the state of the art in structure determination advances.

3.2.1.2 Overview

The PDB (Protein Data Bank) is the major repository of protein structures (and to some extent of nucleic acid structures). This database stores 3-dimensional atomic coordinates of proteins and nucleic acids and the data is obtained by experimental methods like X-ray crystallography, NMR, or computer modeling.

A PDB record includes information similar to that found in the header of genbank entries (organism of origin, authors, literature references etc.) Sequence similarity tools such as BLAST can also be used as the record in database has sequence information. The entry also contains secondary structure information like location of helices and strands and disulfide bonds. The three-dimensional structure information is stored as a series of spatial coordinates for each atom in the molecule (the position of the atom on the x,y and z axes).

Since the three-dimensional atomic coordinates are not very convenient examination of the structure with the naked eye, a large number 3D structure viewers have been designed to graphically view these coordinates. The most common is RasMol (by Roger sayle) programs, it allows drawing the structure using a range of representations, including space fill, ball and stick and wire frame views familiar to chemists, ribbon diagrams and cartoons which emphasize secondary structure elements and many more features are available.

There are other structural databases like the scop (structural classification of proteins) database which classifies proteins according to structural similarity and evolutionary relationships. In scop one can also see the hierarchical classification of proteins in families and super families, with links to relevant PDB structures.

The Notion of 3-D Molecular Structure Data

Let us begin with a mental exercise in recording the 3-D data of a biopolymer. Consider how we might record, on paper, all the details and dimensions of a three-dimensional ball-and-stick model of a protein like myoglobin. One way to begin is with the sequence, which can be obtained by tracing out the backbone of the 3-D model. Beginning from the N-terminus, we identify each amino acid side chain by comparing the atomic structure of each residue with the chemical structure of the 20 common amino acids, possibly guided by an illustration of amino acid structures from a textbook.

Once the sequence has been written down, we proceed with making a two-dimensional sketch of the biopolymer with all its atoms, element symbols, and bonds, possibly taking up several pieces of paper. The same must be done for the heme ligand. After drawing its chemical structure on paper, we might record the three-dimensional data by measuring the distance of each atom in the model starting from some origin point, along some orthogonal axis system. This would provide the x -, y -, and z -axis distances to each atomic "ball" in the ball-and-stick structure.

The next step is to come up with a bookkeeping scheme to keep all the (x, y, z) coordinate information connected to the identity of each atom. The easiest approach may be to write the (x, y, z) value as a coordinate triple on the same pieces of paper used for the 2-D sketch of the biopolymer, right next to each atom.

This mental exercise helps to conceptualize what a three-dimensional structure database record ought to contain. This is an adequate "human-readable" record of the structure, but one probably would not expect a computer to digest it. The computer needs clear, parsable encoding of the associations of atoms, bonds, coordinates, residues, and molecules. Here is where the real exercise in structural bioinformatics begins.

Coordinates, Sequences, and Chemical Graphs

The most obvious data in a typical 3-D structure record, irrespective of the file format in use, is the *coordinate data*, the locations in space of the atoms of a molecule, represented by (x, y, z) triples, distances along each axis to some arbitrary origin in space. The coordinate data for each atom is attached to a list of labeling information in the structure record: which element, residue, and molecule each point in space belongs to. For biopolymers this labeling information can be derived starting with the sequence. Implicit in each sequence is considerable chemical data. We can infer the complete chemical connectivity of the biopolymer molecule directly from a sequence, including all its atoms and bonds, and we could make a sketch, just like the one described earlier, from sequence

information alone. We refer to this "sketch" of the molecule, as the *chemical graph* component of a 3D structure. Sequence is an implicit notation for the complete chemical graph of a biopolymer molecule

When we sketch all the underlying atoms and bonds representing a sequence, we may defer to a textbook showing the chemical structures of each residue, lest we forget a methyl group or two. Likewise, computers could build up a sketchlike representation of the chemical graph of a structure in memory using a *residue dictionary*, which contains a table of the atom types and bond information for each of the common amino acid and nucleic acid building blocks.

Atoms, Bonds, and Completeness

Molecular graphics visualization software performs an elaborate "connect-the-dots" process to make the wonderful pictures of protein structure we see in textbooks of biomolecular structure, like the insulin structure 3INS (shown in Figure 1. The connections used are, of course, the chemical bonds between all the atoms. In current use, 3-D molecular structure database records employ two different "minimalist" approaches regarding the storage of bond data.

The legacy approach to recording atoms and bonds is something we shall call the *chemistry rules* approach. The rules are the observable physical rules of chemistry, such as "The average length of a stable C—C bond is about 1.5 angstroms." Applying these rules to derive the bonds means that any two coordinate locations in space that are 1.5 Å apart and are tagged as carbon atoms always form a single bond. With the chemistry rules approach, we can *simply disregard the bonds*. A perfect and complete structure can be recorded without any bond information, provided it does not break any of the rules of chemistry.

The chemistry rules approach is the basis for the original 3-D biomolecular structure file format, the PDB format from the Protein Data Bank at Brookhaven). These records, in general, lack complete bond information for biopolymers. There is no residue dictionary required to interpret data encoded by this approach, just a table of bond lengths and bond types for every conceivable pair of bonded atoms.

Every software package that reads in PDB datafiles must reconstruct the bonds based on these rules. Since the rules **have never** been explicitly codified for programmers interpret- between sequence and structure databases, the lack of encoding of the active biological unit, and the lack of encoding of the relationship of the observed structure to the parent gene.

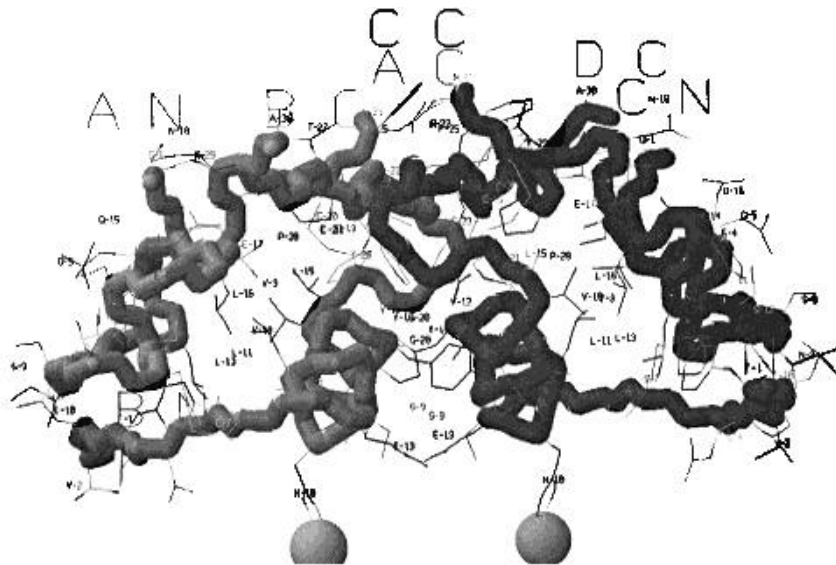


Figure 1 The insulin structure 3INS illustrated using Cn3D. Four chains are depicted in the crystallo graphic unit. This structure illustrates two of many bioinformatics bridges that must be spanned

IDS the bonding in PDB files, software can be inconsistent in the way it draws bonds, especially when different algorithms and distance tolerances are used. The PDB file approach is minimalist in terms of the data stored in a record, but the algorithms required to decipher this information properly are more sophisticated than would be needed if the bonding information and chemical graph were specified in the record. This forces the programmers to do much more work. Exceptions to the bonding rules need to be trapped by complicated logic statements programmed on a case-by-case basis.

The second approach is used in the database records of the Molecular Modeling Database (MMDB), which is derived from the data in PDB. MMDB uses a standard residue dictionary, a record of all the atoms and bonds in the polymer forms of amino acid and nucleic acid residues, plus end-terminal variants. Such data dictionaries are common in the specialized software used by structural scientists to solve structures. The software that reads in MMDB data can use the bonding information supplied in the dictionary to connect atoms together, without trying to enforce the rules of chemistry. As a result the 3-D coordinate data is consistently interpreted by software. This approach also lends itself to simpler software, because exceptions to bonding rules are recorded within the database file itself and read in without the need for another layer of exception-handling code.

Some scientists unfamiliar with structure data often expect all structures in the public databases to be of "textbook" quality. They are surprised when parts of a structure are missing. The availability of a 3-D database record for a particular molecule does not ever imply its *completeness*. Structural completeness is defined as follows: *At least one coordinate value for each and every atom in the chemical graph.*

Completeness is rare in structure database records. Most X-ray structures lack coordinates for hydrogen atoms because the locations of hydrogens in space are not resolved by the experimental method. However some modeling software can be used to

algorithmically estimate the locations of hydrogens and reconstruct a structure record with model hydrogens added. It is easy to identify the products of molecular modeling in structure databases. These often have overly complete coordinate data, usually with all possible hydrogen atoms present that could not have been found using an experimental method.

3.2.1.3 PDB: Protein Data Bank at Brookhaven National Laboratories

Overview

The use of computers in biology has its origins in biophysical methods, such as X-ray crystallography. Thus it is not surprising that the first "bioinformatics" database was built to store this complex 3-D data. The modern Protein Data Bank contains the core public collection of three-dimensional structures of proteins, as well as holding 3-D structures of nucleic acid, carbohydrates, and a variety of complexes experimentally determined by X-ray crystallographers and NMR spectroscopists. This section focuses briefly on the database/bioinformatics services offered by the Protein Data Bank.

The PDB overlaps in scope with several other databases. The Cambridge Crystallographic Data Centre archives the structures of small molecules; oligonucleotides appear in both the CCDC and PDB. This information is extremely useful in studies of conformations of the component units of biological macromolecules, and for investigations of macromolecule-ligand interactions. The Nucleic Acid Structure Databank (NDB) at Rutgers University, New Brunswick, New Jersey, USA complements the PDB. The BioMagResBank, at the Department of Biochemistry, University of Wisconsin, Madison, Wisconsin, USA, archives protein structures determined by Nuclear Magnetic Resonance.

The archives collect not only the results of structure determination, but also the measurements on which they are based. The PDB keeps the new data from X-ray structure determinations, and the BioMagRes Bank those from NMR.

The PDB assigns a four-character identifier to each structure deposited. The first character is a number from 1-9. Do not expect mnemonic significance. In many cases several entries correspond to one protein - solved in different states of ligation, or in different crystal forms, or re-solved using better crystals or more accurate data collection techniques. There have been at least four generations of sperm whale myoglobin crystal structures.

Classifications of protein structures

Several web sites offer hierarchical classifications of the entire PDB according to the folding patterns of the proteins:

- SCOP: Structural Classification of Proteins
- CATH: Class/Architecture/Topology/Homology
- DALI: Based on extraction of similar structures from distance matrices
- CE: A database of structural alignments

These sites are useful general entry points to protein structural data. For instance,

SCOP offers facilities for searching on keywords to identify structures, navigation up and down the hierarchy, generation of pictures, access to the annotation records in the PDB entries, and links to related databases.

PDB Database Services

The World Wide Web site of the Protein Data Bank at Brookhaven National Laboratories offers a number of services for submitting and retrieving 3-D structure data.

Submitting Structures

For those who wish to submit 3-D structure information to PDB, there are now Web-based procedures for submitting structure data via the AutoDep service. Since this procedure is undergoing changes at the time of writing, the most up-to-date information must be found on PDB's Web site. Nucleic acid structures are now being accepted for deposition at NDB, the Nucleic Acids Database. The Biotech Validation Suite sites are mirrored sites that provides services that can be used to screen PDB files for stereochemical and geometrical inconsistencies prior to submitting a structure.

It has been the policy of PDB to reject 3-D structures that result from computational 3-D modeling procedures, rather than a physical experiment. Consult with PDB for the latest details on any exceptions that may have been declared.

PDB-ID Codes

The structure record accessioning scheme of the Protein Data Bank is a unique four-character alphanumeric code called a PDB-ID or PDB code. This scheme uses the digits 0 to 9, and the uppercase letters A to Z. Thus there are over 1.3 million possible combinations. PDB-IDs are not assigned in any particular order. Rather, indexers at the Protein Data Bank try to devise mnemonics that make the structures easier to remember, such as 3INS, the record for insulin shown earlier (Figure 1).

Validating PDB Sequences

To properly validate a sequence from a PDB record, one must first derive the *implicit* sequence in the ATOM records. This is a nontrivial processing step. If the structure has gaps owing to lack of completeness, there may be a set of *implicit sequence fragments* for a given chain. Each of these fragments must be aligned to the *explicit* sequence of the same chain provided in the SEQRES entry. This treatment will produce the complete chemical graph, including the parts of the biological sequence that may be missing coordinate data. This kind of validation is done upon creation of records for the MMDB and mmCIF databases.

The best source of validated protein and nucleic acid sequences in single-letter code derived from PDB structure records is NCBI's MMDB service in the Entrez system. The sequence records for our insulin example have database accessions constructed systematically and can be retrieved from the protein sequence division of Entrez using the accessions: pdb | 3INS | A, pdb | 3INSIB, pdb | 3INS | C, and pdb | 3INS I D. PDB files also have references in DBXEEF records to sequences in the Swiss-Prot protein database. Note that the Swiss-Prot sequences will not necessarily correspond to the structure, since the validation process described here is not carried out when these links are made! Also note that many PDB files currently have ambiguously indicated taxonomy,

reflecting the presence in some in 3-D structures of complexes of molecules that come from different species.

3.2.1.4 Molecular Modeling Database at NCBI *Overview*

NCBI's Molecular Modeling Database, is part of NCBI's Entrez system. It is a compilation of all the Brookhaven Protein DataBank 3-D structures of biomolecules from crystallographic and NMR studies. MMDB is a database of ASN.1-formatted records, not PDBformatted records. Structures in MMDB have value-added information compared to the original PDB structures. These include the addition of the explicit chemical graph information following an extensive suite of validation procedures, the addition of uniformly derived secondary structure definitions, citation matching to MEDLQSTE, and the moleculebased assignment of taxonomy to each biologically derived protein or nucleic acid chain.

MMDB Database Services

A number of services including BLAST (Altschul et al., 1990) search sets of validated sequences, structure-sequence relationships, file format translation, and a programming interface are provided by NCBI's MMDB project.

Free Text Query of Structure Records

Like many other implementations of 3-D structure services, the MMDB database can be searched with free text using the World Wide Web Entrez or Network Entrez front end. MMDB is also referred to as Entrez's Structure division. Search fields in MMDB include PDB and MMDB ID codes, free text from the original PDB Remark records, author name, and other bibliographic fields.

The screenshot shows a Netscape browser window titled "Netscape: MMDDB Structure Summary". The browser's menu bar includes "File", "Edit", "View", "Go", "Bookmarks", "Options", "Directory", "Window", and "Help". The main content area has a header "NCBI MMDDB STRUCTURE SUMMARY" with links for "BLAST", "Entrez", and "?".

Below the header, there is a section for "Structure: 1BNR". The content includes:

- Contents: Barnase Molecule: Barnase (G Specific Endonuclease); Ec: 3.1.27.-; Other_details: Nmr, 20 Structures
- Class: Microbial Ribonuclease
- Source: Organism_scientific: Bacillus Amyloliquefaciens;
- Expression_system: Escherichia Coli; Expression_system_plasmid: Ptz18
- Derived; Expression_system_gene: Barnase
- Authors: M.Bycroft
- PDB Deposition: 31-Mar-95
- References: PubMed/MEDLINE
- MMDB Id: 3832 1BNR at PDB

To the right of this text is a small 3D ribbon diagram of the protein structure.

Below this is a text area titled "Validated Sequence(s) in FASTA format; Heterogen Names" containing the following sequence:

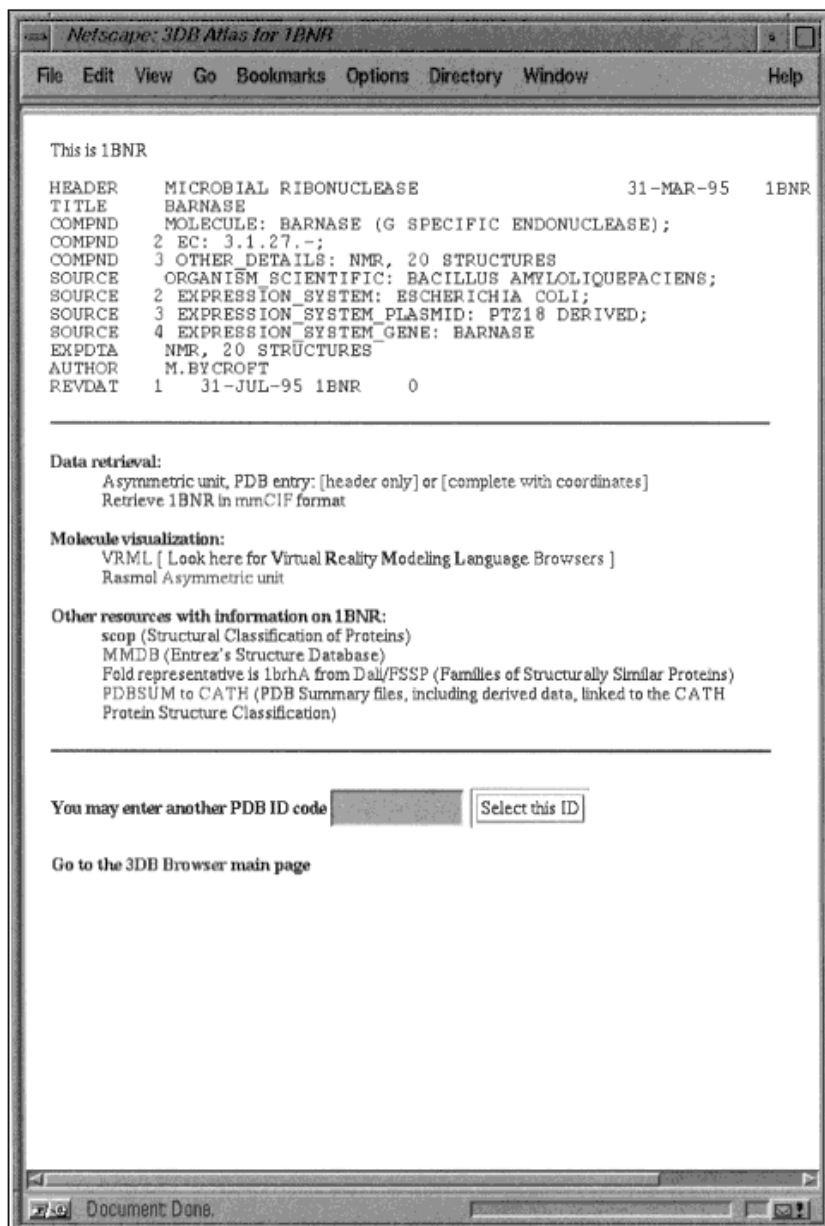
```
>gi|1127282|pdb|1BNR| Barnase Molecule: Barnase (G Spe  
A QVINTFDGVADY LQTYHKLFDNY I TKSEAQLGWVASKGNLADVAPGKS IGGDI  
GKLPKSGRTWREADINY TSGFRNDR ILYSSDWL IYKTTDHYQIFTR IR
```

Further down, there are interactive buttons for "Protein Sequences" and "Protein 3-D Structures", each with a "similar to" dropdown and a "Chain -" dropdown. The "Protein 3-D Structures" button also includes the text "computed by VAST".

Below these is a section for "View/Save: 1BNR". It features a "View" button, a dropdown for "Structure using" (set to "Cn3D"), a dropdown for "showing" (set to "Cn3D Subset"), and a "MAGE Color" dropdown (set to "Molecule Number"). There are also checkboxes for "Launch Viewer", "See File", and "Save".

At the bottom, there is a "Help" section with links for "(MMDB)", "(Cn3D)", "(Cn3D Helper App.)", "(Viewing Options)", and "(VAST)". A "Back to MMDB-Homepage" link is also present. A search bar labeled "Search MMDB text." is located at the very bottom, with "Find" and "Clear" buttons.

(a)



(b)

MMDB Structure Summary

MMDB's Web interface provides a Structure Summary page for each MMDB structure record, as shown in Figure 3.2b. MMDB Structure Summary pages provide the FASTA-formatted sequences for each chain in the structure, links to MEDLME references, links to the 3DB Atlas record and the Brookhaven PDB site, links to protein or nucleic acid sequence neighbors for each chain in the structure, and links to the VAST structure-structure comparison for each domain on each chain in the structure.

3.2.1.5 BLAST Against PDB Sequences: New Sequence Similarities

When a researcher wishes to find a structure related to a new sequence, NCBI's BLAST service provides a copy of all the validated sequences from MMDB in "pdb" (a BLAST search database). The BLAST Web interface, can be used to paste a sequence in FASTA format into the sequence entry box and select the "pdb" sequence database to perform a search against all the validated sequences in the current public structure database.

Entrez Neighboring: Known Sequence Similarities

If one is starting with a sequence that is already in Entrez, BLAST has already been performed. Structures that are sequence similar to a given protein sequence can be found by means of Entrez's neighboring facilities.

To use Entrez's neighboring features to determine whether a sequence-similar 3-D structure exists for a known sequence, start with WWW Entrez's "Search the NCBI protein database" option and perform a query that retrieves the sequence of interest (e.g., query oncomodulin). Upon retrieving the summary of records in the query, select the Structure Links option on the pulldown list above the first hit and press the Display button to view two MMDB records, 1KRO and 1OMD.

The query can be broadened to find remote similarities by performing protein neighboring first, then locating links from that list of protein neighbors to 3-D structures. Starting with the same example protein query, oncomodulin, will reveal that each protein record has a few hundred protein neighbors. Select the protein neighbor list first, then use the [Display] [structure links] commands at the top of the Summary page containing all the protein sequence neighbors. This will give a much longer list of 3-D structures, including all the other homologous calcium-binding proteins (e.g., parvalbumin) in the 3-D structure database.

3.2.1.6 VAST: Structure Similarity by 3-D Shape

VAST (Vector Analysis Search Tool) is a similarity **measure** of 3-D structure. It uses 3-D vector elements derived from secondary structure—no sequence information is used in the searching. VAST is capable of finding structural similarities when no sequence similarity is detected. VAST, like BLAST, is run on all entries in the database in an $N \times W$ manner, and the results are stored for fast retrieval using the Entrez interface. More than 10,000 domain substructures within the current 3-D structure database have been compared to one another using the VAST algorithm, and the structure-structure alignments (Figure 3.2c) and superpositions recorded. The VAST algorithm focuses on similarities that are surprising in the statistical sense. One does not waste time examining many similarities of small substructures that occur by chance in protein structure comparison. For example, very many small segments of β sheets have obvious, but not surprising, similarities. The similarities detected by VAST are often examples of remote homology, undetectable by sequence comparison. As such they may provide a broader view of the structure, function, and evolution of a protein family.

While a sequence-sequence similarity program provides an alignment of two sequences, a structure-structure similarity program provides a 3-D structural superposition. This is a set of 3-D rotation-translation matrix operations that superimpose the similar parts of the structure together. A conventional sequence alignment can be derived from the 3-D superposition by finding the carbons in the protein backbone that are superimposed in space. In addition to a listing of similar structures, VAST-derived structure neighbors contain detailed residue-by-residue

alignments and 3-D transformation matrices for structural superposition. In practice, refined alignments from VAST appear conservative, choosing a highly similar "core" substructure compared with DALI superpositions. With the VAST superposition one easily identifies regions in which protein evolution has modified the structure, whereas DALI superpositions may be more useful for comparisons involved in making structural models. Both VAST and DALI superpositions are excellent tools to investigate relationships in protein structure, especially when used together with the SCOP database of protein families, shown in Figure 3.2*d*.

Netscape: Vast Results

File Edit View Go Bookmarks Options Directory Window Help

NCBI VAST

VAST Homepage and table legends

Similar structures to 1BNR chain_domain 0

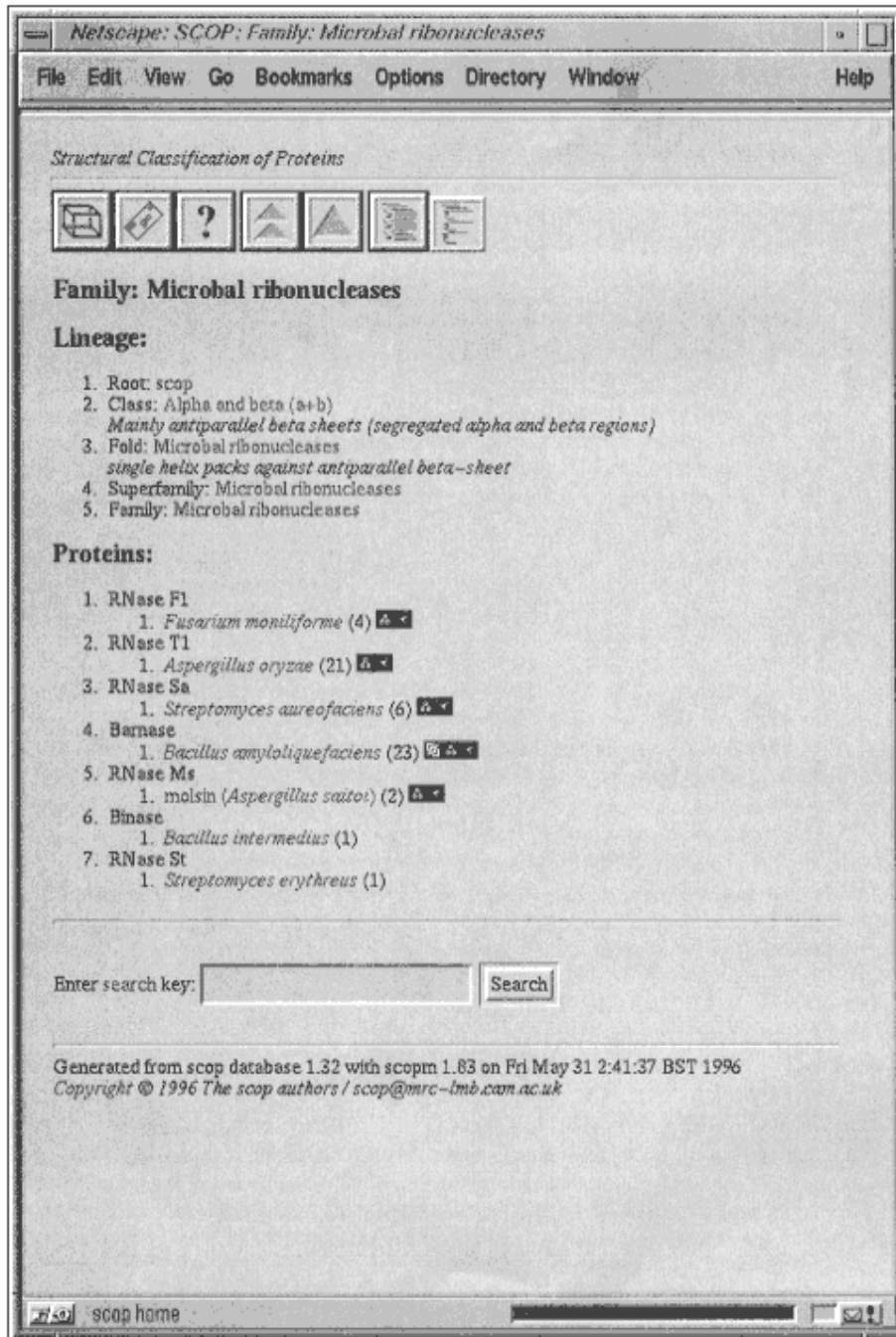
Neighbors of: 1BNR Chain_Domain 0
Barnase Molecule: Barnase (G Specific Endonuclease); Ec: 3.1.27.-; Other_details: Nmr, 20 Structures

Structure	C	D	A	SCO	PVAL	RMSD	NRES	Id	Contents
(P)(K) 1BNF	C	0	1	13.3	8.1	1.1	102	98.0	Barnase (E.C.3.1.27.-) Disulfide Mutant With Thr 70 Replaced By Cys And Ser 92 Replaced By Cys (T70c,S92c)
(P)(K) 1BNE	A	0	1	12.4	8.1	1.4	104	98.1	Barnase (E.C.3.1.27.-) Disulfide Mutant With Ala 43 Replaced By Cys And Ser 80 Replaced By Cys (A43c,S80c)
(P)(K) 1BNG	C	0	1	12.2	7.9	1.3	103	98.1	Barnase (E.C.3.1.27.-) Disulfide Mutant With Ser 85 Replaced By Cys And His 102 Replaced By Cys (S85c,H102c)
(P)(K) 1BRS	C	0	1	12.2	7.9	1.4	107	100.0	Barnase (G Specific Endonuclease) (E.C.3.1.27.-) Complexed With Barstar Mutant With Cys 40 Replaced By Ala And Cys 82 Replaced By Ala (C40a)

Re-Sort table by

Display table as an Entrez Document Summary form.

(c)



(d)

Figure 3.2 Entry via (a) Entrez's Structure Division or (b) PDB's 3DB browser with the structure 1 BNR (Bycroft et al., 1991) can link the user to a variety of other pages with information about this structure, including (c) VAST (Gibrat et al., 1996) structure—structure comparisons and the (d) SCOP (Murzin et al., 1995) database with information on protein family classification.

3.2.1.7 Structure File Formats

PDB

The PDB file format is column-oriented, like that of the punched cards used by early FORTRAN programmers. The file format specification is maintained at PDB's Web site. Most structural software developed by structural scientists is written in FORTRAN, while the rest of bioinformatics has adopted other languages, such as those based on C. PDB files are often a paradox: they look rather easy to parse, but they have a few nasty surprises, as already indicated in this chapter. To the uninitiated, the most obvious problem is that the information about biopolymer bonds is missing, obliging one to program in the rules of chemistry, clues to the identity of each atom given by the naming conventions of PDB, and robust exception handling. PDB parsing software often needs lists of synonyms and tables of exceptions to correctly interpret the information. However this chapter is not intended to be a manual of how to construct a PDB parser.

Two newer chemical-based formats have emerged: mmCIF (MacroMolecular Chemical Interchange Format) and MMDB (Molecular Modeling Database). Both these file formats are attempts to modernize PDB information. Both start by using data description languages, which are consistently machine parsable. The data description languages use "tag value" pairs, which are like variable names and values used in a programming language. In both cases the format specification is composed itself in a machine-readable form, and software uses this format specification document to validate incoming streams of data. Both file formats are populated from PDB file data using the strategy of alignment-based reconstruction of the implicit ATOM and HETATM chemical graphs with the explicit SEQEES chemical graphs, together with extensive validation, which is recorded in the file. As a result, both these file formats are superior for integrating with biomolecular sequence databases than PDB format data files, and their use in future software is encouraged.

mmCIF

The mmCIF file format was originally intended to be a biopolymer extension of the CIF familiar to small-molecule crystallographers, based on a subset of the STAR syntax. CIF software for parsing and validating format specifications is not forward-compatible with mmCIF, since these have different implementations for the STAR syntax. The underlying data organization in an mmCIF record is a set of relational tables. The mmCIF project refers to their format specification as the *mmCIF dictionary*, kept on the World Wide Web at the Nucleic Acids Database site at Rutgers University. The mmCIF dictionary is a large document containing specifications for holding the information stored in PDB files as well as many other data items denvable from the primary coordinate data, such as bond angles. The mmCIF data specification gives this data a consistent interface, which has been used to implement the NDB Protein Finder, a WWW-based query format in a relational database style.

Validating an incoming stream of data against the large mmCIF dictionary entails a rather large overhead in the I/O of mmCIF data. Hence mmCIF is probably destined to be an archival and advanced query format. Software libraries for reading mmCIF tables into relational tables into memory in FORTRAN and C are available for a few UNIX platforms commonly used by crystallographers.

MMDB

The MMDB file format is specified by means of the ASN. 1 data description language, which is used in a variety of other settings, including telecommunications and automotive manufacturing. Since the U.S. National Library of Medicine also uses ASN. 1 data specifications for sequence and bibliographic information, the MMDB format borrows

certain elements from other data specifications, such as the parts used in describing bibliographic references cited in the data record. ASN. 1 files can appear as human-readable text files or as a variety of binary and packed binary files that can be decoded by any hardware platform. The MMDB standard residue dictionary is a lookup table of information about the chemical graphs of standard biopolymer residue types. The MMDB format specification is kept at the MMDB FTP site at NCBI. The MMDB ASN. 1 specification is much more compact and has fewer data items than the mmCIF dictionary, and avoids derivable data altogether.

In contrast to the relational table design of mmCIF, the MMDB data records are structured as hierarchical records. In terms of performance, ASN. 1 -formatted MMDB files are much faster I/O streams than mmCIF or PDB records. Their nested hierarchy requires fewer validation steps at load time than the relational scheme in mmCIF or in the PDB file format. Hence ASN. 1 files are ideal for 3-D structure database browsing.

A complete application programming interface is available for MMDB as part of the NCBI toolkit containing a wide variety of C code libraries and applications. Both an ASN. 1 I/O programming interface layer and a molecular computing layer (MMDB-API) are present in the NCBI toolkit. The NCBI toolkit supports x86 and Alpha-based Windows platforms, Macintosh 68K and PowerPC cpus, and a wide variety of TJNTX platforms. The 3-D structure database viewer, Cn3D, is an MMDB-API—based application with source code included in the NCBI toolkit.

3.2.1.8 Database Structure Viewers

RasMol and RasMol-Based Viewers

Several viewers for examining PDB files are available. The most popular one is Roger Sayle's RasMol. RasMol represents a breakthrough in software-driven 3-D graphics, and its source code is recommended study material for anyone interested in high-performance 3-D graphics. RasMol treats PDB data with extreme caution and often recomputes information, making up for inconsistencies in the underlying database. It does not try to validate the chemical graph of sequences or structures encoded in PDB files. RasMol does not perform internally neither dictionary-based standard residue validations or alignment of explicit and implicit sequences. RasMol ignores information in correlated disorder ensembles and displays only one NMR model at a time. Other data elements encoded in PDB files, such as disulfide bonds, are recomputed based on rules of chemistry, rather than validated.

RasMol contains many excellent output formats and can be used with the Molscrip program to make wonderful PostScript™ ribbon diagrams for publication. For optimal use of RasMol, however, one must master its command line language, a familiar feature of many legacy 3-D structure programs. RasMol executables, a gallery of RasMol images, a RasMol tutorial, source code, and a user-based support mailing list are available from the RasMol home page maintained by Eric Martz at the University of Massachusetts.

Several new programs, free for academic users, are becoming available. Based on RasMol's software-driven 3D rendering algorithms and sparse PDB parser, these include Chime™, a Netscape™ plug-in provided by MDLI Inc. WebMol, a Java applet by Dirk Walther, is a Java-based 3D structure viewer apparently based on RasMol style rendering, as seen in Figure 3.3. WebMol has demonstrated that the Java bytecode interpreters in current use on most PCs and workstations are not sufficiently fast to perform RasMol-style software-based 3-D rendering for structures in excess of 200

residues. This limits WebMol's utility to the smaller structures, or to virtual bond representation of molecules.

MMDB Viewer: Cn3D

Cn3D is a new 3-D structure viewer, used for viewing MMDB data record. Because the chemical graph ambiguities in data in PDB entries have been removed to make MMDB data records and because all the bonding information is explicit, Cn3D has the luxury of being able to display 3D database structures consistently, without the parsing, validation, and exception-handling overhead required of programs that read PDB files. Cn3D's default image of a structure is more intelligently displayed because it works without fear of misrepresenting the data. However, because Cn3D is dependent on the complete chemical graph information in the ASN. 1 records of MMDB, it does not currently read in PDB files.

Cn3D provides a set of control panels that can hide or appear at the side of the 3-D image. For example, the Viewer Control panel, which appears at the top of the 3-D image, has a set of animation controls that look like tape recorder controls and are used for displaying the multiple structure ensembles of NMR structures one after the other, or the multiple structures superimposed in a VAST similarity relationship image. The Go button makes the images animated, and the user can rotate or zoom the structure while it is playing the animation. Cn3D 2.0, which should be available by the time of publication, will have complete state-saving capabilities. This will make it possible to color and render a structure, and then save the information right into the ASN. 1 structure record, a departure from hand-editing PDB files or writing scripts. This information can be shared with other Cn3D users on different platforms.

Other 3D Viewers: Mage, CAD, VRML

A variety of file formats have been used to present 3-D biomolecular structure data lacking in chemistry-specific data representations. These are viewed in generic 3-D data viewers such as those used for "macroscopic" data like engineering software or virtual reality browsers. The journal *Protein Science* published one such generic file format: the Kinemage the first widely used molecular structure software made available to personal computer users prior to the emergence of the Internet and the World Wide Web. File formats like Kinemage and VRML contain 3-D graphical display information but little or no information about the underlying chemical graph of a molecule. Further, it is difficult to encode the variety of rendering styles in such a file; one needs a separate VRML file for a spacefilling model of a molecule, a wireframe model, a ball-and-stick model, and so on, since each explicit list of graphics objects (cylinders, lines, spheres) must be contained in the file.

Biomolecular 3-D structure database records are currently not compatible with "macroscopic" software tools such as those based on CAD software. Computer-aided design software is a mature, robust technology, generally superior to the available molecular structure software. But CAD software and file formats in general are ill-suited to examine the molecular world, owing to the lack of certain "specialty" views and analytical functions built in for the examination of details of protein structures.

Summary:

The PDB (Protein Data Bank) is the major repository of protein structures (and to some extent of nucleic acid structures). The three-dimensional atomic coordinates are not very convenient for examination of the structure with the naked eye, a large number of 3D structure viewers have been designed to graphically view these coordinates. The most obvious data in a typical 3-D structure record, irrespective of the file format in use, is the *coordinate data*, the locations in space of the atoms of a molecule, represented by (x, y, z) triples, distances along each axis to some arbitrary origin in space. Several viewers for examining PDB files are available. The most popular one is Roger Sayle's RasMol. Cn3D is a new 3-D structure viewer, used for viewing MMDB data records.

Model Questions

1. Write a note on different structural databases and their services?
2. Discuss about the tools of structure superimposition and viewers?

References

1. Bioinformatics - Sequence and Genome Analysis- David W. Mount.
2. Bioinformatics – A practical guide to the Analysis of Genes and Proteins – Andreas D. Baxevanis and B F Francis Quelette.
3. Introduction to Bioinformatics by Arthur M. Lesk.

B.M.REDDY M.Tech. (HBTI, Kanpur)
Lecturer, Centre for Biotechnology
Acharya Nagarjuna University.

Lesson 3.2.2**Genomic Databanks****Objective****3.2.2.1 Objective****3.2.2.2 Introduction****3.2.2.3 List of Gnostic Databanks****3.2.2.4 Summary****3.2.2.5 Model Questions****3.2.2.6 References****3.2.2.1 Objective**

- To introduce the use of databanks and their need and importance in biological research.
- To get acquainted to various genome databanks available to aid the research.
- To have a brief idea on the genomic databanks available.

3.2.2.2 Introduction

With the advent of genome biology and sequencing technology DNA sequence data had been produced at an enormous rate, and at last the human genome was almost completely sequenced and immediately published in Nature (1) and the International Nucleotide Sequence Databases (INSD) in 2001. The immediate public release of such a large-scale sequence data was possible perhaps only through INSD. Since INSD was composed of the DNA Data Bank of Japan (DDBJ, <http://www.ddbj.nig.ac.jp>), the EMBL Bank and GenBank, the publication of the human genome data was carried out smoothly on the basis of a well-established international collaboration. As a result, a number of researchers enjoyed worldwide the simultaneous publications of the Nature paper and the relevant sequence data. By use of the DNA data and retrieval and analysis tools available at DDBJ one can push ahead with one's research in various areas of life sciences. With this in mind we at DDBJ have made an effort in collecting as much data as possible mainly from Japanese researchers. The increase rate of the data we collected, annotated and released to the public in the past year is 1.6 times in the number of entries and 1.5 times in the number of bases. The increase rate is being accelerated even after the human genome was sequenced, because sequencing technology has been remarkably advanced and simplified, and research in life sciences has been shifted from the gene

3.2.2.3 List of Gnostic Databanks**OMIM (Online Mendelian Inheritance in Man)**

This database is a catalog of human genes and genetic disorders authored and edited by Dr. Victor A. McKusick and colleagues at NCBI, Bethesda, Maryland.

GDB (Genome DataBase)

GDB holds data on Human gene loci, polymorphisms, mutations, probes, genetic maps, GenBank, citations and contacts.

GeneCards - integrated biomedical genetic information

Although it will take some years until the human genome is totally sequenced, and still a much longer time to learn about the functions of the products of those genes, the complex organization and the vast amount of biomedical information already accessible often cause certain problems that are somehow connected to the phenomenon of "information overflow" and the often very time-consuming process of information retrieval or mining. Thus, many scientists feel that new approaches to organize scientific information are urgently needed. GeneCards is a database that intends to address some of these problems by integrating biomedical information taken from several sources (GDB, MGD, OMIM, SWISS-PROT, HGMD, Doctor's Guide to the Internet), and by presenting them in a way facilitating a quick.

The Unified Database for Human Genome Mapping

The Unified Database (UDB) integrates information on the human genome, with emphasis on mapping information. Mapped DNA segments, classified by categories (such as genes, EST clusters and STSs mapped by various methods) are presented on a Megabase-scale integrated map, with further information and links to relevant databases. UDB includes data from numerous resources, including the Genome Database, Whitehead Institute/MIT Center for Genome Research, Genethon, GeneMap'99 and others. Integrated map locations were calculated from separate method-specific chromosome maps (e.g. genetic linkage, radiation hybrid, and content-contig maps) by a simple scaling algorithm.

OMIM gene map

The OMIM gene map presents the cytogenetic map location of disease genes and other expressed genes described in OMIM. You enter a position, say '17q11' and you get all the omim records in that region, even things that map to '17cen-q12'

Ensembl - annotated human sequences

Ensembl is a joint project between EMBL-EBI and the Sanger Centre to develop a software system for automating analysis of genomic data. It is being applied to the publically released human genome data stream.

The HuGeMap database

HuGeMap is a database that contains:

- the genetic maps from Genethon,
- the genetic maps from Cooperative Human Linkage Center,
- the physical maps from CEPH/Ginithon
- the physical maps from the Whitehead Institute-MIT

HuGeMap is interconnected to the radiation hybrid gene map database RHdb, maintained at EBI. This interconnection is based on CORBA servers that have been implemented at Infobiogen and EBI, and that share the same IDL (see the Object Management Group for an introduction to CORBA).

GenAtlas

Compiles the information relevant to the mapping efforts of the Human Genome Project. GENATLAS/GEN is a repertory of three types of objects : genes, diseases, and markers.

Genome Navigator: Whitehead/MIT STS-based Map of the Human Genome

Genome Navigator is an attempt to provide a visual interactive gateway to major databases containing physical and genetic mapping information about the human genome. Genomic maps of these organisms are displayed using DerBrowser, a Java applet, designed as a universal tool to display and navigate various types of maps. Among other features, it allows a user to query external databases about any map object.

The Genome Channel

This system is a prototype graphical browser for querying the annotated reference genome. The Java interface relies on a number of underlying resources, analysis tools and data-retrieval agents to provide an up-to-date view of genomic sequences as well as computational and experimental annotation. Designed to be simple enough for a layperson, the channel also offers sophisticated capabilities for hypothesis testing.

HGBASE - human genic bi-allelic sequences - SNPs

HGBASE lists human intra-genic promoter to transcription end point DNA sequence polymorphisms. It has been constructed by The Department of Genetics and Pathology at Uppsala University and Interactiva Biotechnologie GmbH. HGBASE does not include gene mutations, but is a catalogue of intra-genic sequence variants found in normal individuals. Despite its name, HGBASE contains all types of gene based variation and is not limited to bi-allelic Single Nucleotide Polymorphisms SNP s. Functionally consequential polymorphisms e.g. promoter and non-silent codon changes and other polymorphisms e.g. intron sequence differences are included. Search tools are provided to find data within HGBASE. Searches utilise a text string or a DNA sequence. Data submission is by a series of Web page data submission forms. All submitted data is made available to other public databases. The exponential growth in polymorphism discovery requires that scientists make every effort to submit their data to HGBASE to ensure it remains up to date. HGBASE does not claim any rights to publicly available or submitted data, instead this remains the property of the original submitter/discoverer. Deposition of data within HGBASE requires only the allelic DNA sequence, the allele frequencies, the host gene name, and the intra-genic domain. Additional information, such as assay conditions, can be supplied but this is optional.

Single Nucleotide Polymorphisms in the Human Genome

This website is designed to provide the human genetics community with access to single nucleotide polymorphism (SNPs) that have been developed as genetic markers on the human genome. The site is organized by chromosomes and cytogenetic location. Each SNP has PCR primer and conditions associated with it. Currently, we only post the SNPs that we have helped to develop. After we have posted all of our SNPs, we'll be adding SNPs from the literature and from collaborators, and we will be happy to have others contribute to the database.

dbSNP - A Database of Single Nucleotide Polymorphisms

In collaboration with the National Human Genome Research Institute, The National Center for Biotechnology Information has established the dbSNP database to serve as a central repository for both single base nucleotide substitutions and short deletion and insertion polymorphisms. Once discovered, these polymorphisms could be used by additional laboratories, using the sequence information around the polymorphism and the specific experimental conditions. (Note that dbSNP takes the looser 'variation' definition for SNPs, so there is no requirement or assumption about minimum allele frequency.) The data in dbSNP will be integrated with other NCBI genomic data. As with all NCBI projects, the data in dbSNP will be freely available to the scientific community and made available in a variety of forms.

Human SNP Database

This is the Whitehead/MIT SNP data.

ALFRED - Allele Frequency Database

This gives gene frequency data for a diverse set of population samples and genetics systems. It contains data on more than 40 populations representing most major regions of the world and data on more than 150 genetic systems including SNPs, STRPs and insertion-deletion polymorphisms.

Whitehead/MIT STS-Based Map of the Human Genome

This contains YAC screening data for several thousand STSs. For each STS, information is held on the following types of raw data (where available):

- Chromosomal assignments
- YAC library screening results
- Radiation hybrid panel screening results

In addition, information is available on the following preliminary preliminary analyses:

- Doubly-linked YAC contigs
- Singly-linked YAC contigs
- Radiation hybrid maps
- Integrations between genetic, radiation hybrid, and YAC contig maps.

Transcript Map of the Human Genome

A small portion of each cDNA sequence is all that is needed to develop unique gene markers, known as sequence tagged sites or STSs, which can be detected in chromosomal DNA by assays based on the polymerase chain reaction (PCR). To construct a transcript map, cDNA sequences from a master catalog of human genes were distributed to mapping laboratories in North America, Europe, and Japan. These cDNAs were converted to STSs and their physical locations on chromosomes determined on one of two radiation hybrid (RH) panels or a yeast artificial chromosome (YAC) library containing human genomic DNA. This mapping data was integrated relative to the human genetic map and then cross-referenced to cytogenetic band maps of the chromosomes. (Further details are available in the accompanying article in the 25 October issue of SCIENCE). The histograms reflect the distributions and densities of genes along the chromosomes. Because the individual genes (>16,000) are too numerous to represent, images have been chosen to illustrate the myriad aspects of

human biology, pathology, and relationships with other organisms that can be revealed by analysis of genes and their protein products.

Human Telomere Information

This is a section of GenLink's Telldb giving literature citations and other information on human telomeric regions.

The Genetic Location Database (LDB)

Ldb is an analytical database for constructing fully integrated genetic and physical maps. The ldb program generates an integrated map (known as the summary map) from partial maps of physical, genetic, regional, somatic hybrid, mouse homology and cytogenetic data. The summary maps and the data used to build up such maps are available from this site. The files for each chromosome are stored in the same directory which include the summary map, partial maps, lod files and the parameter files. As this server is experimental many of the chromosome directories are incomplete with the most complete map sets being chromosomes 1,9,21 and X.

The dysmorphic human and mouse homology database

This consists of three separate databases of human and mouse malformation syndromes together with a database of mouse/human syntenic regions. The mouse and human malformation databases are linked together through the chromosome synteny database. The purpose of the system is to allow retrieval of syndromes according to detailed phenotypic descriptions and to be able to carry out homology searches for the purpose of gene mapping. Thus the database can be used to search for human or mouse malformation syndromes in different ways:-

- By specifying specific malformations or clinical features, or chromosome locations.
- By Homology.
- By asking for human syndromes located at a chromosome region syntenic with a specific mouse chromosome region (and vice versa from human to mouse).

BodyMap - Anatomical Expression Database of Human Genes

BodyMap is a data bank of expression information of human genes, novel or known, in various tissues or cell types. It is created by random sequencing of clones in 3'-directed cDNA libraries. Since these clones were not amplified, redundancy of the same sequence reflects the quantitative aspect of gene expression in various tissues. You can enter your sequence and it will be matched using fasta to the cDNA sequences in this database.

CEPH-Genethon integrated map

This page allows you to search the CEPH-Genethon mapping data used to build the first generation physical map of the human genome. It gives information on the CEPH YAC library and the QUICKMAP database with the infoclone program to get information about a STS or a YAC.

CEPH Genotype database

The Centre d'Etude du Polymorphisme Humain (CEPH) maintains a database of genotypes for all genetic markers that have been tested in the reference families for linkage mapping of the human chromosomes.

- browse data by chromosome, probe, D-number, Gene name and Heterozygote frequency
- output data for several markers on the same chromosome in CEPH, LINKAGE and CRIMAP formats.
- CEPH Collaborators can now submit data for new markers directly through Internet by using a browser supporting JAVA applets.

All genotypes contributed to the CEPH database are also available by anonymous FTP server. Genotypes, markers description, pairwise lodscores may be downloaded from the FTP server. In addition, the server contains databases for published CEPH consortium maps and also breakpoint maps.

Cooperative Human Linkage Center (CHLC)

The goal of the Cooperative Human Linkage Center is to develop statistically rigorous, high heterozygosity genetic maps of the human genome that are greatly enriched for the presence of easy-to-use PCR-formatted microsatellite markers.

- Genetic maps showing the positions of genetic markers
 - Integrated Maps showing the position of genetic markers constructed using genotype data from the CEPH reference panel.
 - CHLC Marker Maps showing the positions of CHLC generated markers in various reference maps.
- Search by name for information on markers.
 - Search by name for information on markers, including map location primer and pcr conditions, and sequence templates.
 - Likely locations of current CHLC markers, in Version 2.0 Skeletal Maps
 - Tables of CHLC markers characterized by linkage analysis.
 - List of full information on markers generated by CHLC
 - In Current Linkage Map
 - Candidate Linkage Markers
 - Somatic Cell Hybrid Assigned
 - Prior versions of the CHLC generated markers.
 - Marshfield CA-repeat Markers
 - Table of initial typing data
 - Table of sequence data
 - Table of PCR Primers
- CHLC publications.
 - *Science* Maps and Data
 - Copies of the CHLC Newsletter
- CHLC project information.

GeneMap '98 - The International RH Mapping Consortium Map

This is the latest Radiation Hybrid Consortium Human map.

Radiation Hybrid Mapping data (RHdb)

Radiation hybrid maps are an indispensable alternative to genetic maps as they can include non-polymorphic markers and are also powerful enough to order unresolved genetic clusters of polymorphic STSs. An international collaborative project has been started which will produce a large number of these hybrids for the human genome. This in turn will allow the generation of a very precise STS map that will be indispensable in the study of multifactorial diseases. RHdb, the radiation hybrid database is an archive of raw data with links to other related databases. The main data is stored in a relational database. Submissions to this database are made using a standard format. Various export formats will be supported, as well as different ways of accessing the data.

dbEST Expressed Sequence Tag Database

The dbEST Database holds many Human ESTs.

UniGene - Unique Human Gene Sequence Collection

This holds clusters of human EST sequences that represent the transcription products of distinct genes. These sequences are being used for transcript mapping in collaboration with several genome mapping centers. Some of the clusters have already been localized to chromosomes, but more detailed mapping map information is not available at this time.

dbSTS Sequence Tagged Site Database

The dbSTS Database holds many Human STSs.

Whitehead Institute/MIT Genome Center

- Human YAC screening data for sequence tagged sites (STSs) screened on the CEPH mega-YAC library with over 1100 contigs assembled using double linkage between STSs.
- For each STS, they report addresses for the YACs found to contain the STS.
- Human, Rat and Mouse marker map data files.

V BASE: A Directory of Human Immunoglobulin V Genes

A directory of human immunoglobulin germline variable region sequences compiled from over a thousand published sequences (including those in the current releases of the Genbank and EMBL data libraries). There are seven directories: D, JH, JK, JL, VH, VK and VL. Each directory consists of a folder or file containing the germline sequences and a file containing the corresponding reference list.

Human CpG Island database

Look at the Human CpG Island database. This is a flat file containing a description of genes and their associated CpG islands.

Human population genetics database (Genography)

A database of human genetic and cultural diversity, to act as a comprehensive community repository supporting work in human population genetics and quantitative anthropology. The currently available version of the database contains 100,000 gene frequencies from almost 2,000 populations, collected from the literature on classical polymorphisms (essentially protein data) published up to 1986. These data were used for calculations on which the book *History and Geography of Human Genes*, by Cavalli-Sforza, Menozzi, and Piazza, is based. Future versions of the database will include an update of the classical polymorphism data, a collection of published and unpublished DNA data by individual and by population (including RFLPs, microsatellites, and SNPs), and the future CEPH database, to be collected in collaboration with the Human Genome Diversity Project. In addition, information about geography, regional ecology, linguistics, mythology, musicology, and physical anthropology will be included. Finally, various analysis and visualization tools will be provided.

Anthony Nolan Research Institute (ANRI)

The WHO Nomenclature and HLA Sequence alignments are available from this site, together with monthly updates.

GDB Nomenclature Committee

- Activities of the Nomenclature Committee
- Nomenclature mailing list
- Committee members
- Guidelines for choosing a gene symbol
- Submit a proposed gene symbol
- Browse approved gene symbols
- Gene Family Nomenclature
- Checking for existing symbols

Atlas of Genetics and Cytogenetics in Oncology and Haematology

The Atlas of Genetics and Cytogenetics in Oncology and Haematology is a cooperative process of reviewing and updating on somatic genetics, clinical entities in cancer, and on cancer-prone diseases; it is made for and by: cytogeneticists, molecular biologists, and geneticists in general, clinicians in oncology and in haematology, and pathologists.

Human Genome - The Third Millennium

This site aims to help researchers find their way in Web-accessible databases containing Human Genome information, and to find answers to problems related to human genomic clones, contigs, sequences and maps.

It contains:

- Links to relevant databases and tools.
- Search strategies to aid users with little or no experience.
- Practical examples with tips for searching and experimental follow-up.

The Human Gene Mutation Database - HGMD (Cardiff)

This database represents an attempt to collate the majority of known (published) gene lesions responsible for human inherited disease. Originally established for the study of mutational mechanisms in human genes (Cooper and Krawczak 1993), these databases have acquired a much broader utility in that they currently represent the only available comprehensive reference source to the spectrum of mutations underlying human genetic disease. They thus provide information of practical diagnostic importance to (i) researchers in human molecular genetics, (ii) physicians interested in a particular inherited condition in a given patient or family and (iii) genetic counsellors.

- The Point Mutation Database
- The Micro-Deletion Database
- Links to Locus Specific Mutation Databases

Mutation Database Website

Information on nomenclature and design of mutation databases.

- Documentation of the progress of the initiative.
- Proposals from working groups.
- Announcements
- Publications in the area of Mutation Databases

Universal Mutation Database

Software and databases for mutations in human genes.

- Analyze the p53 Database
- Analyze the FBN1 Database
- Analyze the VHL Database
- Analyze the WT1 Database
- The APC Database
- Analyze the LDLR Database
- lists of other locus specific databases
- lists of multi-gene databases

Protein Mutation Database

PMD is based on literature (not on proteins); that is, each entry of the database corresponds to one article which describes protein mutations.

The Androgen Receptor Mutations Database

Constitutional mutations in the androgen receptor gene (AR) impair androgen - dependent male sexual differentiation to various degrees . Somatic mutations in the AR have been found in metastatic prostate cancer. Severe constitutional androgen insensitivity (AI) yields an external female phenotype. Partial constitutional AI yields a range of external genital phenotypes that vary from near - normal female to normal or near - normal male, with or without gynecomastia and other relatively "mild" signs of undervirilization.

Antithrombin Mutation Database Homepage

Antithrombin is a plasma inhibitor of thrombin and other blood coagulation proteinases. Its (functional) deficiency is a strong risk factor for venous thrombosis. The gene coding for antithrombin has been localised to chromosome 1q23-25.

Asthma Gene Database

This is a database for asthma and allergy linkages and mutations. As you can enter and change data from every part of the world they have implemented password restriction. Registration to this database is free.

Breast Cancer Mutation Data Base (BIC)

A resource for the molecular biologist investigating inherited breast cancer providing a central repository for information regarding breast cancer susceptibility genes mutations and polymorphisms. This requires you to register as a BIC member.

BCGD - The Breast Cancer Gene Database

Contains information about genes involved in human breast cancer.

BIOMDB - Database of mutations causing tetrahydrobiopterin deficiencies

BIODEF is a locus-specific database with detailed records of disease-producing allelic variations and natural polymorphic markers.

Blood Group Antigen Mutation Database

This database will deal with mutations in loci of allelic genes that specify the common blood group antigens and the allelic variants of those common genes.

BTKbase - agammaglobulinemia XLA-causing mutations

X-linked agammaglobulinemia (XLA) is an immunodeficiency caused by mutations in the gene coding for Bruton's agammaglobulinemia tyrosine kinase (BTK). A database (BTKbase) of BTK mutations has been compiled and the recent update lists 463 mutation entries from 406 unrelated families showing 303 unique molecular events. In addition to mutations, the database also lists variants or polymorphisms.

The European CD40L Defect Database (CD40Lbase)

X-linked Hyper-IgM syndrome-associated mutation database.

Database of Human Type I and Type III Collagen Mutations

Includes accounts of every known mutation in the genes encoding the alpha-1 and alpha-2 chains of type I collagen

Emery-Dreifuss Muscular Dystrophy Mutation Database

Brief description of mutations.

Factor VII Mutation Database

The Factor VII Mutation Database is currently under construction.

GPCRmut, The G Protein-Coupled Receptors mutant database

Mutation analysis of GPCRs

GPCRDB: Information system for G protein-coupled receptors (GPCRs)

Contains information about GPCR sequences, multiple sequence alignments of GPCR families, 3D models, articles, GPCR mutation data and more.

GRAP Mutant Database (GPCRs, Family A)

A database of mutants of family A G-Protein Coupled Receptors. GRAP contains detailed description of the ligand binding and signal transductional properties.

Haemophilia B Mutation Database

A database of point mutations and short additions and deletions in the factor IX gene.

HAMSTeRS - Haemophilia A Mutation, Search, Test and Resource Site

Over the last decade there has been a dramatic increase in our understanding of the pathology of haemophilia A in molecular terms, at the levels both of nucleic acid sequence and to a much lesser extent, protein structure.

- A Review of the Molecular Genetics of Haemophilia A
- Submit a Mutation
- Search Database
- Bioinformatics of point mutations
- Factor VIII Model Structure
- Online Methods
- Links to other sites

Human HPRT database

The database contains information on the mutagen, dose, spontaneous and induced mutant fraction, base position, amino acid position, amino acid change, local DNA sequence, cell type, citation, and other items. In addition, information regarding the cause and effect of mutations affecting splicing is given.

Hypertrophic Cardiomyopathy mutation database

Familial hypertrophic cardiomyopathy is a genetic disorder associated with defects in the sarcomere.

LDLR Mutation Database

Mutations in the LDL receptor gene (LDLR) cause familial hypercholesterolemia (FH), a common autosomal dominant disorder. The LDLR database is a computerized tool that has been developed to provide tools to analyse the numerous mutations that have been identified in the LDLR gene.

Long QT syndrome database

Long QT syndrome (LQTS) is a heart disease manifesting itself by a prolonged QT interval on the ECG and clinically by a propensity for tachyarrhythmias, causing syncope and sudden cardiac death.

Marfan Database

The Marfan database is a software that contains routines for the analysis of mutations identified in the FBN1 gene that encodes fibrillin-1. Mutations in this gene are associated not only with Marfan syndrome but also with a spectrum of overlapping disorders.

MutRes - List of Mutation Resources

MutRes is a public list of databases, websites, programs and people related to collection and computational analysis of mutations. MutRes relies on mutation database community for its accuracy of data. If you know unlisted resources or want to add to an existing entry, please use the MutRes Web submission form. MutRes is made available through Thure Etzold's Sequence Retrieval System (SRS). It allows full text searching and instant hypertext linking to Web resources.

Neuronal Ceroid Lipofuscinoses (NCL) Mutations

Published mutations and polymorphisms in the NCL genes together with unpublished data included with permission.

PAH Genes and alleles (PAHDB)

Mutation data were collated from both published articles and personal communications of 80 investigators from the PAH Mutation Analysis Consortium in 26 countries. Searchable fields of the database available to users are: mutation name, polymorphic haplotype, population, geographic location, gene region, codon number, mutation type, substitution, phenotype, and author's name.

Human p53 database

The database contains information on the cancer type, loss of heterozygosity, base position, amino acid position, amino acid change, local DNA sequence, citation, and other items.

p53 gene mutations

- p53 databases
- p53 mutation analysis
- polymorphism
- p53 Story
- Anatomy of the p53 gene
- A PC database
- P53 monoclonal antibodies
- P53 phylogeny
- Other p53 sites
- Link to the next p53 workshop

Somatic p53 mutations in human tumors and cell lines.

The p53 mutation database contains information on all missense mutations and small deletions reported in human p53 reported in peer-reviewed literature. It does not contain information on p53 mutations in animals nor data on human tumors with no p53 mutations.

Database of germline p53 mutations

A comprehensive database covering all published cases of germline p53 mutations. The current version lists 580 tumours in 448 individuals belonging to 122 independent pedigrees. The database describes each p53 mutation (type of the mutation, exon and codon affected by the mutation, nucleotide and amino acid change), each family (family history of cancer, diagnosis of Li-Fraumeni syndrome), each affected individual (sex, generation, p53 status, from which parent the mutation was inherited) and each tumour (type, age of onset, p53 status-loss of heterozygosity, immunostaining). Each entry contains the original reference(s).

p53link - P53 database integration

Several p53 mutation databases are available in the net. There is no cross-linking between them and consequently it is impossible to know what is the non-redundant set of known mutations in p53. The goal of p53link is to create links between various p53 databases.

PAX2

The Human PAX2 Allelic Variant Database.

PAX6 mutation database

Contains data on human PAX6 mutations.

Schindler Disease

Mutations in the a-N-Acetylgalactosaminidase Gene Causing Schindler Disease

VHL Mutation Database

VHL is a tumor suppressor gene localized on chromosome 3p25-26. Mutations of the VHL gene were described at first in the heritable von Hippel-Lindau disease and in the sporadic Renal Cell Carcinoma (RCC). More recently, VHL has also been shown to harbor mutations in mesothelioma and small cell lung carcinoma.

VMD2 Mutation Database

Sequence, mutation and polymorphism data on the VMD2 gene.

von Willebrand Factor (vWF) Database

Databases of point mutations, insertions, deletions, and polymorphisms found in von Willebrand Factor.

- Mutations database
- Polymorphisms in the vWF gene and pseudogene
- Intron 40 polymorphism
- References etc.

WS-associated WRN mutations

Werner syndrome (WS) is one of a group of human genetic diseases that have recently been linked to deficits in cellular helicase function. We review here the structure and expression of the WRN locus, and the spectrum of WS-associated WRN mutations. The organization and potential functions of the WRN protein are discussed, as are potential mechanistic links between mutational inactivation or loss of WRN and pathogenesis of the WS clinical and cellular phenotypes.

Mouse Genome Informatics (MGI)

MGI is produced by the Jackson Laboratory, in Bar Harbor, Maine. This is a copy at the HGMP for public use. MGI is a comprehensive database of genetic information on the laboratory mouse. It includes:

- References
- Genetic Markers and Mouse Locus Catalog (MLC)
- Probes
- PCR Primers
- Mammalian Homologs
- Maps and Mapping Data
- Combined mouse/human phenotypes (MLC/OMIM)

Plus information on:

- Chromosome Committee Chairs
- Map Manager
- Inbred Strains
- Informatics ftp server
- Mouse Nomenclature Rules and Guidelines

The dysmorphic human and mouse homology database

This consists of three separate databases of human and mouse malformation syndromes together with a database of mouse/human syntenic regions. The mouse and human malformation databases are linked together through the chromosome synteny database. The purpose of the system is to allow retrieval of syndromes according to detailed phenotypic descriptions and to be able to carry out homology searches for the purpose of gene mapping. Thus the database can be used to search for human or mouse malformation syndromes in different ways:-

- By specifying specific malformations or clinical features, or chromosome locations.
- By Homology.
- By asking for human syndromes located at a chromosome region syntenic with a specific mouse chromosome region (and vice versa from human to mouse).

The Whole Mouse Catalog

This serves as a central place to find resources of particular interest to scientific researchers using mice or rats in their work. In addition to a somewhat exhaustive listing of especially useful internet resources, you'll also find conference announcements and other information.

- Genomics (physical, cytogenetic, genetic)
- Genetic Nomenclature and Standardization
- Strains and Strain Development
- Transgenic and Targeted Mutant Animals
- Cell Lines and Cell Culture
- Development
- Anatomy
- Physiology
- Laboratory Animal Suppliers
- Veterinary Resources and Animal Husbandry
- Software
- Books, Guides, News Groups, Conferences and Legal Issues

HGMP-RC Fugu Project

The MRC has granted an award to Sydney Brenner and Greg Elgar through the HGMP-RC to generate a landmark map of the Fugu genome. The Fugu fish has essentially the same number of genes as the human genome, although its genome size is only approx. 400 Mb. Data obtained to date is consistent with an overall 8x compression of the Fugu genome compared to the human. Isolation of Fugu homologies to human genomic regions thus facilitate gene finding.

Cosmids from the Fugu library will be sequenced using a scanning approach. The data arising from other groups using the library, and returning the data to the Resource Centre, will be integrated into the database.

The Zebra Fish Server

This gives information on the genetics, breeding and care of zebra fish.

Zebra Fish Information

This gives links to useful things about Zebra fish genetics and care and breeding of Zebra fish.

Tilapia Genome Project

A first-generation genetic map for tilapia, a group of fishes important to aquaculture around the world. Microsatellite markers consisting of a variable number of dinucleotide repeats have been isolated from an enriched genomic DNA library. These markers will be useful for characterizing the genomic composition and level of inbreeding of commercial tilapia strains, many of which have been derived by hybridization of several related species of *Oreochromis*. The genetic map will facilitate the improvement of strains with respect to traits of commercial importance, such as growth rate and flesh quality, through marker-assisted selection.

MEDAKAFISH

In recent years, the medaka (*Oryzias latipes*) has come to be widely used as a laboratory animal in various fields in biology, especially in developmental biology and genetics. Its relatively short life cycle, capacity to reproduce, and ease of breeding are chiefly responsible for its popularity in these fields.

Rivulus marmoratus

The killifish *Rivulus marmoratus* is the only vertebrate known to reproduce by virtually obligate self-fertilization and whose natural populations are generally comprised of homozygous clones. A webpage discussing the biology of this organism in some detail, including a review article, bibliography, faqs, and some illustrations, is now available.

SheepMap

The purpose of SheepBase is to provide an up-to-date compilation of published data from sheep genome mapping projects. Information is presented using the WWW interface, where both physical and linkage maps of the sheep genome are available, together with information on individual loci and associated references.

Centre for Animal Biotechnology (University of Melbourne)

- Australian Sheep Gene Mapping Meeting Notes
- Updated Sheep Map - ISAG 1996 Poster
- Updated Sheep Map Information
- 1996 ISAG Comparative Mapping Workshop Notes
- Sheep Gene Map - Comparisons with Other Species
- Sheep Marker Details - Chromosome Order
- Sheep Marker PCR Conditions
- List of Markers Mapped by CAB
- Diagram of Markers Mapped by CAB
- Comparative Immune Map (text)
- Links to Genome Sites for Other Species
- CAB Gene Mapping Tips
- Links to Other Useful Sites

GOATMAP

The INRA GOATMAP Database holds data on the goat genome.

Roslin PiGMaP Data

This gives details of the PiGMaP project at the Roslin Institute.

USDA Swine Genome Map

This gives chromosome maps and linkage group data.

NAGRP Pig Gene Map

This gives details of pig mapping coordinated by the NAGRP.

- Details of PIGBASE (Pig Gene Mapping Database).
- Physical Maps.
- USDA/ARS MARC database.

Chicken Genome Map

This gives details of the ChickMap project at the Roslin Institute.

- ChickMap - Information about the Chicken Gene Mapping Effort

- ChickGBASE - The genome database of the chicken
- The latest map information (genetic and physical), microsatellite & genotype submissions

The U.S. Poultry Gene Mapping Homepage

The Chicken Genome Project Homepage at Iowa State University

NAGRP Chicken Gene Map

- Members of International Poultry Gene Mapping Community - Searchable Mail List
- Poultry Genetic Maps
- Microsatellite Marker Information and Available Primer Kits
- CHICKGBASE
- Crittenden reference database
- Inventory of cDNA and Genomic Clone Libraries

Cattle Genome Database (CSIRO Molecular Animal Genetics Centre)

Welcome to the Cattle Genome Database hosted at the CSIRO Molecular Animal Genetics Centre, Brisbane, Australia. The CGD is part of an international collaboration to map the Bovine genome.

- Interactive Search for Loci
- Quantitative Trait Loci
- Citations and Contacts
- Linkage Maps
- Latest Maps Contributors Only
- Comparative Mapping
- Other Genome Resources
- Hints and Shortcuts Revised Features

University of Illinois NCSA Biotechnology Center

- Immunogenetics of Bovine Leukemia Virus (BLV) Infection in Cattle
- Gene Mapping
 - Linkage Map
 - Marker Table
 - The Illinois Reference/Resource Families (IRRF)
 - The Dairy Bull DNA Repository (DBDR)

Bovine Genome Database (NAGRP)

- U.S. Bovine ARKDB
- Bovine and Mouse on Human Comparative Maps
- Human and Mouse on Bovine Comparative Maps

BOVMAP - INRA

The goal of the BOVMAP database is to offer to the scientific community working on the bovine genome a useful and practical tool to follow the rapid mapping progress. As such, BOVMAP contains information on loci, alleles, genetic and physical maps,

polymorphisms, homologies (comparative mapping), probes, primers and bibliographic references. At the moment near 900 loci, 500 polymorphism and 360 bibliographic references are registered.

Cattle Genome Map

This gives the cattle genome marker tables and a chromosome map.

The Dog Genome Project

The dog genome project is a collaborative study involving scientists at the University of California, the University of Oregon, and the Fred Hutchinson Cancer Research Center aimed at producing a map of all of the chromosomes in dogs, which can be used to map the genes causing disease and those genes controlling morphology and behavior. Different dog breeds are distinguished by varieties of morphologies and behaviors that exceed the range of variation in any other species on earth.

The FHCRC Dog Genome Project

The Dog Genome Project at the Fred Hutchinson Cancer Research Center is working to develop resources necessary to map and clone canine genes.

- Dog Genetic Markers. Primer sequences and other information about the Canine Microsatellite markers that comprise the bulk of the canine linkage map.
- Canine Linkage Map. The current Canine Linkage Map. Order and spacing of mapped markers
- Canine-Rodent Hybrid Cell Lines. Syntenic groups of microsatellites and genes
- Software used to develop the Canine Linkage Map
- Links to other Dog Genome pages, and canine and genome information sources
- References to publications in the field of Canine research.

DogMap

DogMap is an international collaboration between 34 labs from 17 different countries towards a low resolution canine marker map under the auspices of the International Society for Animal Genetics (ISAG). The map under development should achieve a resolution of about 20 cM and some of the markers should be mapped physically. The participants have agreed to use microsatellites as markers on a common panel of reference families which will provide the backbone of the marker map. It is foreseen to also include type I markers in the mapping effort and to produce cosmid derived microsatellites for physical mapping. For this purpose part of the effort focuses on the standardization of the canine karyotype. Special attention is paid to hereditary diseases where efforts are under way to establish resource families either by collecting families or by specific breeding.

Canine Genetics Research

- Breed Specific Database
- International Canine Microsatellite Workshop
- The Center for Companion Animal Health (CCAH) Akita Immunogenetics project

Horse Genetics

Information on horse genetics, gene mapping and horse typing services at the Veterinary Genetics Laboratory at Davis University.

- Gene mapping
- Coat Color
- Cytogenetics
- Genetic Disorders
- Genetic Marker Reports
- Search Horse Genetics

HORSEMAP

The INRA HORSEMAP Database holds data on the horse genome.

ROOBASE

Kangeroo genome database.

JGBASE - The Animal Genome Database

Contains records in various tables about pig, cattle, horse, mouse, human, rat and sheep.

3.2.2.4 Summary

Biology is a huge science with the wild inflow of data that has to be collected, controlled and maintained with efficiency. This need gives rise to the importance of databanks in biology. There are now a wide of biological databanks available to aid the study of biology. Of them are the genomic databanks that hold or contain the data related to the genomes of various organisms present on the earth. A variety of the genomic databanks are listed in this chapter that help know them for the best use of human research.

3.2.2.5 Model questions

1. What is a databank? What is the need and purpose of databanks in biology?
2. Name a few Genomic Databanks along with their significance
3. Explain the significance of GDB?

References

[http:// www. ncbi. nih. nlm. gov/](http://www.ncbi.nlm.nih.gov/)
<http://www.cbi.pku.edu.cn/mirror/GenomeWeb/>
<http://www.cbi.pku.edu.cn/mirror/GenomeWeb/vert-gen-db.html>
<http://gdbwww.gdb.org/gdb/>

Lesson 3.2.3**Metabolic Pathway Databanks****Objective**

- 3.2.3.1 Introduction**
- 3.2.3.2 Metabolic Pathway Databases**
 - 3.2.3.2.1 Kegg**
 - 3.2.3.2.2 Ecocyc And Metacyc**
 - 3.2.3.2.3 Brenda**
 - 3.2.3.2.4 Emp**
 - 3.2.3.2.5 Wit/Ergo**
 - 3.2.3.2.6 Expasy-Biochemical Pathways**
 - 3.2.3.2.7 Enzyme**

Objective

- To know about the different pathway databases available and the characteristics of data stored in them.

3.2.3.1 Introduction

Enormous amounts of data result from genome sequencing projects and new experimental methods. Within this tremendous amount of genomic data 30-40 per cent of the genes being identified in an organism remain unknown in terms of their biological function. As a consequence of this lack of information the overall schema of all the biological functions occurring in a specific organism cannot be properly represented. To understand the functional properties of the genomic data more experimental data must be collected. A pathway database is an effort to handle the current knowledge of biochemical pathways and in addition can be used for interpretation of sequence data. Some of the existing pathway databases can be interpreted as detailed functional annotations of genomes because they are tightly integrated with genomic information. However, experimental data are often lacking in these databases.

Databases developed in the last decades can be classified into different categories, including genome databases, protein databases, enzyme databases, pathway databases, literature databases and some very specialised databases. This classification of databases is often based on their biological content. Although the content of the databases is mostly restricted to specific biochemical compounds or functions, a lot of overlap occurs. For example protein sequence information is listed in genome databases and functional data of corresponding enzymes can be found in protein and pathway databases.

A biological pathway database is a database that describes biochemical pathways, reactions, enzymes that catalyze the reactions, and the substrates that participate in these reactions. A pathway genome database (PGDB) integrates pathway information with information about the complete genome of various sequenced organisms. Two of the popular PGDBs available today are the Kyoto Encyclopedia of Genes and Genomes (KEGG) and MetaCyc.

A PGDB schema describes pathways in terms of these five biological entities:

1. Metabolic overview: the union of all described pathways;
2. Pathways: the individual pathways;
3. Reactions: the reactions within these pathways;
4. Compounds: the compounds that participate in these reactions; and
5. Gene products: enzymes that form a subset of gene products and that catalyze these reactions.

The various genes in the nucleus of the cell that constitute the genome are transcribed to produce gene products. A map of various genes and their chromosomal loci form the genomic map. The genes, their products, and the genomic map form the genomic component (enclosed in the smaller dotted rectangle in Figure 1.1). The metabolic overview, pathways, reactions, compounds, and the gene products together constitute the pathway component (enclosed in the larger dotted rectangle in Figure 1.1). Types of pathways that can be incorporated in this scheme include biosynthesis, degradation, energy metabolism, and intermediary metabolism for compounds such as amino acids, carbohydrates, fatty acids, nucleotides, and enzyme cofactors. For our purposes, a genome consists of the following three biological entities:

1. Genomic sequences;
2. Constituent genes; and
3. Gene products that link genes and pathways.

It should be noted that PGDBs typically describe all known or predicted gene products and not just enzymes.

The proliferation of biological databases in general raises several questions for the life scientist. Which of these databases is most accurate, most current, or most comprehensive? Do they have a standard format? Do they complement each other? Overall, which database should be used for what purpose? If more than one database is deemed relevant, it is desirable to have a unified database containing information from all the short listed databases. While XML based pathway data exchange standards such as BioPAX and SBML are emerging, these do not address the basic problems such as inconsistent nomenclature and substrate matching between databases in the unification of pathway databases. a model organism database (MOD) is a candidate, such as EcoCyc for *Escherichia coli*.

3.2.3.2 Metabolic Pathway Databases

This section introduces metabolic pathway databases that store information about metabolic pathways, reactions, reactants, and the relationships among genes, enzymes, and reactions.

3.2.3.2.1 KEGG

The Kyoto Encyclopedia of Genes and Genomes (KEGG) is a suite of databases and associated software, integrating the current knowledge on metabolic networks (the PATHWAY database), genomic and proteomic information (the GENES/SSDB/KO databases), and information about chemical compounds and reactions (the COMPOUND/REACTION databases).

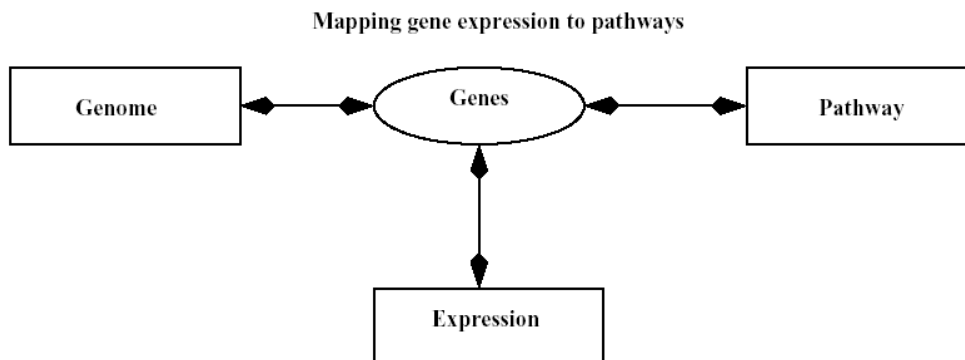


Figure 2.1: The KEGG Database organization.

The KEGG system is organized into tightly connected databases as follows:

1. EXPRESSION database contains microarray gene expression data and information about individual spots.
2. GENES: Gene sequence and information on genes of all completely sequenced organisms and some partially sequenced organisms.
3. LIGAND: Information about over ten thousand chemical compounds, enzyme molecules, and enzymatic and non-enzymatic reactions.
4. PATHWAY: Diagrams of metabolic/regulatory pathways.

The chemical structures of compounds are stored as GIF images and as 2D coordinates stored in an MDL-MOL file (a specific file format for molecular structures from MDL Information Systems Inc.). Either of the two file formats can be used to launch an appropriate drawing application. In the flat file downloadable version, there are files of 7 different types (files with extensions: orth, html, gif, gene, coord, conf, and tab) organized into species-specific directories. The database organization used in KEGG for mapping gene expression data onto pathway diagrams is shown in Figure 2.1.

In addition to the above databases, KEGG provides many links to other databases that are integrated within its database retrieval system called DBGET. The KEGG genes database contains all publicly available nucleotide sequences and their functional annotations. A sample entry from the GENES database is shown in Figure 2.2.


```

ENTRY      351          CDS      H.sapiens
NAME       APP
DEFINITION amyloid beta (A4) precursor protein (protease nexin-II, Alzheimer
           disease) [SP:A4_HUMAN]
CLASS      Human Diseases; Neurodegenerative Disorders; Alzheimer's disease
           [PATH:hsa05010]
POSITION   21q21.3
DBLINKS    LocusLink: 351
           GDB: 119692
           OMIM: 104760
           NCBI: 4502167
CODON_USAGE
           T          C          A          G
T   9  12  3  10  10  7  5  1  7  13  0  1  5  13  0  9
C   7  12  2  22  14  9  10  2  14  11  7  29  3  11  8  5
A  10  13  1  24  5  27  14  4  12  19  15  26  9  3  7  3
G  15  10  8  32  15  30  15  3  30  20  44  48  9  12  8  9

AASEQ      770
MLPGLALLLLAAWTARALEVPTDGNAGLLAEPQIAMFCGRLNMHMNVQNGKWDSDPSGKTC
IDTKEGILQYCCQEVYPELQITNVVEANQPVTIQNWCKRGRKQCKTHPHFVIPYRCLVGEFVSD
ALLVDPKCKFLHQERMDVCETHLHWHTVAKETCSEKSTNLHDYGMLLPCGIDKFR AA sequence
cut here.

NTSEQ      2313
atgctgccccggtttggcactgctcctgctggccgctggacggctcgggctggaggtagccactgat
ggtaatgctggcctgctggctgaacccagattgccatggttctgtggcagactgaacatgcacatgaat
gtccagaatgggaagtgggattcagatccatcagggacaaaacctgcattgataccaaggaaggcatc
ctgcagtattgccaagaagtctaccctgaactg NT sequence cut here.

```

Figure 2.2: A sample entry from the KEGG database.

The KEGG data is in accordance with two international standards IUPAC and IUBMB. The KEGG web service provides access to information such as: metabolic or regulatory pathways, pathways for a specific species, for specific enzymes, compounds, or reactions. However, it does not provide complex query options. For example, the user cannot ask for all enzymes in species X, Y, and Z participating in pathways a, b, or c. KEGG provides several tools for accessing pathways.

Access to the data

KEGG's main menu offers several different ways to enter the pathway systems: one can select an organism from a list of complete or partially sequenced organisms, or select a pathway from the classification of pathways (subdivided into metabolic and regulatory pathways), or search for a specific compound or enzyme in the LIGAND database or for a gene in the GENE database. The query interfaces do not allow complex queries; for example, a query for all known enzymes that use a defined substrate.

The general overview of all pathways using a colour-code according to the classification of pathways enables the initial entry of the program. This allows access to specific pathway classes, eg carbohydrate metabolism. A window opens which represents links to all specific pathways within carbohydrate metabolism.

Data representation

Pathways in KEGG are classified according to the chemical structures of their main compounds, eg carbohydrates, lipids, amino acids. All specific pathway maps and overviews are manually drawn pictures where pathway maps consist of links to specific information about compounds, enzymes and genes. Pathway maps contain all known reactions catalysed by proteins/enzymes derived from gene products. Non-enzymatic reactions are not always included in that system. However, in some cases non-enzymatic reactions are shown within a pathway map but do not have links to detailed information of this specific reaction. Reactions within the pathway maps do not represent side compounds, eg ATP (adenosine triphosphate) or NADH (reduced nicotinamide adenine dinucleotide). Pathway maps are also linked to some other related pathways connected by their contributing compounds. This allows the user to get an overview about connections to other pathways, eg one main final product of glycolysis - pyruvate - is used in the different pathways of amino acid metabolism. Unfortunately not all possible connections are given.

As well as the reference pathway the user can select pathway information of a specific organism. Within the reference pathway all the enzymes that are known to be expressed in the specific organism (found by homology search) are highlighted in green. Furthermore other selections for specific information are also available,¹⁰ eg highlighting of all enzymes with identified 3D structure, enzymes with sequences in the SWISS-PROT, GenBank or PIR databases, and enzymes that have a link to genetic diseases in the OMIM (Online Mendelian Inheritance in Man) database.

Genome maps are represented as graphics that give information about gene positions and their relationship with the pathways. Gene catalogues contain hierarchical texts that include all known genes for each organism, listed according to the pathway classification. Information about orthologous genes are stored in tables containing the information of a conserved; functional unit in a pathway, a comparative list of genes for the functional unit in different organisms, and the positional information of genes clustered in each genome.

Molecule catalogues are represented by hierarchical texts containing macromolecules (including proteins and RNAs) and small chemical compounds. These data are based on classifications of enzymes and chemical compounds. Enzymes are classified based on IUPAC/ IUBMB recommendations and additionally according to PIR superfamilies, SCOP (Structural Classification of Proteins) 3D-folds or PROSITE (database of protein families and domains) motifs. The pathway query result shows all pathways containing a given enzyme or compound but offers no graphical representation. Organism-specific information is also not available within such results. Searching for a pathway by selecting a specific organism does not provide information about the enzymes available but only links to the gene information related to that organism. A comparison of pathways of two or more organisms cannot be implemented.

Enzyme pages contain the following information: names, EC classification according to IUBMB, reactions and their participants (substrates, products, cofactors, inhibitors,

etc.), pathways, related genes in organisms and links to diseases (OMIM), motifs (PROSITE) and 3D structures (PDB).

Compounds are represented with names, chemical and structural formula, pathways in which the compound occurs and enzymes that catalyse a reaction containing the compound as participant.

Reliability/inconsistency

KEGG does not contain all possible reactions catalysed by an enzyme. When comparing reaction equations with the lists of substrates and products, inconsistencies sometimes arise, eg the substrate list of pyruvate kinase (EC 2.7.1.40) contains more compounds than those listed in the reaction equations. Often reactions catalysed by an enzyme, eg alcohol dehydrogenase (EC 1.1.1.1), with a wide substrate specificity are given as general reaction equations, which means loss of detailed information concerning the correlating products of each substrate. Moreover, enzymes are represented with their substrates and products, but reaction equations and pathway information are missing, eg peptidases.

Within compounds some errors and inconsistencies arise. For example sometimes one compound ID is given for two different compounds (eg C00023 for both Fe²⁺ and Fe³⁺) or different compound IDs are given for the same compound (C02038 and C02156 for glycyl-peptide or C04230 and C01174 for 1-acyl-glycero-3-phosphocholine among others).

Additional tools

KEGG offers some tools for querying within different maps and also methods to compute with KEGG data. Tools for searching enzymes (EC numbers - names not possible), compounds (compound ID - names not possible) or genes (gene name or accession number) are available. Within genome maps queries for gene positions or homologous gene clusters are available as well as comparisons of two genome maps for exhaustive search of homologous gene clusters. Tools for colouring and thereby highlighting the genes in the pathway or genome map make the system user-friendly and enable a quick overview on the result of the query.

One unique feature of the KEGG system is the creation of a pathway between two given compounds.¹⁰ As a starting point, two compound IDs, the parameter level and hierarchy of relaxation have to be defined. Unfortunately there is no help function or documentation about the parameters. Based on binary relations two algorithms (Dijkstra/Floyd) are used to find the shortest way between two compounds. In such cases where a reaction has two or more substrates the implementation is not able to distinguish between different substrates and their correlating products or compartments and transport mechanisms. Pathways are created based on one main substrate without paying attention to side substrates or products, compound/enzyme locations and cofactors. Results of the pathway creation contain only the compound ID and EC number. For users who are not familiar with the compound IDs used in KEGG it is very time-consuming to understand the resulting pathway since there is no graphical representation of the search results. This tool does not apply organism-specific searches and creates pathways containing all possible reactions regardless of whether the specific enzyme has been identified in the specific organism or not.

Highlighting these identified enzymes and proven pathways among them would be very helpful.

Automatic reconstruction of organism-- specific metabolic pathways can be accomplished by matching the genes corresponding to the enzymes in the gene catalogues with the enzymes on the reference pathway diagrams.

3.2.3.2.2 EcoCyc and MetaCyc

Karp, *et al.*] describe EcoCyc as an organism-specific pathway and genome database that includes the metabolic and signal-transduction pathways of *E.coli K-12*, its enzymes, its transport proteins, and its mechanisms of transcriptional control of gene expression. EcoCyc is freely available on the web. The MetaCyc database is based on the same database schema as Ecocyc. The MetaCyc database initially contained only the information in the Ecocyc database and subsequently extended with information about more than hundred different species. These databases contain extensive references to literature citations on enzymes and reactions whenever available. The species attribute for each pathway in the database lists all species in which a particular pathway has been cited to be observed in the literature. Hence, absence of a species reference for a pathway does not imply that the pathway does not exist in that species.

The MetaCyc database contains information about all enzymes, reactions, and metabolic pathways of a variety of other organisms with a microbial focus. Since the design of the MetaCyc database and the pathway tools software are completely based on the Ecocyc design, the two resources essentially go together. BioCyc is a collection of several MODs, including EcoCyc, available at the URL. MetaCyc is not merely a collection of related reaction steps from different organisms but is a set of complete pathway information elucidated in specific organisms. Maranas and Burgard note that MetaCyc also provides a wealth of literature citations and in-depth commentary on each enzyme and pathway. All these databases can be accessed using a software environment called pathway tools, which provides querying, editing, and visualization capabilities. The top level hierarchy that forms the basis of the MetaCyc design is shown in Figure 2.3.

A limitation is that one database cannot possibly encompass the complete metabolic picture of every sequenced organism. Following are the intended uses of MetaCyc:

1. As a resource for analysis of microbial genomes at the level of individual genes and an accurate reference for inferring gene function by sequence similarity,
2. MetaCyc describes the subunit structures of many enzymes, and therefore can be used as training or validation datasets for algorithms that depict protein-protein interactions,
3. MetaCyc can serve as a test set for algorithms that infer genetic networks from gene expression data,
4. For studies of pathway evolution
5. As an aid in teaching biochemistry

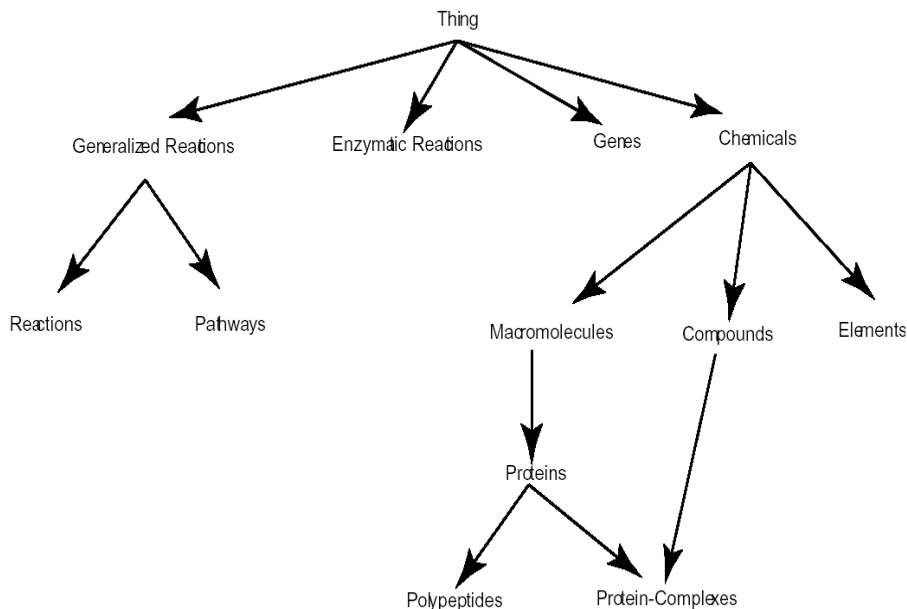


Figure 2.3: Top of the class hierarchy for the MetaCyc Knowledge Base (KB).

MetaCyc aids in the process of metabolic engineering through genetic engineering which involves:

- Inserting a new enzyme or pathway into an organism
- Replacing an existing enzyme or pathway with a substitute, or
- Removing an enzyme or pathway.

The installable application has a number of capabilities beyond those available through the BioCyc.org website: More query options, particularly on the full metabolic overview, such as

- Compare full metabolic maps of two or more organisms
- Find compounds, reactions, and genes on the overview
- Highlight enzymes controlled by a specified transcription factor
- Customizable partial gene maps
- Show an object in a species
- Programmable (Lisp and Perl-based APIs with documentation)

Paley, *et al.* designed the pathway tools software as a reusable software tool for creating model-organism specific databases (MODs) such as EcoCyc for the species *E.coli k-12*. The pathway tools software allows the EcoCyc database to be queried by providing a multitude of query operations and visualization tools. Both EcoCyc and MetaCyc are contained in a bigger collection of model organism databases called BioCyc. The majority of databases in the BioCyc collection are computationally derived databases that are generated by a program called PathoLogic. PathoLogic predicts the metabolic pathways of an organism from its genome; in that sense, the metabolic pathways in such databases are computationally derived, in contrast to the literature-derived pathways in the EcoCyc and MetaCyc databases. The input required by PathoLogic is an annotated genome for the organism, such as in the form of a Genbank entry. The output produced by PathoLogic is a new pathway/genome database for the organism. For example, the AgroCyc pathway/genome database for the bacterium *Agrobacterium tumefaciens* was created by

the company SRI International using the PathoLogic program. In general, most of the metabolic pathways in a computationally derived database are predicted computationally, but in some cases, pathways that have been observed experimentally in the organism are added manually to the database. Additional unknown pathways are likely to be present in each organism. Hence, such databases have to be interpreted with caution by the life scientists.

3.2.3.2.3 BRENDA

BRENDA is a primary collection of enzyme functional data. BRENDA is maintained and developed at the Institute of Biochemistry at the University of Cologne. Data on enzyme function are extracted directly from the literature by qualified scientists. Formal and consistency checks are automated by computer programs, each data set on a classified enzyme is checked manually by at least one biologist and one chemist.

3.2.3.2.4 EMP

The Enzymes and Metabolic Pathways (EMP) database claims to be a unique and comprehensive electronic resource of biochemical data. EMP contains information that is indispensable in the analysis and mathematical simulation of metabolic pathways, reaction mechanisms, rate laws and a very wide spectrum of numeric data. The database is being constructed at Pushchino, Moscow region, Russia. It contains about 30,000 records derived from 15,000 original experimental journal publications.

3.2.3.2.5 WIT/ERGO

Overbeek, *et al.* develop WIT (What Is There?), a WWW-based system to support the curating of function assignments made to genes and the development of metabolic models. It is described as 'an interactive metabolic reconstruction on the web'. It uses data from the EMP family of databases and includes over 40 genomes as of year 2002. Its main purpose is to support comparative analysis of sequenced genomes and to generate metabolic reconstructions based on chromosomal sequences and metabolic modules from EMP. It also includes transport and signal transduction pathways.

The WIT web interface includes four components:

1. A functional overview that provides a hierarchical listing of pathways present in a chosen organism. But, this hierarchy is not consistent with hierarchies found in other databases.
2. Open reading frame pages that provide protein information and links to similar proteins.
3. Pathway pages that provide a list of proteins participating in a pathway in a particular species.
4. Assertion table that lists the presence or absence of a particular pathway member in each of the species.

3.2.3.2.6 ExPASy-biochemical pathways

The ExPASy (Expert Protein Analysis System) proteomics server maintained by the Swiss Institute of Bioinformatics (SIB) is dedicated to the analysis of protein sequences and structures as well as 2-D poly acrylamide gel electrophoresis (PAGE). It provides a digitized version of the complete metabolic map being maintained by Roche Applied

Science. This map includes the cellular and molecular processes and links to the ENZYME database.

EC Number (w)	Meaning
1	oxidoreductase
2	transferase
3	hydrolase
4	lyase
5	isomerase
6	ligase

Table 2.1: Top level classification of chemical reactions based on EC.

3.2.3.2.7 ENZYME

ENZYME is a repository of enzyme nomenclature. ENZYME employs the recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology (IUBMB) and contains those characterized enzymes having an EC (Enzyme Commission) number. The Enzyme Commission is an *ad hoc* committee formed in the 1950s to tackle the many difficulties arising from the uncontrolled naming of the rapidly increasing number of known enzymes. Some names were misleading, while others conveyed little or nothing about the nature of the reaction catalyzed. Enzymes catalyzing similar reactions sometimes had names suggesting they belong to different groups, while some enzymes of different types had been placed in the same group. For example, the pyrophosphorylases included both glycosyltransferases and phosphotransferases. In other cases, a name that had been well established for many years with a definite meaning, such as the term synthetase, was later employed with different meanings, thus causing confusion.

An EC number is of the form EC w.x.y.z where w, x, y, and z are integers. The main classification based on EC numbers is shown in Table 2.1. IUBMB provides an official listing of EC numbers and their meanings. In particular, ENZYME provides EC number, recommended names, alternative names (if any), catalytic activity, cofactors (if any), pointers to the Swiss-PROT protein sequence entries(s) that correspond to the enzyme (if any), and pointers to human disease(s) associated with a deficiency of the enzyme (if any). Each entry in the ENZYME database is a bioreaction linked to SWISS-PROT sequences for enzymes that catalyze the reaction.

Summary

The databases described above are currently not complete and will not be complete in the near future. The KEGG database deserves first place with respect to user friendly access and the amount of data curated. Its nomenclature is consistent with international standards such as IUBMB and IUPAC. The design of the database and the classification of pathways, substrates, and reactions are in agreement with the expectations of the biologist. Hence, KEGG is the first choice for a typical user. For users interested in biochemical pathways of the *E.coli* bacterium, the EcoCyc system is the most comprehensive information resource on this model organism. The system requires a JavaScript-enabled web browser to provide direct links to the literature and the experimental data. With reference to the graphical representation of the metabolic maps, ExPASy provides more advanced and contextual information, such as subcellular location

and interconnections between different cellular processes. Hence, ExPASy may be the choice of an advanced user. The individual metabolic pathways of KEGG are more consistent, clear, and simple to follow, though a complete metabolic map is not available. Some of these databases are commercializing and require license agreements for use. However, such license agreements are typically available free of cost to academic and nonprofit organizations. For example, MetaCyc and EMP require such license agreements. This is an indication of the potential demand for the information that these databases disseminate. In summary, we note the following points about these databases.

1. Biological data are distributed worldwide and are mainly available as a web service or as downloadable flat files.
2. No standard nomenclature is followed for various biological entities.
3. The schema design and classification schemes of each database is different.
4. These databases are updated independently and often.
5. The user interface and manipulation tools for the pathways are significantly different in these databases.
6. Errors and inconsistencies are to be expected and accommodated (they are not perfect).

Model Questions

1. Write a brief note on metabolic databases?
2. Explain KEGG briefly?

References

1. A Methodology for the Unification of Metabolic Pathway Databases by Harsha K. Rajasimha
2. Introduction to Bioinformatics by Arthur M. Lesk.

AUTHOR:

B.M.REDDY M.Tech. (HBTI, Kanpur)

Lecturer, Centre for Biotechnology

Acharya Nagarjuna University.

Lesson 3.2.4**Microbial Databanks****Contents**

- 3.2.4.1 Introduction**
- 3.2.4.2 Objective**
- 3.2.4.3 Various Microbial Databanks**
- 3.2.4.4 Summary**
- 3.2.4.5 Model Questions**
- 3.2.4.6 References**

3.2.4.1 Introduction

Microorganisms are everywhere - a largely unseen world of activities that helped to create the biosphere and that continue to support the life processes on earth. Of all the cells that make up the normal, healthy human body, more than 99 per cent are the cells of microorganisms living on the skin or in the gut, etc. This normal resident microbial population includes potential pathogens as well as organisms that help to keep the potential pathogens in check.

The evidence is fragmentary, and is obtained from three major sources:

- the fossil record, consisting of the preserved remains of organisms themselves (of limited value for microorganisms)
- geological deposits that are believed to result from biological activities
- changes in the oxidation states of sediments (e.g. banded iron formation) indicating the progressive development of an oxygenic atmosphere.

Since the explosion of the web, biologists no longer need to have accounts on centralized servers or to use email for accessing sequence collections or to run analysis programs. With a connection to the Internet and a web browser it is possible to access easily almost all existing data banks and software devoted to sequence study. Some limitations still exist in the use of such servers. A significant problem is that it is often not possible to keep track of the previous interrogations from one session to another: once the user is disconnected from the server, all his/her work is lost. Also, there are no real links between the two aspects of sequence retrieval and sequence analysis. Usually, a method can be launched only on a single sequence and not on a set of sequences previously selected by the user.

Analysis of any newly sequenced bacterial genome starts with the identification of protein-coding genes. Despite the accumulation of multiple complete genome sequences, which provide useful comparisons with close relatives among other organisms during the annotation process, accurate gene prediction remains quite difficult. A major reason for this situation is that genes are tightly packed in prokaryotes, resulting in frequent overlap. Thus, detection of translation initiation sites and/or selection of the correct coding regions remain difficult unless appropriate biological knowledge (about the structure of a gene) is imbedded in the approach.

3.2.4.2 Objective

- To know what a microbial databank would constitute of
- The acquaintance with the various microbial databanks

3.2.4.3 Various Microbial Databanks

E. coli Genome Center

The E. coli Genome Center is a laboratory of the Genetics Department, College of Agricultural and Life Sciences, at the University of Wisconsin - Madison Campus. This is the complete sequence of the E. coli K-12 genome and several of the E. coli phages. The data is being analysed.

Functional search of the known genomes using E. coli

A tool for function keyword searching. They have built a table that relates E. Coli to *Saccharomyces cerevisiae*, *Methanococcus jannaschii* and *Bacillus subtilis* using blast with a Karlin Altschul score of $< 10E-17$. This keyword searching tool will print a list of every sequence identifier that is close to the E. Coli gene of the cluster where the keyword is found.

EcoCyc: Encyclopedia of E. coli Genes and Metabolism

EcoCyc is a project to describe the genes and intermediary metabolism of the bacterium E. coli. It will describe each pathway and bioreaction of E. coli metabolism, and the enzyme that carries out each bioreaction, including its cofactors, activators, inhibitors, and the subunit structure of the enzyme. When known, the genes encoding the subunits of an enzyme will be listed, as well as the map position of a gene on the E. coli chromosome. In addition, the KB will describe every chemical compound involved in each bioreaction, listing synonyms for the compound name, the molecular weight of the compound, and in many cases its chemical structure.

GenProtEC - E.coli genome and proteome database

GenProtEC is a database dedicated to E. coli genome and proteome. It aims to provide biochemists with the most updated and consolidated information about E. coli genes and proteins resulted from both the traditional experimental research and our computational analysis.

RegulonDB - a database on transcriptional regulation in Escherichia coli

RegulonDB is a DataBase that integrates biological knowledge of the mechanisms that regulate the transcription initiation in E.coli, as well as knowledge on the organization of the genes and regulatory signals into operons in the chromosome. The operon is the basic structure used in RegulonDB to describe the elements and properties of transcriptional regulation. The current version contains information around 500 regulatory mechanisms, mainly for sigma 70 promoters.

Blast protein against the complete genomes of Saccharomyces, Methanococcus, E. coli and B. subtilis

This Blast interface allows you to search against two subsets of the available putative open reading frames of these genomes using blastp. A

Computational Functional Genomics

This contains comparative *E. coli* and Yeast Genome, Transcriptome, Proteome, Physiome and Biome data

- Comparative genome sequences
- *E. coli* in-frame genome engineering
- Gene clustering
- Yeast mRNA Abundance Data
- Motif software
- *E. coli* motifs
- *E. coli* Proteomics: Subcellular localization, abundance, protein sequence, ESI-MS, 2D Gels
- *E. coli* multiplex competition selections (phenome)

Escherichia coli WWW Home Page

We compiled the DDBJ entries that had been submitted by *Escherichia coli* genome project teams worldwide into a preliminary database from 0 min. to 100 min. It includes the sequences from 28 min. to 50min., that the Japan *Escherichia coli* genome project team determined in 1996 and registered into DDBJ by December 18th.

The *E. coli* Index

These pages contain a comprehensive guide to information relating to the model organism *Escherichia coli*.

-

TubercuList - *Mycobacterium spp.* genome data

Its purpose is to collate and integrate various aspects of the genomic information from *M. africanum*, *M. bovis*, *M. bovis BCG*, *M. canetti*, *M. microti*, and above all, *M. tuberculosis*. TubercuList provides a complete dataset of DNA and protein sequences derived from the paradigm strain *M. tuberculosis H37Rv*, linked to the relevant annotations and functional assignments. It allows one to easily browse through these data and retrieve information, using various criteria (gene names, location, keywords, etc.).

The *M. pneumoniae* Genome Project

The genome has a length of 816394 bp with a G+C content of 40.01 %. We predicted 677 open reading frames (ORFs) with an average molecular weight of 39500 kDa. Adding the number of ORFs to the amount of RNAs (5S-, 16S-, 23S-rRNA, 33 tRNAs, 4.5S RNA, 10Sa RNA and RNaseP RNA) we define 716 coding regions (88.7%) in the genome. Almost 6 % of the genome is engaged by the 4 repetitive sequences of the P1 operon of *M. pneumoniae*. The derived gene density is one gene per 1.14 kb. So far, 50 % of all ORFs/genes showed a significant sequence homology to defined ORFs/genes with known function from other bacteria.

SubtiList - *Bacillus subtilis* genome project

This is a database dedicated to the analysis of the *Bacillus subtilis* genome: SubtiList. The purpose of this database is to integrate various aspects on the genomic information of *B. subtilis*, the paradigm of sporulating Gram-positive bacteria. As such, it provides a clean dataset of non-redundant DNA sequences of *B. subtilis* (strain 168), associated to relevant annotations and protein sequences. It allows one to easily browse

through these data and retrieve information, using various criteria (gene names, keywords, location, etc.).

The data contained in SubtiList originates mainly from the *B. subtilis* genome sequencing project, but this dataset also benefits from the *B. subtilis* entries present in the EMBL/GenBank/DDBJ databanks.

Micado: MICRobial Advanced Database Organization

The database is primarily devoted to the *Bacillus subtilis* genome sequencing program. It links the genetic map of the microbe with its sequences, together with other bacteria. DNA comes from primary databanks entries, plus data from the SubtiList database.

NRSub: Bacillus subtilis Database

This provides access to a non-redundant set of DNA sequences from *Bacillus subtilis*. All the duplications from the general sequences collections have been removed and all detected overlapping sequences have been merged into contigs. Additional data on gene mapping, codon usage have been added, as cross-references with EMBL, Swiss-Prot, Enzyme, and Medline collections.

Haemophilus influenzae Rd Genome Database (HIDB)

The *Haemophilus influenzae* Rd genome is the first genome of a free living organism to be completed. This page offers access to the latest versions of the sequence data and related annotation.

Mycoplasma genitalium Genome Database (MGDB)

The *Mycoplasma genitalium* genome is the first genome of a gram positive-like bacterium to be completed. This page offers access to the latest versions of the sequence data and related annotation.

TB Genomes analysis server

The Mycobacterium tuberculosis genome analysis server.

- BLAST searches of predicted ORFS and annotated ORFS and post-blast search tools
- MYCdb web browser and data retrieval
- Three levels of query complexity/views on the Genome and its associated data
- Links to other major sites and viewers
- 'Find a gene' Using established gene names

The server is unique in that it provides an easy to use interface to browse MYCdb with inclusion of all public sequence EMBL entries of TB Genome sequences in MYCdb . The easy to use Web interface includes features to search MYCdb, retrieve files and provides a graphical view of the data.

Links for retrieved sequences allow connection to the DDBJ javaserver in Japan. Searches can be performed to retrieve genome sequence, predicted ORFS and annotated ORFS, of both the Sanger sequenced TB genome and partial leprae genome.

CyanoBase

Synechocystis/CyanoBase provides an easy way of accessing the sequence and all-inclusive annotation data through image maps, keyword searches and the gene category list.

The cyanobacterium carries a complete set of genes for oxygenic photosynthesis, which is the most fundamental life process on the earth. This organism is also interesting from an evolutionary viewpoint, for it was born in a very ancient age and has survived in various environments. Chloroplast is believed to have evolved from cyanobacterial ancestors which developed an endosymbiotic relationship with a eukaryotic host cell.

Chlamydia Genome Project

The goal of the Chlamydia Genome Project is to determine the DNA sequence of the chromosome of *Chlamydia trachomatis*, serovar D (D/UW-3/Cx), trachoma biovar, and L2/434/Bu, LGV biovar. The project is a collaborative effort involving scientists at the University of California at Berkeley and Stanford University .

Pseudomonas Genome Project

The bacterium *Pseudomonas aeruginosa* causes significant infections in humans. People with cystic fibrosis, burn victims, individuals with cancer, and patients requiring extensive stays in intensive care units are particularly at risk. Greater knowledge of the DNA sequence of the *Pseudomonas* genome will suggest directions for novel drug development and new therapeutic strategies for treating these infections.

ARCHAIC: ARCHAebacterial Information Collection

The aim of ARCHAIC is to analyze archaeobacterial genomic DNA sequences that have been determined and will be determined by ourselves and by other groups, by the same standard in a consistent way, in order to understand the overall organization of these genomes and in order to permit comparison of different species on the basis of their genomic DNA sequences.

Ureaplasma urealyticum - The Complete Genomic Sequence

- Data Analysis
- Sequence Data
- Contact Information

Pyrococcus horikoshii OT3 database

NITE first worked on a hyperthermophile found in the hot waters of the Okinawa Trench. (The microorganism can grow at high temperatures, favoring a temperature of 98C.) The DNA of this organism has 1.74 million base pairs. Among these, it is estimated that there are 2,061 genes for heat-resistant proteins and enzymes.

M. thermoautotrophicum gene classification table

- Amino Acid Metabolism
- Purine, Pyrimidine, Nucleoside and Nucleotide Metabolism
- Sugars
- Transcription and Translation
- Cellular Processes and Cofactor Metabolism

- Energy Metabolism
- RNA products
- Other

Methanococcus jannaschii Functions database

This page provides an updating of the functional content of the first completely sequenced Archaeal genome, this of *Methanococcus jannaschii*.

The WWW Virtual Library: Microbiology

This has links to many microbiology sites.

DOE Microbial Genome Initiative (MGI)

Description of the DOE Microbial Genome Initiative, including which organisms are being sequenced and who is contracted to sequence them.

Genome Information Broker for Microbial Genomes

The GIB holds information on the following genomes:

- *Saccharomyces cerevisiae*
- *Aquifex aeolicus*
- *Bacillus subtilis*
- *Borrelia burgdorferi*
- *Chlamydia trachomatis*
- *Escherichia coli*
- *Haemophilus influenzae*
- *Helicobacter pylori*
- *Mycobacterium tuberculosis*
- *Mycoplasma genitalium*
- *Mycoplasma pneumoniae*
- *Synechocystis PCC6803*
- *Treponema pallidum*
- *Archaeoglobus fulgidus*
- *Methanobacterium thermoautotrophicum*
- *Methanococcus jannaschii*
- *Pyrococcus horikoshii*

The following services and views of the data are available:

- Genomic View : displays genome information in diagram.
- Retrieve Clone : retrieves clone information.
- Retrieve ORF : retrieves ORF information.
- Retrieve Gene : retrieves gene information.
- Thumbnail sketch of this server : illustrates a brief overview of the server system.
- Genome sequence FTP service : Allows you to obtain nucleotide sequences.

TIGR Microbial Database

TIGR Microbial Database: a listing of microbial genomes completed and in progress.

HOBACGEN : Homologous Bacterial Genes Database

HOBACGEN is a database system that contains all the protein sequences of bacteria organized into families. It allows one to select sets of homologous genes from bacterial species and to visualize multiple alignments and phylogenetic trees. Thus HOBACGEN is particularly useful for comparative genomics, phylogeny and molecular evolution studies on bacteria.

Microbial Genomics

These microbial genome pages were created as a reference for the community and contain a list of current or completed eubacterial, archaeal and eukaryotic genome sequencing projects. Each main page includes the name of the organism being sequenced, which sequencing group(s) are involved in the effort, background information on the organism, and its current evolutionary location

The Archaeon *Pyrobaculum aerophilum* Genome Project

Pyrobaculum aerophilum is a rod-shaped hyperthermophilic archaeon that has recently been isolated from a boiling marine hole. The goal of the project is to complete sequencing and annotating its 2.3 Mbp genome.

Microbial genomes at NCBI

A collection of microbial genome information.

Genomes at LMB

Directories of the sequences of selected organisations.

Yeast Genome Project

Holds the complete genome of *Saccharomyces cerevisiae*.

- Text search of annotation in MIPS yeast database entries
- Text search of yeast entries in PIR
- TEXT search of yeast homologs to human ESTs
- Search of over/under represented N-mers in the *S. cerevisiae* chromosomes
- Search for yeast gene names
- xChromo - view a whole chromosome
- Get DNA fragments
- Get Protein sequences
- Protein homology search
- Genome-Browser - Compare yeast chromosomes to each other

***Saccharomyces cerevisiae* Genome Database**

The SGD project collects information for and maintains a database of the yeast *Saccharomyces cerevisiae*. This database includes a variety of genomic and biological information.

The complete sequence of *Saccharomyces cerevisiae*, strain S288C, is now public and can be retrieved from SGD (USA), MIPS (Germany), or EBI (England). Sequence searches using BLAST and FASTA are available from SGD. A summary of the yeast chromosomes is being assembled, including references, maps, and a DNA sequence

retrieval form. SGD has begun the process of integrating the newly released sequence within its database and is waiting for the standard ORF designations from the authors.

Genome Navigator: *Saccharomyces cerevisiae* Genome Index

S. cerevisiae genome is available for viewing using a Java map display tool, DerBrowser. Among other features, it allows for querying external data sources: SacchDB, MIPS, YPD and GeneQuiz.

Yeast Protein Database (YPD)

YPD is a database of Gene names and properties of the *Saccharomyces cerevisiae* proteins of known sequence.

The *Schizosaccharomyces pombe* Genome Sequencing Project

The fission yeast *Schizosaccharomyces pombe* is a unicellular ascomycete. It has a simple eukaryotic genome of approximately equivalent size to that of budding yeast, *Saccharomyces cerevisiae*, at around 14 Mb. Unlike *S. cerevisiae*, the *S. pombe* genome is spread between only three chromosomes.

Yeast gene relationships

This is a database of yeast gene relationships based on data made available by Eisen et al. (PNAS Vol 95, p14863). This database catalogues the 2500 genes analyzed and gives the closest related genes based on a probabilistic analysis.

NIH Campus Yeast Interest Group

- NIH campus Yeast Research Groups
- Meeting Schedules
- The NIH Yeast Interest Group Mailing List
- NIH resources for the Yeast Community

Mycological Resources

The Mycology page gives details of many mycology sites, resources and information.

The *Neurospora* Genome Project

The *Neurospora* Genome Project (NGP) represents an effort to obtain partial or complete nucleotide sequences from a large number of cDNA clones derived from conidial, mycelial, and perithecial libraries of *N. crassa*.

- blast searches
- project description
- *Neurospora* information

Candida albicans

Information on molecular biology and the genome of *C. albicans*.

- Genetics
 - list of cloned genes
 - morphological mutants
- Physical map
 - Organization of the mapping data

- Chromosomes
- list of DNA probes
- Mapping information in the fosmid database
- Strains used for mapping and their morphology
- Sequence data
 - Summary of sequenced genes
 - *C. albicans* Genbank entries
 - Unpublished sequences
 - Primers for PCR of *Candida* gene
- *Candida* resources

Fungal Genome Resource

The Fungal Genome Resource has been established to promote research in the area of fungal genetics.

- Physical Maps of *Aspergillus Nidulans*.
- Fungal Genome Database
- mapping software used to create physical maps of fungi.
- Information on ODS, a Physical Mapping Software.
- Information on the management plan for the Fungal Genome Resource.

3.2.4.4 Summary

There has been life on earth for much of the planet's history. It is difficult to say when life first evolved or arrived here, but microbial life has been present for at least 3,500 million years, and the earth itself was only formed about 4,600 million years ago, after which its surface would need to have cooled to physiological temperatures. We have always known that the current methods of sampling and culturing of organisms from natural environments are deficient - these methods tend to select for the fastest-growing organisms in the culture conditions that are used. But the DNA sequencing methods discussed earlier have been a powerful new tool for detecting and ultimately isolating unknown (and even unsuspected) microorganisms. The mentioned microbial databanks available to make data available and enable the research personnel to evaluate and analyze for the human benefit.

3.2.4.5 Model Questions

1. What is a microbial databank ? What are its components ?
2. Mention a few microbial databanks in brief.
3. What are the various types of microbial databanks available?

3.2.4.6 References

<http://www.infobiogen.fr/deambulium>
<http://helios.bto.ed.ac.uk/bto/microbes/>
www.cbi.pk.edu.

Lesson 3.3.1.

THE NCBI DATA MODEL

Contents

- 3.3.1.1 Introduction**
- 3.3.1.2. Some Examples of the Model**
- 3.3.1.3. PUBs: PUBLICATIONS OR PERISH**
- 3.3.1.4. SEQ-IDs: WHAT'S IN A NAME?**
- 3.3.1.5. BIOSEQs: SEQUENCES**
- 3.3.1.6. BIOSEQ-SETs: COLLECTIONS OF SEQUENCES**
- 3.3.1.7. SEQ-ANNOT: ANNOTATING THE SEQUENCE**
- 3.3.1.8. SEQ-DESCR: DESCRIBING THE SEQUENCE**
- 3.3.1.9. USING THE MODEL**

Summary

Model Questions

Reference

Objective

- To understand the data model of NCBI Database
- To know what annotation is
- To understand annotated data and its features

3.3.1.1 Introduction

Why use a data model?

Most biologists are familiar with the use of animal models to study human diseases. Although a disease that occurs in humans may not be found in exactly the same form in animals, often an animal disease shared enough attributes with a human counterpart to allow data gathered on the animal disease to be used to make inference about the process in humans. Mathematical models describing the forces involved in musculoskeletal motions can be built by imagining that muscles are combinations of springs and hydraulic pistons and bones are lever arms, and, often times, such models allow meaningful predictions to be made and tested about the obviously much more complex biological system under consideration. The more closely and elegantly a model follows a real phenomenon, the more useful it is in predicting or understanding the natural phenomenon it is intended to mimic.

In this same vein, some 12 years ago, the National Center for Biotechnology Information (NCBI) introduced a new model for sequence-related information. This new and more powerful model made possible the rapid development of software and the integration of databases that underlie the popular Entrez retrieval system and on which the GenBank database is now built. The advantages of the model (e.g., the ability to move effortlessly from the published literature to DNA sequences to the proteins they encode, to chromosome maps of the genes, and to the three-dimensional structures of the proteins) have been apparent for years to biologists using Entrez, but very few biologists understand the foundation on which this model is built. As genome information becomes richer and more complex, more of the real, underlying data model

is appearing in common representations such as GenBank files. Without going into great detail, this chapter attempts to present a practical guide to the principles of the NCBI data model and its importance to biologists at the bench.

3.3.1.2. Some Examples of the Model

The GenBank flatfile is a "DNA-centered" report, meaning that a region of DNA coding for a protein is represented by a "CDS feature," or "coding region," on the DNA. A *qualifier* (/translation="MLLYY") describes a sequence of amino acids produced by translating the CDS. A limited set of additional *features* of the DNA, such as *mat_peptide*, are occasionally used in GenBank flatfiles to describe cleavage products of the (possibly unnamed) protein that is described by a / translation, but clearly this is not a satisfactory solution. Conversely, most protein sequence databases present a "protein-centered" view in which the connection to the encoding gene may be completely lost or may be only indirectly referenced by an accession number. Often times, these connections do not provide the exact codon-to-amino acid correspondences that are important in performing mutation analysis.

The NCBI data model deals directly with the two sequences involved: a DNA sequence and a protein sequence. The translation process is represented as a link between the two sequences rather than an annotation on one with respect to the other. Protein-related annotations, such as peptide cleavage products, are represented as features annotated directly on the protein sequence. In this way, it becomes very natural to analyze the protein sequences derived from translations of CDS features by BLAST or any other sequence search tool without losing the precise linkage back to the gene. A collection of a DNA sequence and its translation products is called a *Nuc-prot set*, and this is how such data is represented by NCBI. The GenBank flatfile format that many readers are already accustomed to is simply a particular style of report, one that is more "human-readable" and that ultimately flattens the connected collection of sequences back into the familiar one-sequence. DNA-centered view. The navigation provided by tools such as Entrez much more directly reflects the underlying structure of such data. The protein sequences derived from GenBank translations that are returned by BLAST searches are, in fact, the protein sequences -from the Nuc-prot sets described above.

The standard GenBank format can also hide the multiple-sequence nature of some DNA sequences. For example, three genomic exons of a particular gene are sequenced, and partial flanking, noncoding regions around the exons may also be available, but the full-length sequences of these intronic sequences may not yet be available. Because the exons are not in their complete genomic context, there would be three GenBank flatfiles in this case, one for each exon. There is no explicit representation of the complete set of sequences over that genomic region; these three exons come in genomic order and are separated by a certain length of unsequenced DNA. In GenBank format there would be a Segment line of the form SEGMENT 1 of 3 in the first record, SEGMENT 2 of 3 in the second, and SEGMENT 3 of 3 in the third, but this only tells the user that the lines are part of some undefined, ordered series (Fig. 2.1A). Out of the whole GenBank release, one locates the correct Segment records to place together by an algorithm involving the LOCUS name. All segments that go together use the same first combination of letters, ending with the numbers appropriate to the segment, e.g: HsDDT1, HsDDT2, and

HSDDT3. Obviously, this complicated arrangement can result in problems when LOCUS names include numbers that inadequately interfere with such series. In addition, there is no one sequence record that describes the whole assembled series, and there is no way to describe the distance between the individual pieces. There is no segmenting convention in the EMBL sequence database at all, so records derived from that source or distributed in that format lack even this imperfect information.

The NCBI data model defines a sequence type that directly represents such a segmented series, called a "segmented sequence." Rather than containing the letters A,G,C, and T, the segmented sequence contains instructions on how it can be built from other sequences. Considering again the example above, the segmented sequence would contain the instructions "take all of HSDDT1, then a gap of unknown length, then all of HSDDT2, then a gap of unknown length, then all of HSDDT3." The segmented sequence itself can have a name (e.g., HSDDT), an accession number, features, citations, and comments, like any other GenBank record. Data of this type are commonly stored in a so-called "Seg-set" containing the sequences HSDDT, HSDDT1, HSDDT2, HSDDT3 and all of their connections and features. When the GenBank release is made, as in the case of Nuc-prot sets, the Seg-sets are broken up into multiple records, and the segmented sequence itself is not visible. However, GenBank, EMBL, and DDBJ have recently agreed on a way to represent these constructed assemblies, and they will be placed in a new CON division, with CON standing for "contig" (Fig. 2.1 B). In the Entrez graphical view of segmented sequences, the segmented sequence is shown as a line connecting all of its component sequences (Fig. 2.1 C).

An NCBI segmented sequence does not require that there be gaps between the individual pieces. In fact the pieces can overlap, unlike the case of a segmented series in GenBank format. This makes the segmented sequence ideal for representing large sequences such as bacterial genomes, which may be many megabases in length. This is what currently is done within the Entrez Genomes division for bacterial genomes, as well as other complete chromosomes such as yeast. The NCBI Software Toolkit (Ostell, 1996.) contains functions that can gather the data that a segmented sequence refers to "on the fly," including constituent sequence and features, and this information can automatically be remapped from the coordinates of a small, individual record to that of a complete chromosome. This makes it possible to provide graphical views, GenBank flatfile views, or FASTA views or to perform analyses on whole chromosomes quite easily, even though data exist only in small, individual pieces. This ability to readily assemble a set of related sequences on demand for any region of a very large chromosome has already proven to be valuable for bacterial genomes. Assembly on demand will become more and more important as larger and larger regions are sequenced, perhaps by many different groups, and the notion that an investigator will be working on one huge sequence record becomes completely impractical.

(A) LOCUS HSDDT1 166 bp DNA PRI 01-FEB-2000
 DEFINITION Homo sapiens D-dopachrome tautomerase (DDT) gene, exon 1.
 ACCESSION AF012432
 VERSION AF012432.1 GI:2352911
 KEYWORDS
 SEGMENT 1 of 3

 LOCUS HSDDT2 216 bp DNA PRI 01-FEB-2000
 DEFINITION Homo sapiens D-dopachrome tautomerase (DDT) gene, exon 2.
 ACCESSION AF012433
 VERSION AF012433.1 GI:2352912
 KEYWORDS
 SEGMENT 2 of 3

 LOCUS HSDDT3 271 bp DNA PRI 01-FEB-2000
 DEFINITION Homo sapiens D-dopachrome tautomerase (DDT) gene, exon 3 and
 complete cds.
 ACCESSION AF012434
 VERSION AF012434.1 GI:2352913
 KEYWORDS
 SEGMENT 3 of 3

(B) LOCUS HSDDT 653 bp DNA CON 01-FEB-2000
 DEFINITION Homo sapiens D-dopachrome tautomerase (DDT) gene, complete cds.
 ACCESSION AH006997
 VERSION AH006997.2 GI:6849043
 KEYWORDS
 SOURCE human.
 ORGANISM Homo sapiens
 Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Mammalia;
 Eutheria; Primates; Catarrhini; Hominidae; Homo.
 REFERENCE 1 (bases 1 to 653)
 AUTHORS Esumi,N., Budarf,M., Ciccarelli,L., Sellinger,B., Kozak,C.A.
 and Wistow,G.
 TITLE Conserved gene structure and genomic linkage for D-dopachrome
 tautomerase (DDT) and MIP
 JOURNAL Mamm. Genome 9 (9), 753-757 (1998)
 MEDLINE 98384542
 PUBMED 9716662
 REFERENCE 2 (bases 1 to 653)
 AUTHORS Esumi,N. and Wistow,G.
 TITLE Direct Submission
 JOURNAL Submitted (07-JUL-1997) Molecular Structure and Function, NEI,
 Building 6, Rm. 331, NIH, Bethesda, MD 20892, USA
 COMMENT On Feb 1, 2000 this sequence version replaced gi:2352914.
 FEATURES
 Location/Qualifiers
 source 1..653
 /organism="Homo sapiens"
 /db_xref="taxon:9606"
 /chromosome="22"
 CONTIG join(AF012432.1:1..166,gap()),AF012433.1:1..216,gap()
 AF012434.1:1..271)
 //

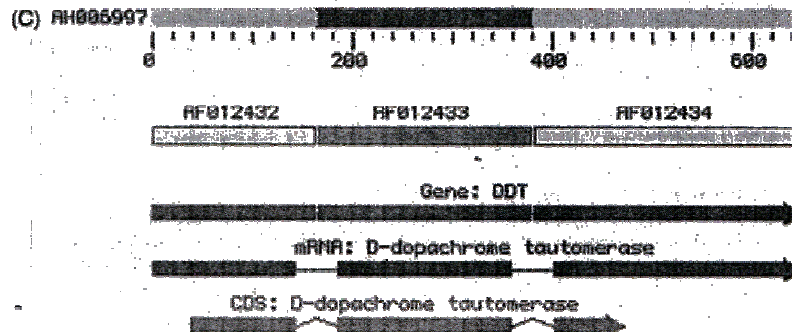


Figure 2.1. (A) Selected parts of GenBank-formatted records in a segmented sequence. GenBank format historically indicates merely that records are part of some ordered series; it offers no information on what the other components are or how they are connected. To see the complete view of these records, see

[http://www.ncbi.nlm.nih.gov/htbin-](http://www.ncbi.nlm.nih.gov/htbin-post/Entrez/query?uid=6849043&form=6&db=n&Dopt=g)

[post/Entrez/query?uid=6849043&form=6&db=n&Dopt=g](http://www.ncbi.nlm.nih.gov/htbin-post/Entrez/query?uid=6849043&form=6&db=n&Dopt=g). (B) Representation of segmented sequences in the new CON (contig) division. A new extension of GenBank format allows the details of the construction of segmented records to be presented. The CONTIG line can include individual accessions, gaps of known length, and gaps of unknown length. The individual components can still be displayed in the traditional form, although no features or sequences are present in this format. (C) Graphical representation of a segmented sequence. This view displays features mapped to the coordinates of the segmented sequence. The segments include all exonic and untranslated regions plus 20 base pairs of sequence at the ends of each intron. The segment gaps cover the remaining intronic sequence.

What Does ASN.1 Have to Do With It?

The NCBI data model is often referred to as, and confused with, the "NCBI ASN.I" or "ASN.I Data Model." *Abstract Syntax Notation 1* (ASN.1) is an International Standards Organization (ISO) standard for describing structured data that reliably encodes data in a way that permits computers and software systems of all types to reliably exchange both the structure and the content of the entries. Saying that a data model is written in ASN.I is like saying a computer program is written in C or FORTRAN. The statement identifies the *language*; it does not say what the program *does*. The familiar GenBank flatfile was really designed for humans to read, from a DNA-centered viewpoint. ASN.1 is designed for a *computer* to read and is amenable to describing complicated data relationships in a very specific way. NCBI describes and processes data using the ASN.1 format. Based on that single, common format, a number of human-readable formats and tools are produced, such as Entrez, GenBank, and the BLAST databases. Without the existence of a common format such as this, the neighboring and hard-link relationships that Entrez depends on would not be possible. This chapter deals with the structure and content of the NCBI data model and its implications for biomedical databases and tools. Detailed discussions about the choice of ASN.1 for this task and its overall form can be found elsewhere (Ostell, 1995).

What to Define?

We have alluded to how the NCBI data model defines sequences in a way that supports a richer and more explicit description of the experimental data than can be obtained with the GenBank format. The details of the model are important, and will be expanded on in the ensuing discussion. At this point, we need to pause and briefly describe the reasoning and general principles behind the model as a whole.

There are two main reasons for putting data on a computer: retrieval and discovery. Retrieval is basically being able to get back out what was put in. Amassing sequence information without providing a way to retrieve it makes the sequence information, in essence, useless. Although this is important, it is even *more* valuable to be able to get back from the system *more* knowledge than was put in to begin with—that is, to be able to

use the information to make biological discoveries. Scientists can make these kinds of discoveries by discerning connections between two pieces of information that were not known when the pieces were entered separately into the database or by performing computations on the data that offer new insight into the records. In the NCBI data model, the emphasis is on facilitating discovery; that means the data must be defined in a way that is amenable to both linkage and computation.

A second, general consideration for the model is stability. NCBI is a US Government agency, not a group supported year-to-year by competitive grants. Thus, the NCBI staff takes a very long-term view of its role in supporting bioinformatics efforts. NCBI provides large-scale information systems that will support scientific inquiry well into the future. As anyone who is involved in biomedical research knows, many major conceptual and technical revolutions can happen when dealing with such a long time span. Somehow, NCBI must address these changing views and needs with software and data that may have been created years (or decades) earlier. For that reason, basic observations have been chosen as the central data elements, with interpretations and nomenclature (elements more subject to change) being placed outside the basic, core representation of the data.

Taking all factors into account, NCBI uses four core data elements: bibliographic citations, DNA sequences, protein sequences, and three-dimensional structures. In addition, two projects (taxonomy and genome maps) are more interpretive but nonetheless are so important as organizing and linking resources that NCBI has built a considerable base in these areas as well.

3.3.1.3. PUBs: PUBLICATIONS OR PERISH

Publication is at the core of every scientific endeavor. It is the common process whereby scientific information is reviewed, evaluated, distributed, and entered into the permanent record of scientific progress. Publications serve as vital links between factual databases of different structures or content domains (e.g.: a record in a sequence database and a record in a genetic database may cite the same article). They serve as valuable entry points into factual databases ("I have read an article about this, now I want to see the primary data").

Publications also act as essential annotation of function and context to records in factual databases. One reason for this is that factual databases have a structure that is essential for efficient use of the database but may not have the representational capacity to set forward the full biological, experimental, or historical context of a particular record. In contrast, the published paper is limited only by language and contains much fuller and more detailed explanatory information than will ever be in - a record in a factual database. Perhaps more importantly, authors are evaluated by their scientific peers based on the content of their published papers, not by the content of the associated database records. Despite the best of intentions, scientists move on and database records become static, even though the knowledge about them has expanded, and there is very little incentive for busy scientists to learn a database system and keep records based on their own laboratory studies up to date.

Generally the form and content of citations have not been thought about carefully by those designing factual databases, and the quality, form, and content of citations can vary widely from one database to the next. Awareness of the importance of having a link

to the published literature and the realization that bibliographic citations are much less volatile than scientific knowledge led to a decision that a careful and complete job of defining citations was a worthwhile endeavor. Some components of the publication specification described below may be of particular interest to scientists or *users* of the NCBI databases, but a full discussion of all the issues leading to the decisions governing the specifications themselves would require another chapter in itself.

Authors:

Author names are represented in many formats by various databases: last name only, last name and initials, last name-comma-initials, last name and first name, all authors with initials and the last with a full first name, with or without honorifics(Ph.D.) or suffixes (Jr., III), to name only a few. Some bibliographic databases (such as MEDLINE) might represent only a fixed number of authors. Although this inconsistency is merely ugly to a human reader, it poses *severe* problems for database systems incorporating names from many sources and providing functions as simple as looking up citations by author last name, such as Entrez does. For this reason, the specification provides two alternative forms of author name representation: one a simple string and the other a structured form with fields for last name, first name, and so on. When data are submitted directly to NCBI or in cases when there is a consistent format of author names from a particular source (such as MEDLINE), the structured form is used. When the form cannot be deciphered, the author name remains as a string. This limits its use for retrieval but at least allows data to be viewed when the record is retrieved by other means.

Even the structured form of author names must support diversity, since some sources give only initials whereas others provide a first and middle name. This is mentioned to specifically emphasize two points. First, the NCBI data model is designed both to direct our view of the data into a more useful form and to accommodate the available existing data. (This pair of functions can be confusing to people reading the specification and seeing alternative forms of the same data defined.) Second, software developers must be aware of this range of representations and accommodate whatever form had to be used when a particular source was being converted. In general, NCBI tries to get as much of the data into a uniform, structured form as possible but carries the rest in a less optimal way rather than losing it altogether.

Author affiliations (i.e., authors' institutional addresses) are even more complicated. As with author names, there is the problem of supporting both structured forms and unparsed strings. However, even sources with reasonably consistent author name conventions often produce affiliation information that cannot be parsed from text into a structured format. In addition, there may be an affiliation associated with the whole author list, or there may be different affiliations associated with each author. The NCBI data model allows for both scenarios. At the time of this writing only the first form is supported in either MEDLINE or GenBank, both types may appear in published articles.

Articles

The most commonly cited bibliographic entity in biological science is an article in a journal; therefore, the citation formats of most biological databases are defined with that type in mind. However, "articles" can also appear in books, manuscripts, theses, and now

in electronic journals as well. The data model defines the fields necessary to cite a book, a journal, or a manuscript. An article citation occupies one field; other fields display additional information necessary to uniquely identify the article in the book, journal, or manuscript—the author(s) of the article (as opposed to the author or editor of the book), the title of the article, page numbers, and so on.

There is an important distinction between the fields necessary to uniquely identify a published article from a citation and those necessary to describe the same article meaningfully to a database user. The NCBI Citation Matching Service takes fields from a citation and attempts to locate the article to which they refer. In this process, a successful match would involve only correctly matching the journal title, the year, the first page of the article, and the last name of all author of the article. Other information (e.g., article title, volume, issue, full pages, author list) is useful to look at but very often is either not available or outright incorrect. Once again, the data model must allow the minimum information set to come in as a citation, be of desired against MEDLINE, and then be replaced by a citation having the full set of desired fields obtained from MEDLINE to produce accurate, useful data for consumption by the scientific public.

Patents

With the advent of patented sequences it became necessary to cite a patent as a bibliographic entity instead of an article. The data model supports a very complete patent citation, a format developed in cooperation with the US Patent Office. In practice, however, patented sequences tend to have limited value to the scientific public. Because a patent is a *legal* document, not a scientific one, its purpose is to present and support the claims of the patent, *not* to fully describe the biology of the sequence itself. It is often prepared in a lawyer's office, not by the scientist who did the research. The sequences presented in the patent may function only to illustrate some discreet aspect of the patent, rather than being the focus of the document. Organism information, location of biological features, and so on may not appear at all if they are not germane to the patent. Thus far, the vast majority of sequences appearing in patents also appear in a more useful form (to scientists) in the public databases. .

In NCBI's view, the main purpose of listing patented sequences in GenBank is to be able to retrieve sequences by similarity searches that may serve to locate patents related to a given sequence. To make a legal determination in the case, however, one would still have to examine the full text of the patent. To evaluate the biology of the sequence, one generally must locate information other than that contained in the patent. Thus, the critical linkage is between the sequence and its patent number. Additional fields in the patent citation itself may be some interest, such as the title of the patent and the names of the inventors.

Citing Electronic Data Submission

A relatively new class of citations comprises the act of data submission to a database, such as GenBank. This is an act of publication, similar but not identical to the publication of an article in a journal. In some cases, data submission precedes article publication by a considerable period of time, or a publication regarding a particular sequence may never appear in press. Because of this, there is a separate citation designed

for deposited sequence data. The submission citation, because it is indeed an act of publication, may have an author list, showing the names of scientists who worked on the record. This mayor may not be the same as the author list on a subsequently published paper also cited in the same record. In most cases, the scientist who submitted the data to the database is also an author on the submission citation. (In the case of large sequencing centers_ this may not always be the case.) Finally, NCBI has begun the practice of citing the update of a record with a submission citation as well. A comment can be included with the update, briefly describing the changes made in the record. All the submission citations can be retained in the record, providing a history of the record over time.

MEDLINE and PubMed Identifiers

Once an article citation has been matched to MEDLINE, the simplest and most reliable key to point to the article is the MEDLINE unique identifier (MUID). This is simply an integer number. NCBI provides many services that use MUID to retrieve the citation and abstract from MEDLINE, to link together data citing the same article, or to provide Web hyperlinks.

Recently, in concert with MEDLINE and a large number of publishers. NCBI has introduced *PubMed*. PubMed contains *all* of MEDLINE as well as citations provided directly by the publishers. As such, PubMed contains more recent articles than MEDLINE, as well as articles that may never appear in MEDLINE because of their subject matter. This development led NCBI to introduce a new article identifier, called a PubMed identifier (PMID). Articles appearing in MEDLINE will have *both* a PMID and an MUID. Articles appearing only in PubMed will have only a PMID. PMID serves the same purpose as MUID in providing a simple, reliable link to the citation, a means of linking records together, and a means of setting up hyperlinks.

Publisher have also started to send information on ahead-or-print articles to PubMed, so this information may now appear before the printed journal. A new project *PubMed Central*, is meant to allow electronic publication to occur in lieu of or ahead of publication in a traditional, printed journal. PubMed Central records contain the full text of the article, not just the abstract, and include all figures and references.

The NCBI data model stores most citations as a collection called a Pub-equiv, a set of equivalent citations that includes a reliable identifier (PMID or MUID) and the citation itself. The presence of the citation form allows a useful display without an extra retrieval from the database. Where as the identifier provides a reliable key for linking or indexing the same citation in the record.

3.3.1.4. SEQ-IDs: WHAT'S IN A NAME?

The NCBI data model defines a whole class of objects called Sequence Identifiers (Seq-id). There has to be a whole class of such objects because NCBI integrates sequence data from many sources that name sequence records in different ways and where, of course, the individual names have different meanings. In one simple case, PIR, SWISS.PROT, and the nucleotide sequence databases all use a string called an "accession number," all having a similar format. Just saying "A10234" is not enough to uniquely identify a sequence record from the collection of all these databases. One must distinguish "A 10234" in SWISS-PROT from " A 10234" in PIR. (The DDBJ/EMBL/GenBank nucleotide databases share a common set of accession numbers; therefore, "A12345" in EMBL is the

same as "A12345" in GenBank or DDBJ.) To further complicate matters, although the sequence database define their records as containing a single sequence, PDB records contain a single *structure*, which may contain more than one sequence. Because of this, a PDB Seq-id contains a molecule name and a chain ID to identify a single unique sequence. The subsections that follow describe the form and use of a few commonly used types of Seq-ids.

Locus Name

The *locus* appears on the LOCUS line in GenBank and DDBJ records and in the ID line in EMBL records. These originally were the only identifier of a discrete GenBank record. Like a genetic locus name, it was intended to act both as a unique identifier for the record and as a mnemonic for the function and source organism of the sequence. Because the LOCUS line is in a fixed format, the locus name is restricted to ten or fewer numbers and uppercase letters. For many years in GenBank, the first three letters of the name were an organism code and the remaining letters a code for the gene (e.g., HUMHBB was used for "human β -globin region"). However, as with genetic locus names, locus names were changed when the function of a region was discovered to be different from what was originally thought. This instability in locus names is obviously a problem for an identifier for retrieval. In addition, as the number of sequences and organisms represented in GenBank increased geometrically over the years, it became impossible to invent and update such mnemonic names in an efficient and timely manner. At this point, the locus name is dying out as a useful name in GenBank, although it continues to appear prominently on the first line of the t1atfile to avoid breaking the established format.

Accession Number

Because of the difficulties in using the locus/ID name as the unique identifier for a nucleotide sequence record, the International Nucleotide Sequence Database Collaborators (DDBJ/EMBL/GenBank) introduced the accession number. It intentionally carries no biological meaning, to ensure that it will remain (relatively) stable. It originally consisted of one uppercase letter followed by five digits. New accessions consist of two uppercase letters followed by six digits. The first letters were allocated to the individual collaborating databases so that accession numbers would be unique across the Collaboration (e.g., an entry beginning with a "U" was from GenBank)

The accession number was an improvement over the locus/ID name, but, with use, problems and deficiencies became apparent. For example, although the accession is stable over time, many users noticed that the sequence retrieved by a particular accession was not always the same. This is because the accession identifies the *whole database record*. If the sequence in a record was updated (say by the insertion of 1000 bp at the beginning), the accession number did not change, as it was an updated version of the same record. If one had analyzed the original sequence and recorded that at position 100 of accession UOOOO I there was a putative protein-binding site, after the update a completely different sequence would be found at position 100!

The accession number appears on the ACCESSION line of the GenBank record. The first accession on the line, called the "primary" accession, is the key for retrieving this record. Most records have only this type of accession number. However, other accession

may follow the primary accession on the `ACCESSION` line. These "secondary" accessions are intended to give some notion of the history of the record. For example, if `U00001` and `U00002` were merged into a single updated record, then `U00001` would be the primary accession on the new record and `U00002` would appear as a secondary accession. In standard practice, the `U00002` record would be removed from GenBank, since the older record had become obsolete, and the secondary accessions would allow users to retrieve whatever records superseded the old one. It should also be noted that, historically, secondary accession numbers do not always mean the same thing; therefore, users should exercise care in their interpretations (policies at individual databases differed, and even shifted over time in a given database.) The use of secondary accession numbers also caused problems in that there was still not enough information to determine exactly what happened and why. Nonetheless, the accession number remains the most controlled and reliable way to point to a record in DDBJ/EMBL/GenBank.

gi Number

In 1992, NCBI began assigning GenInfo Identifiers (`gi`) to all sequences processed into Entrez, including nucleotide sequences from DDBJ/EMBL/GenBank, the protein sequences from the translated CDS features, protein sequences from SWISS-PROT, PIR, PRF, PDB, patents, and others. The `gi` is assigned in addition to the accession number provided by the source database. Although the form and meaning of the accession `Seq-id` varied depending on the source, the meaning and form of the `gi` is the same for all sequences regardless of the source.

The `gi` is simply an integer number, sometimes referred to as a *GI number*. It is, in fact, an identifier for a particular sequence only. Suppose a sequence enters GenBank and is given an accession number `U00001`. When the sequence is processed internally at NCBI, it enters a database called ID. ID determines that it has not seen `U00001` before and assigns it a `gi` number—for example, 54. Later, the submitter might update the record by changing the citation, so `U00001` enters ID again. ID, recognizing the record, retrieves the first `U00001` and compares its sequence with the new one. If the two are completely identical, ID reassigns `gi` 54 to the record. If the sequence differs in any way, even by a single base pair, it is given a new `gi` number, say 88. However, the new sequence retains accession number `U00001` because of the semantics of the source database. At this time, ID marks the old record (`gi` 54) with the date it was replaced and adds a "history" indicating that it was replaced by `gi` 88. ID also adds a history to `gi` 88 indicating that it replaced `gi` 54.

The `gi` number serves three major purposes:

- It provides a single identifier across sequences from many sources.
- It provides an identifier that specifies an exact sequence. Anyone who analyzes `gi` 54 and stores the analysis can be sure that it will be valid as long as `U00001` has `gi` 54 attached to it.
- It is stable and retrievable. NCBI keeps the last version of every `gi` number. Because the history is included in the record, anyone who discovers that `gi` 54 is no longer part of the GenBank release can still retrieve it from ID through NCBI and examine the history to see that it was replaced by `gi` 88. Upon aligning `gi` 54 to `gi` 88 to determine their relationship, a researcher may decide to remap the former analysis to `gi` 88 or perhaps to reanalyze the data. This can be done at any time,

not just at GenBank release time, because gi 54 will always be available from ID.

For these reasons, all internal processing of sequences at NCB I, from computing Entrez sequence neighbors to determining when new sequence should be processed or producing the BLAST databases, is based on gi numbers.

Accession.Version Combined Identifier:

Recently, the members of the International Nucleotide Sequence Database Collaboration (GenBank, EMBL, and DDBJ) introduced a "better" sequence identifier, one that combines an accession (which identifies a particular sequence record) with a version number (which tracks changes to the sequence itself). It is expected that this kind of Seq-id will become the preferred method of citing sequences.

Users will still be able to retrieve a record based on the accession number alone. With out having to specify a particular version. In that case, the latest version of the record will be obtained by default, which is the current behavior for queries using Entrez and other retrieval programs.

Scientists who are analyzing sequences in the database (e.g., aligning all alcohol dehydrogenase sequences from a particular taxonomic group) and wish to have their conclusions remain valid over time will want to reference sequences by accession and the given version number. Subsequent modification of one of the sequences by its owner (e.g., 5' extension during a study of the gene's regulation) will result in the version number being incremented appropriately. The analysis that cited accession and version remains valid because a query using both the accession and version will return the desired record.

Combining accession and version makes it clear to the casual user that a sequence has changed since an analysis was done. Also, determining how many times a sequence has changed becomes trivial with a version number. The accession, version number appears on the VERSION line of the GenBank flatfile. For sequence retrieval, the accession, version is simply mapped to the appropriate gi number, which remains the underlying tracking identifier at NCBI.

Accession Numbers on Protein Sequences

The International Sequence Database Collaborators also started assigning accession, version numbers to *protein* sequences within the records. Previously, it was -difficult to reliably cite the translated product of a given coding region feature, except by its gi number. This limited the usefulness of translated products found in BLAST results, for example. These sequences will now have the same status as protein sequences submitted directly to the protein databases, and they have the benefit of direct linkage to the nucleotide sequence in which they are encoded, showing up as a CDS feature's /protein_id qualifier in the flatfile view. Protein accessions in these records consist of three uppercase letters followed by five digits and an integer indicating the version.

Reference Seq-id

The NCBI RefSeq project provides a curated, nonredundant set of reference sequence standards for naturally occurring biological molecules, ranging from chromosomes to transcripts to proteins. RefSeq identifiers are in accession. version form but are prefixed with NC- (chromosomes), NM- (mRNAs), NP- (proteins), or NT- (constructed genomic contigs). The NG- prefix will be used for genomic regions or gene clusters (e.g.,

immunoglobulin region) in the future. RefSeq records are a stable reference point for functional annotation, point mutation analysis, gene expression studies, and polymorphism discovery.

General Seq-id

The General Seq-id is meant to be used by genome centers and other groups as a way of identifying their sequences. Some of these sequences may never appear in public databases, and others may be preliminary data that eventually will be submitted. For example, records of human chromosomes in the Entrez Genomes division contain multiple physical and genetic maps, in addition to sequence components. The physical maps are generated by various groups, and they use General Seq-ids to identify the proper group.

Local Seq-id

The Local sequence identifier is most prominently used in the data submission tool Sequin. Each sequence will eventually get an accession. Version identifier and a gi number, but only when the completed submission has been processed by one of the public databases. During the submission process, Sequin assigns a local identifier to each sequence. Because many of the software tools made by NCBI require a sequence identifier, having a local Seq-id allows the use of these tools without having to first submit data to a public database.

3.3.1.5. BIOSEQs: SEQUENCES

The Bioseq, or biological sequence, is a central element in the NCBI data model. It comprises a single, continuous molecule of either nucleic acid or protein, thereby defining a linear, integer coordinate system for the sequence. A Bioseq must have at least one sequence identifier (Seq-id). It has information on the physical type of molecule (DNA, RNA, or protein). It may also have annotations, such as biological features referring to specific locations on specific Bioseqs, as well as descriptors.

Descriptors provide additional information, such as the organism from which the molecule was obtained. Information in the descriptors describes the entire Bioseq.

However, the Bioseq isn't necessarily a fully sequenced molecule. It may be a segmented sequence in which, for example, the exons have been sequenced but not all of the intronic sequences have been determined. It could also be a genetic or physical map, where only a few landmarks have been positioned.

Sequences are the same

All Bioseqs have an integer coordinate system, with an integer length value, even if the actual sequence has not been completely determined. Thus, for physical maps, or for exons in highly spliced genes, the spacing between markers or exons may be known only from a band on a gel. Although the coordinates of a fully sequenced chromosome are

known exactly, those in a genetic or physical map are a best guess, with the possibility of significant error from the "real" coordinates.

Nevertheless, any Bioseq can be annotated with the same kinds of information. For example, a gene feature can be placed on a region of sequenced DNA or at a discrete location on a physical map. The map and the sequence can then be aligned on the basis of their common gene features. This greatly simplifies the task of writing software that can display these seemingly disparate kinds of data.

Sequences are Different

Despite the benefits derived from having a common coordinate system, the different Bioseq classes do differ in the way they are represented. The most common classes (Fig. 2.2)' are described briefly below.

Virtual Bioseq: In the virtual Bioseq, the molecule type is known, and its length and topology (e.g., linear, circular) may also be known, but the actual sequence is not known. A virtual Bioseq can represent an intron in a genomic molecule in which only the exon sequences have been determined. The length of the putative sequence may be known only by the size of a band on an agarose gel.

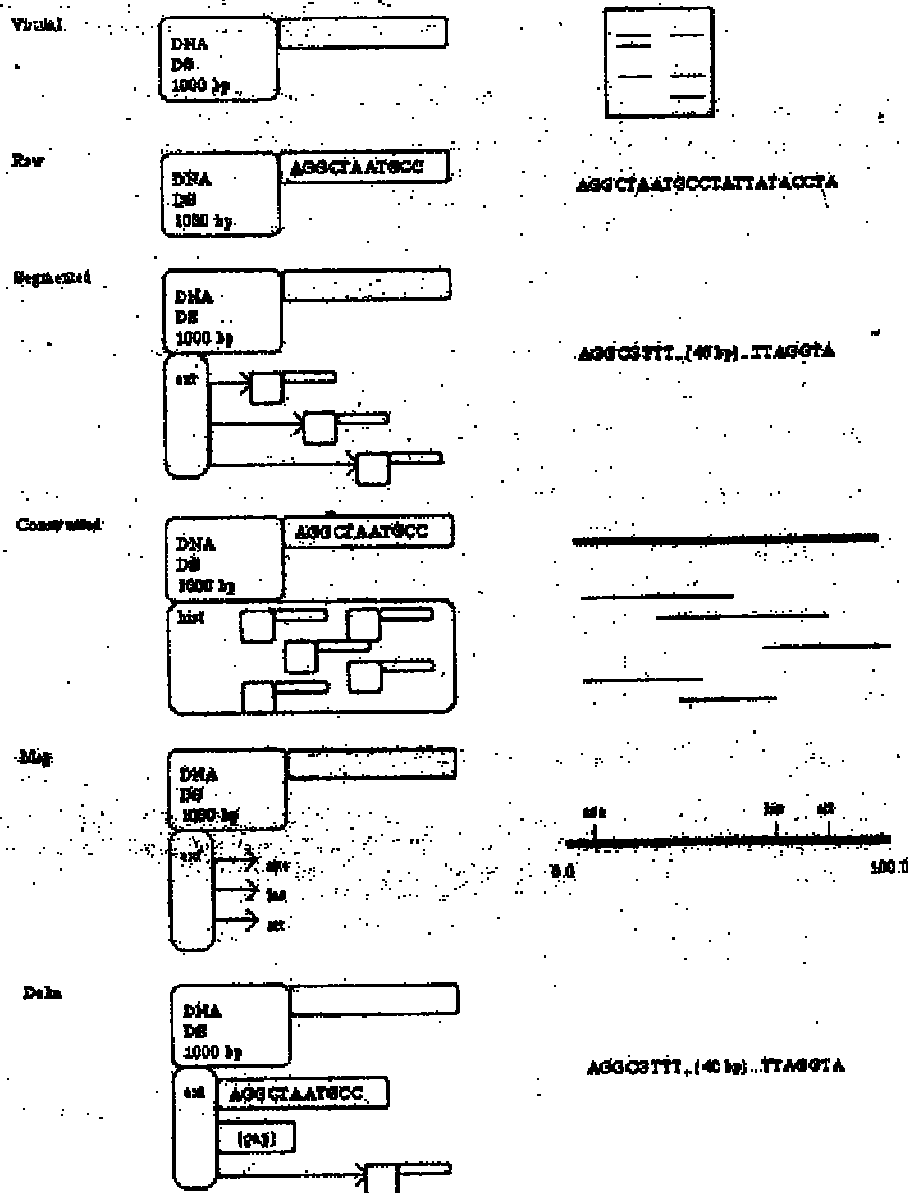


Figure 2.2. Classes of Bioseqs. All Bioseqs represent a single, continuous molecule of nucleic acid or protein, although the complete sequence may not be known. In a virtual Bioseq, the type of molecule is known, but the sequence is not known, and the precise length may not be known (e.g., from the size of a band on an electrophoresis gel). A raw Bioseq contains a single contiguous string of bases or residues. A segmented Bioseq points to its components, which are other raw or virtual Bioseqs (e.g., sequenced exons and undetermined introns). A constructed sequence takes its original components and

subsumes them, resulting in a Bioseq that contains the string of bases or residues and a "history" of how it was built. A map Bioseq places genes or physical markers, rather than sequence, on its coordinates. A delta Bioseq can represent a segmented sequence but without the requirement of assigning identifiers to each component (including gaps of known length) although separate raw sequences can still be referenced as components. The delta sequence is used for unfinished high-throughput genome sequences (HTGS) from genome centers and for genomic contigs.

Raw Bioseq. This is what most people would think of as a sequence, a single contiguous string of bases or residues, in which the actual sequence is known. The length is obviously known in this case, matching the number of bases or residues in the sequence.

Segmented Bioseq. A segmented Bioseq does not contain raw sequences but instead contains the identifiers of other Bioseqs from which it is made. This type of Bioseq can be used to represent a genomic sequence in which only the exons are known. The "parts" in the segmented Bioseq would be the individual, raw Bioseqs representing the exons and the virtual Bioseqs representing the introns.

Delta Bioseq. Delta Bioseqs are used to represent the unfinished high-throughput genome sequences (HTGS) derived at the various genome sequencing centers. Using delta Bioseqs instead of segmented Bioseqs means that only one Seq-id is needed for the entire sequence, even though subregions of the Bioseq are not known at the sequence level. Implicitly, then, even at the early stages of their presence in the databases, delta Bioseqs maintain the same accession number.

Map Bioseq. Used to represent genetic and physical maps, a map Bioseq is similar to a virtual Bioseq in that it has a molecule type, perhaps a topology, and a length that may be a very rough estimate of the molecule's actual length. This information merely supplies the coordinate system, a property of every Bioseq. Given this coordinate system for a genetic map, we estimate the positions of genes on it based on genetic evidence. The table of the resulting gene features is the essential data of the map Bioseq, just as bases or residues constitute the raw Bioseq's data.

3.3.1.6. BIOSEQ-SETS: COLLECTIONS OF SEQUENCES

A biological sequence is often most appropriately stored in the context of other, related sequences. For example, a nucleotide sequence and the sequences of the protein products it encodes naturally belong in a set. The NCBI data model provides the Bioseq-set for this purpose.

A Bioseq-set can have a list of *descriptors*. When packaged on a Bioseq, a descriptor applies to all of that Bioseq. When packaged on a Bioseq-set, the descriptor applies to every Bioseq in the set. This arrangement is convenient for attaching publications and biological source information, which are expected on all sequences but frequently are identical within sets of sequences. For example, both the DNA and protein sequences are obviously from the same organism, so this descriptor information can be applied to the set. The same logic may apply to a publication.

The most common Bioseq-sets are described in the sections that follow.

Nucleotide/Protein Sets

The Nuc-prot set, containing a nucleotide and one or more protein products, is the type of set most frequently produced by a Sequin data submission. The component Bioseqs are connected by coding sequence region (CDS) features that describe how translation from nucleotide to protein sequence is to proceed. In a traditional nucleotide or protein sequence database, these records might have cross-references to each other to indicate this relationship. The Nuc-prot set makes this explicit by packaging them together. It also allows descriptive information that applies to all sequences (e.g., the organism or publication citation) to be entered once (see *Seq-descr: Describing the Sequence*, below).

Population and Phylogenetic Studies

A major class of sequence submissions represent the results of population or phylogenetic studies. Such research involves sequencing the same gene from a number of individuals in the same species (population study) or in different species (phylogenetic study). An alignment of the individual sequences may also be submitted (see *Seq-llign: Alignments*, below). If the gene encodes a protein, the components of the Population or Phylogenetic Bioseq-set may themselves be Nuc-prot sets.

Other Bioseq-sets

A Seg set contains a segmented Bioseq and a Parts Bioseq-set, which in turn contains the raw Bioseqs that are referenced by the segmented Bioseq. This may constitute the nucleotide component of a Nuc-prot set.

An Equiv Bioseq-set is used in the Entrez Genomes division to hold multiple equivalent Bioseqs. For example, human chromosomes have one or more genetic maps, physical maps derived by different methods and a segmented Bioseq on which "islands" of sequenced regions are placed. An alignment between the various Bioseqs is made based on references to any available common markers.

3.3.1.7.SEQ-ANNOT: ANNOTATING THE SEQUENCE

A Seq-annot is a self-contained package of sequence annotations or information that refers to specific locations on specific Bioseqs. It may contain a feature table, a set of sequence alignments, or a set of graphs of attributes along the sequence.

Multiple seq-annots can be placed on a Bioseq or on a Bioseq-set. Each, Seqannot can have specific attribution, For example. PowerBLAST (Zhang and Madden, 1997) produces a seq-annot containing sequence alignments, and each seq-annot is named based on the BLAST program used (e.g., BLASTN, BLASTX, etc.). The individual blocks of alignments are visible in the Entrez and Sequin viewers.

Because the components of a Seq-annot have specific references to locations on Bioseqs, the seq-annot can stand alone or be exchanged with other scientists, and it need not reside in a sequence record. The scope of descriptors, on the other hand, does depend on where they are packaged. Thus, information *about* Bioseqs can be created, exchanged, and compared independently of the Bioseq itself. This is an important attribute of the seq-annot and of the NCBI data model.

Seq-feat: Features

A sequence feature (Seq-feat) is a block of structured data explicitly attached to a region of a Bioseq through one or two sequence locations (Seq-locs). The Seq-feat itself can carry information common to all features. For example, there are flags to indicate whether a feature is partial (i.e., goes beyond the end of the sequence of the Bioseq), whether there is a biological exception (e.g., RNA editing that explains why a codon on the genomic sequence does not translate to the expected amino acid), and whether the feature was experimentally determined (e.g., an mRNA was isolated from a proposed coding region).

A feature must always have a location. This is the Seq-loc that states where on the sequence the feature resides. A coding region's location usually starts at the ATG and ends at the terminator codon. The location can have more than one interval if it is on a genomic sequence and mRNA splicing occurs. In cases of alternative splicing, separate coding region features are created, with one multi-interval Seq-loc for each isolated molecular species.

Optionally, a feature may have a product. For a coding region, the product Seqloc points to the resulting protein sequence. This is the link that allows the data model to separately maintain the nucleotide and protein sequences, with annotation on each sequence appropriate to that molecule. An mRNA feature on a genomic sequence could have as its product an mRNA Bioseq whose sequence reflects the results of posttranscriptional RNA editing. Features also have information unique to the kind of feature. For example, the CDS feature has fields for the genetic code and reading frame, whereas the tRNA feature has information on the amino acid transferred.

This design completely modularizes the components required by each feature type. If a particular feature type calls for a new field, no other field is affected. A new feature type, even a very complex one, can be added without changing the existing features. This means that software used to display feature locations on a sequence need consider only the location field common to all features.

Although the DDBJ/EMBL/GenBank feature table allows numerous kinds of features to be included (see Chapter 3), the NCBI data model treats some features as "more equal" than others. Specifically, certain features directly model the central dogma of molecular biology and are most likely to be used in making connections between records and in discovering new information by computation. These features are discussed next.

Genes. A gene is a feature in its own right. In the past, it was merely a qualifier on other features. The Gene feature indicates the location of a gene, a heritable region of nucleic acid sequence that confers a measurable phenotype. That phenotype may be achieved by many components of the gene being studied, including, but not limited to, coding regions, promoters, enhancers, and terminators. The Gene feature is meant to approximately cover the region of nucleic acid considered by workers in the field to be the gene. This admittedly fuzzy concept has an appealing simplicity, and it fits in well with higher-level views of genes such as genetic maps. It has practical utility in the era of large genomic sequencing when a biologist may wish to see just the "xyz gene" and not a whole chromosome. The Gene feature may also contain cross-references to genetic databases, where more detailed information on the, gene may be found.

RNAs. An RNA feature can describe both coding intermediates (e.g., mRNAs) and structural RNAs (e.g., tRNAs, rRNAs). The locations of an mRNA and the corresponding coding region (CDS) completely determine the locations of 5' and 3' untranslated regions (UTRs), exons, and introns.

Coding Regions. A Coding Region (CDS) feature in the NCBI data model can be thought of as "instructions to translate" a nucleic acid into its protein product, via a genetic code (Fig. 2.3). A coding region serves as a link between the nucleotide and protein. It is important to note that several situations can provide exceptions to the classical colinearity of gene and protein. Translational stuttering (ribosomal slippage) for example, merely results in the presence of overlapping intervals in the feature's location Seq-loc.

The genetic code is assumed to be universal unless explicitly given in the Coding Region feature. When the genetic code is not followed at specific positions in the sequence—for example, when alternative initiation codons are used in the first position, when suppressor tRNAs bypass a terminator, or when selenocysteine is added—the Coding Region feature allows these anomalies to be indicated.

Proteins. A Protein feature names (or at least describes) a protein or proteolytic product of a protein. A single protein Bioseq may have many Protein features on it. It may have one over its full length describing a pro-peptide, the primary product of translation. (The name in this feature is used for the /product qualifier in the CDS feature that produces the protein.) It may have a shorter protein feature describing the mature peptide or, in the case of viral polyproteins, several mature peptide features. Signal peptides that guide a protein through a membrane may also be indicated.

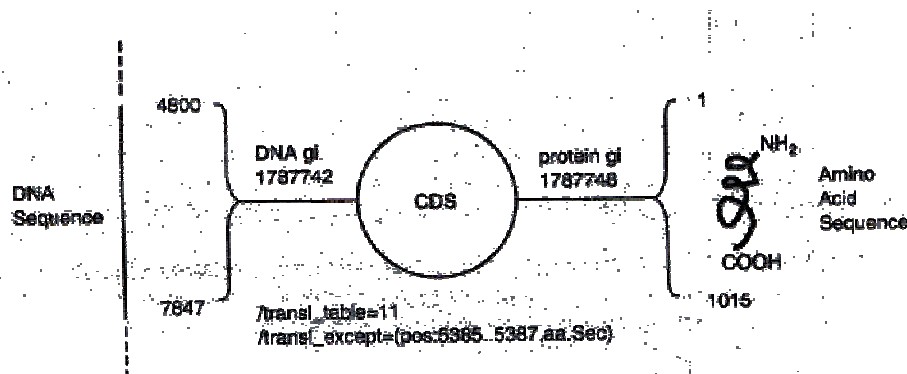


Figure 2.3. *The Coding Region (CDS) feature links specific regions on a nucleotide sequence with its encoded protein product. All features in the NCBI data model have a "location" field, which is usually one or more intervals on a sequence. (Multiple intervals on a CDS feature would correspond to individual exons.) Features may optionally have a "product" field, which for a CDS feature is the entirety of the resulting protein sequence. The CDS feature also contains a field for the genetic code. This appears in the GenBank flat file as a /trans 1- table qualifier. 'n this example, the Bacterial genetic code (code 11) is indicated. A CDS may also have translation exceptions indicating that a particular residue is not what is expected, given the codon and the genetic code. In this example, residue 196 in the protein is selenocysteine, indicated by the /transl _except qualifier. NCBI software includes functions for converting between codon locations and residue locations, using the CDS as its guide. This capability is used to support the historical conventions of GenBank format, allowing a signal peptide, annotated on the protein sequence, to appear in the GenBank flat file with a location on the nucleotide sequence.*

Others. Several other features are less commonly used. A Region feature provides a simple way to name a region of a chromosome (e.g., "major histocompatibility complex") or a domain on a polypeptide. A Bond feature annotates a bond between two residues in a protein (e.g., disulfide). A Site feature annotates a known site (e.g., active, binding, glycosylation, methylation, phosphorylation).

Finally, numerous features exist in the table of legal features, covering many aspects of biology. However, they are less likely than the above-mentioned features to be used for making connections between records or for making discoveries based on computation.

Seq-align: Alignments

Sequence alignments simply describe the relationships between biological sequences by designating portions of sequences that correspond to each other. This correspondence can reflect evolutionary conservation, structural similarity, functional similarity, or a random event. An alignment can be generated algorithmically by software (e.g., BLAST produces a Seq-annot containing one or more Seq-aligns) or directly by a scientist (e.g., one who is submitting an aligned population study using a favorite alignment tool and a submission program like Sequin). The Seq-align is designed to capture the final result of the process, not the process itself. Aligned regions can be given scores according to the probability that the alignment is a chance occurrence.

Regardless of how or why an alignment is generated or what its biological significance may be the data model records, in a condensed format, which regions of which sequences are said to correspond. The fundamental unit of an alignment is a segment, which is defined as an unbroken region of the alignment. In these segments, each sequence is present either without gaps or is not present at all (completely gapped). The alignment below has four segments, delineated by vertical lines:

```

MRLTLLC-----EGEEGSELPLCASCQRIELKYKPECYPDVKNSLHV
MRLTLLCCTWREERMGEEGSELPVCASCQORLELKYKPECFPDVKNSIHA
MRLTCLCRTWREERMGEEGSEIPVCASCQORIELKYKPE-----

```

Note that mismatches do not break a segment; only a gap opening or closing event will force the creation of a new segment.

By structuring the alignment in this fashion, it can be saved in condensed form. The data representation records the start position in sequence coordinates for each sequence in a segment and the length of the segment. If a sequence is gapped in a segment, its start position is -1. Note that this representation is independent of the actual sequence; that is, nucleotide and protein alignments are represented the same way, and only the score of an alignment gives a clue as to how many matches and mismatches are present in the data.

The Sequence Is Not the Alignment

Note that the gaps in the alignment are not actually represented in the Bioseqs as dashes. A fundamental property of the genetic code is that it is "commaless" (Crick et-al.. 1961). That is, there is no "punctuation" to distinguish one codon from the next or to keep translation in the right frame. The gene is a contiguous string of nucleotides. We remind the reader that sequences themselves are also "gapless." Gaps are shown only in the alignment report, generated from the alignment data; they are used only for comparison.

Classes of Alignments

Alignments can exist by themselves or in sets and can therefore represent quite complicated relationships between sequences. A single alignment can only represent a continuous and linear correspondence, but a set of alignments can denote a continuous, discontinuous, linear, or nonlinear relationship among sequences. Alignments can also be local, meaning that only portions of the sequences are included in the alignment, or they can be global, so that the alignment completely spans all the sequences involved. I

A continuous alignment does not have regions that are unaligned; that is, for each sequence in the alignment, each residue between the lowest-numbered and highest-numbered residues of the alignment is also contained in the alignment. More simply put, there are no pieces missing. Because such alignments are necessarily linear, they can be displayed with one sequence on each line, with gaps representing deletions or insertions. To show the differences from a "master" sequence, one of the sequences can be displayed with no gaps and no insertions; the remaining sequences can have gaps or inserted segments (often displayed above or below the rest of the sequence), as needed. If pairwise, the alignment can be displayed in a square matrix as a squiggly line traversing the two sequences.

A discontinuous alignment contains regions that are unaligned. For example, the alignment below is a set of two local alignments between two protein sequences. The regions in between are simply not aligned at all:

```
12 MA-TLICCTWREGRMG 26 45 KPECFPDVKNSIHV 58  
15 MRLTLLCC'I'WREERMG 30 35 KPECFPDAKNSLHV 48
```

This alignment could be between two proteins that have two matching (but not identical) structural domains linked by a divergent segment. There is simply no alignment for the regions that are not shown above. A discontinuous alignment can be linear, like the one in the current example, so that the sequences can still be shown one to a line without breaking the residue order. More complicated discontinuous alignments may

have overlapping segments, alignments on opposite strands (for nucleotides), or repeated segments, so that they cannot be displayed in linear order. These nonlinear alignments are the norm and can be displayed in square matrices (if pairwise), in lists of aligned regions, or by complex shading schemes.

Data Representations of Alignments

A continuous alignment can be represented as a single list of coordinates, as described above. Depending on whether the alignment spans all of the sequences, it can be designated global or local.

Discontinuous alignments must be represented as sets of alignments, each of which is a single list of coordinates. The regions between discontinuous alignments are not represented at all in the data, and, to display these regions, the missing pieces must be calculated. If the alignment as a whole is linear, the missing pieces can be fairly simply calculated from the boundaries of the aligned regions. A discontinuous alignment is usually local, although if it consists of several overlapping pieces it may in fact represent a global correspondence between the sequences.

Seq-graph: Graphs

Graphs are the third kind of annotation that can go into Seq-annots. A Seq-graph defines some continuous set of values over a defined interval on a Bioseq. It can be used to show properties like G+C content, surface potential, hydrophobicity, or base accuracy over the length of the sequence.

3.3.1.8. SEQ-DESCR: DESCRIBING THE SEQUENCE

A seq-descr is meant to describe a Bioseq (or Bioseq-set) and place it in its biological and/or bibliographic context. Seq-descrs apply to the whole Bioseq or to the whole of each Bioseq in the Bioseq-set to which the Seq-descr is attached.

Descriptors were introduced in the NCBI data model to reduce redundant information in records. For example, the protein products of a nucleotide sequence should always be from the same biological source (organism, tissue) as the nucleotide itself. And the publication that describes the sequencing of the DNA in many cases also discusses the translated proteins. By placement of these items as descriptors at the Nuc-prot set level, only one copy of each item is needed to properly describe all the sequences.

BioSource: The Biological Source

The BioSource includes information on the source organism (scientific name and common name), its lineage in the NCBI integrated taxonomy, and its nuclear and (if appropriate) mitochondrial genetic code. It also includes information on the location of the sequence in the cell (e.g., nuclear genome or mitochondrion) and additional modifiers (e.g., strain, clone, isolate, chromosomal map location).

A sequence record for a gene and its protein product will typically have a single

BioSource descriptor at the Nuc-prot set level. A population or phylogenetic study, however, will have BioSource descriptors for each component. (The components can be nucleotide Bioseqs or they can themselves be Nuc-prot sets.) The BioSources in a population study will have the same organism name and usually will be distinguished from each other by modifier information, such as strain or clone name.

MolInfo: Molecule Information

The MolInfo descriptor indicates the type of molecule [e.g., genomic, mRNA (usually isolated as cDNA), rRNA, tRNA, or peptide], the technique with which it was sequenced (e.g., standard, EST, conceptual translation with partial peptide sequencing for confirmation), and the completeness of the sequence [e.g., complete, missing the left (5' or amino) end, missing both ends]. Each nucleotide and each protein should get its own MolInfo descriptor. Normally, then, this descriptor will not appear attached at the Nuc-prot set level. (it may go on a Seg set, since all parts of a segmented Bioseq should be of the same type.)

3.3.1.9. USING THE MODEL

There are a number of consequences of using the NCBI data model for building databases and generating reports. Some of these are discussed in the remainder of this section.

GenBank Format

GenBank presents a "DNA-centered" view of a sequence record. (GenPept presents the equivalent "protein-centered" view.) To maintain compatibility with these historical views, some mappings are performed between features on different sequences or between overlapping features on the same sequence.

In GenBank format, the protein product of a coding region feature is displayed as a / translation qualifier, not as a sequence that can have its own features. The largest protein feature on the product Bioseq is used as the /product qualifier. Some of the features that are actually annotated on the protein Bioseq in the NCBI data model, such as mature peptide or signal peptide, are mapped onto the DNA coordinate system (through the CDS intervals) in GenBank format.

The Gene feature names a region on a sequence, typically covering anything known to affect that gene's phenotype. Other features contained in this region will pick up a / gene qualifier from the Gene feature. Thus, there is no need to separately annotate the / gene qualifier on the other features.

FASTA Format

FASTA format contains a definition line and sequence characters and may be used as input to a variety of analysis programs _ The definition line starts with a right angle bracket (>) and is usually followed by the sequence identifiers in a parsable form, as in this example:

```
>gi12352912IgbIAFO12433.1IHSDDT2
```

The remainder of the definition line, which is usually a title for the sequence, can be generated by software from features and other information in a Nuc-prot set.

For a segmented Bioseq, each raw Bioseq part can be presented separately, with a dash separating the segments. (The regular BLAST search service uses this method for producing search databases, so that the resulting "hits" will map to individual GenBank records.) The segmented Bioseq can also be treated as a single sequence, in which case the raw components will be catenated. (This form is used for generating the BLAST neighbors in Entrez;

BLAST

The Basic Local Alignment Search Tool (BLAST; Altschul et al., 1990) is a popular method of ascertaining sequence similarity. The BLAST program takes a query sequence supplied by the user and searches it against the entire database of sequences maintained at NCBI. The output for each "hit" is a Seq-align, and these are combined into a Seq-annot.

The resulting Seq-annot can be used to generate the traditional BLAST printed report, but it is much more useful when viewed with software tools such as Entrez and Sequin. The viewer in these programs is now designed to display alignment information in useful forms. For example, the Graphical view shows only insertions and deletions relative to the query sequence, whereas the Alignment view fetches the individual sequences and displays mismatches between bases or residues in aligned regions. The Sequence view shows the alignment details at the level of individual bases or residues. This ability to zoom in from an overview to fine details makes it much easier to see the relationships between sequences than with a single report.

Finally, the Seq-annot, or any of its Seq-aligns, can be passed to other tools (such as banded or gapped alignment programs) for refinement. The results may then be sent back into a display program.

Entrez

The Entrez sequence retrieval program (Schuler et al., 1996) was designed to take advantage of connections that are captured by the NCBI data model. For example, the publication in a sequence record may contain a MEDLINE DID or PubMed ID. These are direct links to the PubMed article, which Entrez can retrieve. In addition, the product Seq-loc of a Coding Region feature points to the protein product Bioseq, which Entrez can also retrieve. The links in the data model allow retrieval of linked records at the touch of a button. The Genomes division in Entrez takes further advantage of the data model by providing "on the fly" display of certain regions of large genomes, as is the case when one hits the ProtTable button in Web Entrez.

Sequin

Sequin is a submission tool that takes raw sequence data and other biological information and assembles a record (usually a Bioseq-set) for submission to one of the DDBJ/EMBL/GenBank databases). It makes full use of the NCBI data model and takes advantage of redundant information to validate entries. For example, because the user supplies both the nucleotide and protein sequences, Sequin can determine the coding region location (one or more intervals on the nucleotide that, through the genetic code, produce the protein product). It compares the translation of the coding region to the supplied protein and reports any discrepancy. It also makes sure that each Bioseq has BioSource information applied to it. This requirement can be satisfied for a nucleotide and its protein products by placing a single BioSource descriptor on the Nuc-prot set.

Sequin's viewers are all interactive, in that double-clicking on an existing item (shown as a GenBank flatfile paragraph or a line in the graphical display of features on a sequence) will launch an editor for that item (e.g., feature, descriptor, or sequence data).

LocusLink

LocusLink is, an NCBI project to link information applicable to specific genetic loci from several disparate databases. Information maintained by LocusLink includes official nomenclature, aliases, sequence accessions (particularly RefSeq accessions), phenotypes, Enzyme Commission numbers, map information, and Mendelian Inheritance in Man numbers. Each locus is assigned a unique identification number, which additional databases can then reference.

Summary

The NCBI data model is a natural mapping of how biologists think of sequence relationships and how they annotate these sequences. The data that results can be saved, passed to other analysis programs, modified, and then displayed, all without having to go through multiple format conversions. The model definition concentrates on fundamental data elements that can be measured in a laboratory, such as the sequence of an isolated molecule. As new biological concepts are defined and understood, the specification for data can be easily expanded without the need to change existing data. Software tools are stable over time, and only incremental changes are needed for a program to take advantage of new data fields. Separating the specification into domains (e.g., citations, sequences, structures, maps) reduces the complexity of the data model. Providing neighbors and links between individual records increases the richness of the data and enhances the likelihood of making discoveries from the databases.

Model Questions

1. **what is a datamodel and its role in a database?**
2. **what is a gi number ?**
3. **what are the various types of sequences ?**
4. **what is meant by annotation? Explain some of the features?**

References

1. Bioinformatics, A practical guide to the Analysis of Genes and Proteins by Andreas D. Baxevanis, B.F. Francis Oullette.
2. Altschul, S. F, Gish, w., Miller, W., Meyers, E. w., and Lipman, D. J. (1990). Basic Local Alignment Search Tool. 1. *Mol. Biol.* 215,403-410.
3. Crick, F H. c., Barnett, L., Brenner. 5., and Walls-Tobin, R. J. (1961). General nature of the genetic code for proteins. *Nature* t 92, 1227-1232.
4. Ostell, J. M. (1995). Integrated access, to heterogeneous biomedical data from NCBI. *IEEE Eng.Med. Biol.* 14.730--736.
5. Ostell. J. M. (1996) The NCBI software tools. In *Nucleic Acid and Protein Analysis: A Practical Approach*. M. Bishop and C. Rawlings, Eds. (IRL Press, Oxford), p. 31-43.

Lesson - 3.3.2

GenBank Datamodel

<u>3.3.2.1</u>	Introduction
<u>3.3.2.2</u>	Objective
<u>3.3.2.3</u>	Using the GenBank Database
<u>3.3.2.4</u>	Model Questions
<u>3.3.2.5</u>	Summary
<u>3.3.2.6</u>	Reference

3.3.2.1 Introduction

GenBank is the NIH genetic sequence database, an annotated collection of all publicly available DNA sequences. A new release is made every two months. GenBank is part of the International Nucleotide Sequence Database Collaboration, which is comprised of the DNA DataBank of Japan (DDBJ), the European Molecular Biology Laboratory (EMBL), and GenBank at the National Center for Biotechnology Information. These three organizations exchange data on a daily basis.

Each GenBank entry includes a concise description of the sequence, the scientific name and taxonomy of the source organism, and a table of features that identifies coding regions and other sites of biological significance, such as transcription units, sites of mutations or modifications, and repeats. Protein translations for coding regions are included in the feature table. Bibliographic references are included along with a link to the Medline unique identifier for all published sequences.

3.3.2.2 Objective

- To know what a GenBank Database is.
- To study the various features and attributes of the GenBank .
- To understand the GenBank databank model.

3.3.2.3 Using the GenBank Database

3.3.2.3.1 Searching the database with a query sequence

There are a number of programs that can be used to search the GenBank database with a query sequence. See the list of database searching software or the GCG Wisconsin software for more information.

3.3.2.3.2 Retrieving a database entry

On the Sequence Analysis Resource through the GCG Wisconsin software.
On the VMS front ends through the GCG Wisconsin software.

Version

The current release numbers can be obtained when the GCG Wisconsin package is initialized. In addition, the release information can be found at the top of every GenBank flat file. The corresponding live record for U49845 can be viewed in Entrez. Examples of other records that show a range of biological features are listed below.

A nucleotide sequence identification number that represents a single, specific sequence in the GenBank database. This identification number uses the accession.version format implemented by GenBank/EMBL/DDBJ in February 1999.

If there is any change to the sequence data (even a single base), the version number will be increased, e.g., U12345.1 → U12345.2, but the accession portion will remain stable.

The accession.version system of sequence identifiers runs parallel to the GI number system, i.e., when any change is made to a sequence, it receives a new GI number AND an increase to its version number.

For more information, see section 1.3.2 of the GenBank 111.0 release notes, and section 3.4.7 of the current GenBank release notes.

A Sequence Revision History tool is available to track the various GI numbers, version numbers, and update dates for sequences that appeared in a specific GenBank record (more information and example).

More details about sequence identification numbers and the difference between GI number and version are provided in Sequence Identifiers: A Historical Note.

Entrez Search Field: use the default setting of "All Fields"

Accession [ACCN]

Contains the unique accession number of the sequence or record, assigned to the nucleotide, protein, structure, genome record, or PopSet by a sequence database builder. The Structure database accession index contains the PDB IDs but not the MMDB IDs.

The unique identifier for a sequence record. An accession number applies to the complete record and is usually a combination of a letter(s) and numbers, such as a single letter followed by five digits (e.g., U12345) or two letters followed by six digits (e.g., AF123456). Some accessions might be longer, depending on the type of sequence record.

Accession numbers do not change, even if information in the record is changed at the author's request. Sometimes, however, an original accession number might become secondary to a newer accession number, if the authors make a new submission that combines previous sequences, or if for some reason a new submission supercedes an earlier record.

Records from the RefSeq database of reference sequences have a different accession number format that begins with two letters followed by an underscore bar and six or more digits, for example:

NT_123456 constructed genomic contigs

NM_123456 mRNAs

NP_123456 proteins

NC_123456 chromosomes

Note: compare accession number with Sequence Identifiers such as Version and GI for nucleotide sequences and protein_id and GI for amino acid sequences.

Search Tip: The letters in the accession number can be written in upper- or lowercase.

RefSeq accessions must contain an underscore bar between the letters and the

numbers, e.g., NM_002111. It is better to search for the actual accession number rather than the locus name, because the accessions are stable and locus names can change.

All Fields [ALL]

Contains all terms from all searchable database fields in the database.

GenInfo [GI]

"GenInfo Identifier" sequence identification number, in this case, for the nucleotide sequence. If a sequence changes in any way, a new GI number will be assigned. A separate GI number is also assigned to each protein translation within a nucleotide sequence record, and a new GI is assigned if the protein translation changes in any way (see below).

GI sequence identifiers run parallel to the new accession.version system of sequence identifiers. For more information, see the description of Version, above, and section 3.4.7 of the current GenBank release notes. A Sequence Revision History tool is available to track the various GI numbers, version numbers, and update dates for sequences that appeared in a specific GenBank record (more information and example).
Entrez Search Field: use the default setting of "All Fields".

Reference

Publications by the authors of the sequence that discuss the data reported in the record. References are automatically sorted within the record based on date of publication, showing the oldest references first.

Some sequences have not been reported in papers and show a status of "unpublished" or "in press". When an accession number and/or sequence data has appeared in print, sequence authors should send the complete citation of the article to update@ncbi.nlm.nih.gov and the GenBank staff will revise the record.

Various classes of publication can be present in the References field, including journal article, book chapter, book, thesis/monograph, proceedings chapter, proceedings from a meeting, and patent.

The **last citation** in the REFERENCE field usually contains information about the submitter of the sequence, rather than a literature citation. It is therefore called the "**submitter block**" and shows the words "**Direct Submission**" instead of an article title. Additional information is provided below, under the header Direct Submission. Some older records do not contain a submitter block.

Entrez Search Field: The various subfields under References are searchable in the Entrez search fields noted below.

Author Name [AUTH]

Contains all authors from all references in the database records. The format is last name space first initial(s), without punctuation (e.g., marley jf).

EC/RN Number [ECNO]

Number assigned by the Enzyme Commission or Chemical Abstract Service (CAS) to designate a particular enzyme or chemical, respectively.

PubMed Identifier [PMID]

References that include PubMed IDs contain links from the sequence record to the corresponding PubMed record. Conversely, PubMed records that contain accession number(s) in the SI (secondary source identifier) field contain links back to the sequence record(s).

Entrez Search Field: It is not possible to search the Nucleotide or Protein sequence databases by PubMed ID. However, you can search the PubMed (literature) database of Entrez for the PubMed ID, and then link to the associated sequence records.

Direct Submission

Contact information of the submitter, such as institute/department and postal address. This is always the last citation in the References field. Some older records do not contain the "Direct Submission" reference. However, it is required in all new records.

The Authors subfield contains the submitter name(s), Title contains the words "Direct Submission", and Journal contains the address. The date in the Journal subfield is the date on which the author prepared the submission. In many cases, it is also the date on which the sequence was received by the GenBank staff, but it is not the date of first public release. If you need to know the latter, send a message to info@ncbi.nlm.nih.gov. We will check the history of the record for you.

Entrez Search Field: Use the Author Field [AUTH] if searching for the author name. Use All Fields [ALL] if searching for an element of the author's address (e.g., Yale University). Note, however, that retrieved records might contain the institution name in a field such as Comment, rather than in the Direct Submission reference, so you might get some false hits. Search Tip: It is sometimes helpful to search for both the full spelling and an abbreviation, e.g., "Washington University" OR "WashU", because the spelling used by authors might vary.

Feature Key [FKEY]

Contains the biological features assigned or annotated to the nucleotide sequences and defined in the DDBJ/EMBL/GenBank Feature Table (<http://www.ncbi.nlm.nih.gov/collab/FT/index.html>). Not available for the Protein or Structure databases.

Information about genes and gene products, as well as regions of biological significance reported in the sequence. These can include regions of the sequence that code for proteins and RNA molecules, as well as a number of other features. (See section 3.4.12 of the GenBank release notes for more info.)

A complete list of features is available in the following places:

- Appendix III: Feature keys reference of the DDBJ/EMBL/GenBank Feature Table provides definitions, optional qualifiers, and comments for each feature. An alphabetical list is also available. Appendix IV: Summary of qualifiers for feature keys provides definitions for the Feature qualifiers.
- Sequin Help documentation (scroll down to 'Features' in the table of contents to see an alphabetical list of features with links to descriptions)
- section 3.4.12.1 of the GenBank release notes

The location of each feature is provided as well, and can be a single base, a contiguous span of bases, a joining of sequence spans, and other representations. If a feature is located on the complementary strand, the word "complement" will appear before the base span. If the "<" symbol precedes a base span, the sequence is partial on the 5' end (e.g., CDS <1..206). If the ">" symbol follows a base span, the sequence is partial on the 3' end (e.g., CDS 435..915>). For more information about feature locations, see the Sequin Help Documentation and section 3.4.12.2 of the GenBank release notes.

The sample record shown here only includes a small number of features (source, CDS, and gene, all of which are described below). The Other Features section, below, provides links to some GenBank records that show a variety of additional features. Search Tip: To scroll through the list of available features, view the Feature Key field in Index mode. You can then select one or more features from the index to include in your query. For example, you can limit your search to records that contain both primer_bind and promoter features.

Filter [FILT]

Contains predetermined or filtered subsets of the various databases. These subsets or filters are created by grouping records that are commonly linked to other Entrez databases or within the same database.

For example, the PopSet database Filter index includes PopSet all, PopSet medline, PopSet nucleotide, and PopSet protein. The PopSet medline filter includes all PopSet records with links to PubMed; the PopSet nucleotide filter includes all PopSet records with links to the nucleotide database; and, the PopSet protein filter includes all PopSet records with links to the protein database. The PopSet all filter includes all PopSet records. The Nucleotide database Filter index contains a great deal more filters because the database records are linked to numerous external links.

Gene Name[GENE]

Contains the standard and common names of genes found in the database records. This field is not available in Structure database.

A region of biological interest identified as a gene and for which a name has been assigned. The base span for the gene feature is dependent on the furthest 5' and 3' features. Additional examples of records that show the relationship between gene features and other features such as mRNA and CDS are AF165912 and AF090832.

Complement

Indicates that the feature is located on the complementary strand.

Search Tip: You can use this field to limit your search to records that contain a particular feature, such as a gene. To scroll through the list of available features, view the Feature Key field in Index mode. A complete list of features is also available from the resources noted above.

Issue [ISS]

Contains the issue number of the journal in which the data were published.

Journal Name [JOUR]

Contains the name of the journal in which the data were published. Journal names are indexed in the database in abbreviated form (e.g., J Biol Chem). Journals are also indexed by their by ISSNs. Browse the index if you do not know the ISSN or are not sure how a particular journal name is abbreviated.

Keyword [KYWD]

Contains special index terms from the controlled vocabularies associated with the GenBank, EMBL, DDBJ, SWISS-Prot, PIR, PRF, or PDB databases. Browse the Keyword indexes of the individual databases to become familiar with these vocabularies. A Keyword index is not available in the Structure database.

Word or phrase describing the sequence. If no keywords are included in the entry, the field contains only a period. The Keywords field is present in sequence records primarily for historical reasons, and is not based on a controlled vocabulary. Keywords are generally present in older records. They are **not** included in newer records unless: (1) they are not redundant with any feature, qualifier, or other information present in the record; or (2) the submitter specifically asks for them to be added and #1 is true; or (3) the record contains a special type of sequence such as EST, STS, GSS, HTG, etc. Search Tip: Because keywords are not present in many records, it is best not to search that field. Instead, search All Fields [ALL], the Text Word [WORD] field, or the Title Word [TITL] field, for progressively narrower retrieval.

Modification Date [MDAT]

Contains the date that the most recent modification to that record is indexed in Entrez, in the format YYYY/MM/DD (e.g., 1999/08/05). A year alone, (e.g., 1999) will retrieve all records modified for that year; a year and month (e.g., 1999/03) retrieves all records modified for that month that are indexed in Entrez.

The date in the LOCUS field is the **date of last modification**. The sample record shown here was last modified on 21-JUN-1999.

In some cases, the modification date might correspond to the release date, but there is no way to tell just by looking at the record. If you need to know the first date of public availability for a specific sequence record, send a message to info@ncbi.nlm.nih.gov. We will check the history of the record for you, and let you know the date of first public release. If the sequence was originally submitted to our collaborators at DDBJ or EMBL, rather than to GenBank, we will ask them to send the release date information to you. (See also notes re: date in the Direct Submission reference.)

Search Tips: (1) Enter search term in the format: yyyy/mm/dd, e.g., 1999/07/25. (2) To retrieve records modified between two dates, use the colon as a range operator, e.g., 1999/07/25:1999/07/31[MDAT]. (3) You can use the Publication Date [PDAT] field of Entrez to limit search results by the date on which records were added to the Entrez system. Publication date can be in the form of a range, just like the Modification Date.

Molecular Weight [MOLWT]

Molecular weight of a protein, in Daltons (Da). Note that molecular weight must be entered as a fixed 6 digit field, filled with leading zeros (not letter O), e.g., 002002 [MOLWT]

Organism [ORGN]

Contains the scientific and common names for the organisms associated with protein and nucleotide sequences.

Free-format information including an abbreviated form of the organism name, sometimes followed by a molecule type. (See section 3.4.10 of the GenBank release notes for more info.)

Search Tip: For some organisms that have well-established common names, such as baker's yeast, mouse, and human, a search for the common name will yield the same results as a search for the scientific name, e.g., a search for "baker's yeast" in the organism field retrieves the same number of documents as "Saccharomyces cerevisiae". This is true because the Organism field is connected to the NCBI Taxonomy Database, which contains cross-references between common names, scientific names, and synonyms for organisms represented in the Sequence databases.

The formal scientific name for the source organism (genus and species, where appropriate) and its lineage, based on the phylogenetic classification scheme used in the NCBI Taxonomy Database. If the complete lineage of an organism is very long, an abbreviated lineage will be shown in the GenBank record and the complete lineage will be available in the Taxonomy Database. (See also the /db_xref=taxon:nnnn Feature qualifier, below.)

Search Tip: You can search the Organism field by any node in the taxonomic hierarchy, e.g., you can search for the term "Saccharomyces cerevisiae", "Saccharomycetales", "Ascomycota", etc. to retrieve all the sequences from organisms in a particular taxon.

Taxon

A stable unique identification number for the taxon of the source organism. A taxonomy ID number is assigned to each taxon (species, genus, family, etc.) in the NCBI Taxonomy Database. See also the Organism field, above.

Entrez Search Field: The Taxonomy ID number is not searchable in the Organism search field of Entrez but is searchable in the Taxonomy Browser.

Note: The /db_xref qualifier is one of many that can be applied to various features. A complete list is available in Appendix IV: Summary of qualifiers for feature keys of the DDBJ/EMBL/GenBank Feature Table, and in section 3.4.12.3 of the GenBank release notes. Appendix III: Feature keys reference shows which qualifiers can be used with specific features.

Page Number [PAGE]

Contains the number of the first journal page of the article in which the data were published.

Primary Accession [PACC]

Contains the primary accession number of the sequence or record, assigned to the nucleotide, protein, structure, genome record, or PopSet by a sequence database builder. A Primary Accession index is not available in the Structure database.

Properties [PROP]

Contains properties of the nucleotide or protein sequence. For example, the Nucleotide database's Properties index includes molecule types, publication status,

molecule locations, and GenBank divisions. A Properties index is not available in the Structure database.

Protein Name [PROT]

Contains the standard names of proteins found in database records. Common names may not be indexed in this field so it is best to also consider All Fields or Text Words. A Protein Name index is not available in the Structure database.

The type of molecule that was sequenced. In this example, the molecule type is DNA.

Each GenBank record must contain contiguous sequence data from a single molecule type. The various molecule types are described in the Sequin documentation and can include genomic DNA, genomic RNA, precursor RNA, mRNA (cDNA), ribosomal RNA, transfer RNA, small nuclear RNA, and small cytoplasmic RNA.

Search Tip: Search term should be in the format: biomol_genomic, biomol_mRNA, etc. For more examples, view the Properties field in the Index mode.

Publication Date [PDAT]

Contains the date that records are released into Entrez, in the format YYYY/MM/DD (e.g., 1999/08/05). It is the date the entry first appeared in GenBank explicitly indexed in Entrez. A year alone, (e.g., 1999) will retrieve all records for that year; a year and month (e.g., 1999/03) will retrieve all records released into GenBank for that month.

SeqID String [SQID]

Contains the special string identifier, similar to a FASTA identifier, for a given sequence. A SeqID String index is not available in the Structure database.

Sequence Length [SLEN]

Contains the total length of the sequence. Sequence Length indexes are not available in the Structure or PopSet databases.

Number of nucleotide base pairs (or amino acid residues) in the sequence record. In this example, the sequence length is 5028 bp.

There is no maximum limit on the size of a sequence that can be submitted to GenBank. You can submit a whole genome if you have a contiguous piece of sequence from a single molecule type. However, there is a limit of 350 kb on an individual GenBank record (with some exceptions, as noted in section 1.3.2 of the release notes for GenBank 112.0 target="one"). That limit was agreed upon by the international collaborating sequence databases to facilitate handling of sequence data by various software programs. (For more information, see NCBI News articles on Complete Genomes and GenBank Enters Megabase Era.) The minimum length required for submission is 50 bp, although there might be some shorter records from past years.

Search Tips: (1) To retrieve records within a range of lengths, use the colon as the range operator, e.g., 2500:2600[SLEN]. (2) To retrieve all sequences shorter than a certain number, use 2 as the lower bound, e.g., 2:100[SLEN]. (3) To retrieve all sequences longer than a certain number, use a series of 9's as the upper bound, e.g., 325000:99999999[SLEN].

Substance Name [SUBS]

Contains the names of any chemicals associated with this record from the CAS registry and the MEDLINE Name of Substance field. Substance Name indexes are not available in the Genome or PopSet databases.

Text Word [WORD]

Contains all of the "free text" associated with a record.

Title Word [TITL]

Includes only those words found in the definition line of a record. The definition line summarizes the biology of the sequence and is carefully constructed by database staff. A standard definition line will include the organism, product name, gene symbol, molecule type and whether it is a partial or complete cds. Title Word indexes are not available in the Structure or PopSet databases.

Brief description of sequence; includes information such as source organism, gene name/protein name, or some description of the sequence's function (if the sequence is non-coding). If the sequence has a coding region (CDS), description may be followed by a completeness qualifier, such as "complete cds". (See GenBank release notes section 3.4.5 for more info.)

Title of the published work or tentative title of an unpublished work.

Sometimes the words "**Direct Submission**" instead of an article title. This is usually true for the last citation in the REFERENCE field because it tends to contain information about the submitter of the sequence, rather than a literature citation. The last citation is therefore called the "**submitter block**". Additional information is provided below, under the header Direct Submission. Some older records do not contain a submitter block.

Entrez Search Field: Text Word [WORD]

Note: For sequence records, the Title Word [TITL] field of Entrez searches the Definition Line, not the titles of references listed in the record. Therefore, use the Text Word field to search the titles of references (and other text-containing fields). Search Tip: If a search for a specific term does not retrieve the desired records, try other terms that authors might have used, such as synonyms, full spellings, or abbreviations. The 'related records' (or 'neighbors') function of Entrez also allows you to broaden your search by retrieving records with similar sequences, regardless of the descriptive terms used by the submitters.

Search Tip: Although nucleotide definition lines follow a structured format, GenBank does not use a controlled vocabulary, and authors determine the content of their records. Therefore, if a search for a specific term does not retrieve the desired records, try other terms that authors might have used, such as synonyms, full spellings, or abbreviations. The "related records" (or "neighbors") function of Entrez also allows you to broaden your search by retrieving records with similar sequences, regardless of the descriptive terms used by the submitters.

Uid [UID]

Contains the Medline unique identifier for records that contain published references that are linked to PubMed. The Uid index is not browsable.

Volume [VOL]

Contains the volume number of the journal in which the data were published.

Locus

The LOCUS field contains a number of different data elements, including locus name, sequence length, molecule type, GenBank division, and modification date. Each element is described below.

Locus Name

The locus name in this example is SCU49845.

The locus name was originally designed to help group entries with similar sequences: the first three characters usually designated the organism; the fourth and fifth characters were used to show other group designations, such as gene product; for segmented entries, the last character was one of a series of sequential integers. (See GenBank release notes section 3.4.4 for more info.)

However, the 10 characters in the locus name are no longer sufficient to represent the amount of information originally intended to be contained in the locus name. The only rule now applied in assigning a locus name is that it must be unique. For example, for GenBank records that have 6-character accessions (e.g., U12345), the locus name is usually the first letter of the genus and species names, followed by the accession number. For 8-character character accessions (e.g., AF123456), the locus name is just the accession number.

The RefSeq database of reference sequences assigns formal locus names to each record, based on gene symbol. RefSeq is separate from the GenBank database, but contains cross-references to corresponding GenBank records.

GenBank Division

The GenBank division to which a record belongs is indicated with a three letter abbreviation. In this example, GenBank division is PLN.

The GenBank database is divided into 18 divisions:

1. PRI - primate sequences
2. ROD - rodent sequences
3. MAM - other mammalian sequences
4. VRT - other vertebrate sequences
5. INV - invertebrate sequences
6. PLN - plant, fungal, and algal sequences
7. BCT - bacterial sequences
8. VRL - viral sequences
9. PHG - bacteriophage sequences
10. SYN - synthetic sequences
11. UNA - unannotated sequences
12. EST - EST sequences (expressed sequence tags)

13. PAT - patent sequences
14. STS - STS sequences (sequence tagged sites)
15. GSS - GSS sequences (genome survey sequences)
16. HTG - HTG sequences (high-throughput genomic sequences)
17. HTC - unfinished high-throughput cDNA sequencing
18. ENV - environmental sampling sequences

Some of the divisions contain sequences from specific groups of organisms, whereas others (EST, GSS, HTG, etc.) contain data generated by specific sequencing technologies from many different organisms. The organismal divisions are historical and do not reflect the current NCBI Taxonomy. Instead, they merely serve as a convenient way to divide GenBank into smaller pieces for those who want to FTP the database. Because of this, and because sequences from a particular organism can exist in technology-based divisions such as EST, HTG, etc., the NCBI Taxonomy Browser should be used for retrieving all sequences from a particular organism.

The RNA division of GenBank was removed in release 113.0 (August 1999). Sequences that were previously in the RNA division have been moved to the appropriate organismal division. (See section 1.3.2 of the GenBank 113.0 release notes for additional information.) The HTC division was added to GenBank in release 123.0 (April 2001) and is described in Section 1.3.3 of the GenBank 123.0 release notes.

An 18th division, called CON, was added in release 115.0 (December 1999) but is not listed above because it is still experimental. Records in that division contain no sequence data. Instead, they contain instructions on how to construct contigs from multiple GenBank records. See the Fall 1999 NCBI News and section 1.3.3 of GenBank 115.0 release notes for details.

Search Tip: Search term should be in the format: gbdiv_pri, gbdiv_est, etc. For more examples, view the Properties field in the Index mode. For example, to eliminate all sequences from a particular division, such as all ESTs, you can use a Boolean query formatted

such as:

human[ORGN] NOT gbdiv_est[PROP]

For the reasons noted above, do not use GenBank divisions to retrieve all sequences from a specific organism. Instead, use the NCBI Taxonomy Browser.

Coding sequence [CDS]

Region of nucleotides that corresponds with the sequence of amino acids in a protein (location includes start and stop codons). The CDS feature includes an amino acid translation. Authors can specify the nature of the CDS by using the qualifier "/evidence=experimental" or "/evidence=not_experimental".

Submitters are also encouraged to annotate the mRNA feature, which includes the 5' untranslated region (5'UTR), coding sequences (CDS, exon), and 3' untranslated region (3'UTR). Entrez Search Field: Feature Key [FKEY]

Search Tip: You can use this field to limit your search to records that contain a particular feature, such as CDS. To scroll through the list of available features, view the Feature Key field in Index mode. A complete list of features is also available from the resources noted above.

<1.. 206

Base span of the biological feature indicated to the left, in this case, a CDS feature. (The CDS feature is described above, and its base span includes the start and

stop codons.) Features can be complete, partial on the 5' end, partial on the 3' end, and/or on the complementary strand. Examples:

1. **complete** feature is simply written as ***n..m***

Example: 687..3158

The feature extends from base 687 through base 3158 in the sequence shown

2. **<** indicates **partial on the 5' end**

Example: <1..206

The feature extends from base 1 through base 206 in the sequence shown, and is partial on the 5' end

3. **>** indicates **partial on the 3' end**

Example: 4821..5028>

The feature extends from base 4821 through base 5028 and is partial on the 3' end

(complement) indicates that the feature is on the complementary strand

Example:

complement(3300..4037)

The feature extends from base 3300 through base 4037 but is actually on the complementary strand. It is therefore read in the opposite direction on the reverse complement sequence. (For an example, see the third CDS feature in the sample record shown on this page. In this case, the amino acid translation is generated by taking the reverse complement of bases 3300 to 4037 and reading that reverse complement sequence in its 5' to 3' direction.)

Translation

The amino acid translation corresponding to the nucleotide coding sequence (CDS). In many cases, the translations are conceptual. Note that authors can indicate whether the CDS is based on experimental or non-experimental evidence.

Entrez Search Field: It is not possible to search the translation subfield using Entrez. If you want use a string of amino acids as a query to retrieve similar protein sequences, use BLAST instead.

Protein_ID

A protein sequence identification number, similar to the Version number of a nucleotide sequence. Protein IDs consist of three letters followed by five digits, a dot, and a version number. If there is any change to the sequence data (even a single amino acid), the version number will be increased, but the accession portion will remain stable (e.g., AAA98665.1 will change to AAA98665.2).

The accession.version format of protein sequence identification numbers was implemented by GenBank/EMBL/DDBJ in February 1999 and runs parallel to the GI number system. More details about sequence identification numbers and the difference between GI number and version are provided in Sequence Identifiers

Entrez Search Field: use the default setting of "All Fields"

Other Features

Examples of other records that show a variety of biological features; a graphic format is also available for each sequence record and visually represents the annotated features:

- **AF165912** (gene, promoter, TATA signal, mRNA, 5'UTR, CDS, 3'UTR) GenBank flat file
- **AF090832** (protein bind, gene, 5'UTR, mRNA, CDS, 3'UTR) GenBank flat file
- **L00727** (alternatively spliced mRNAs) GenBank flat file

A complete list of features is available from the resources noted above.

3.3.2.4 Self Assessment

1. How did the GenBank Databank originate ?
2. What are the salient features of the GenBank Datamodel?
3. How is the GenBank result evaluated?

3.3.2.5 Summary

Most sequence analysis programs on PSC supercomputers are capable of reading in GenBank data in the GenBank flat file format. The location of the data in the flat file format is built into the MAKSEQ program. However, if you find it necessary to view the GenBank files in the flat file format, they can be found in the AFS directory */afs/psc/biomed/db/genbank*. Accessible from: the Cray C90, the Sequence Analysis Resource, and the VMS front ends.

3.3.2.6 Reference

<http://www.ncbi.nlm.nih.gov/Genbank/index.html>

<http://www.ncbi.nlm.nih.gov/Genbank/GenbankSearch.html>

<http://www.psc.edu/general/software/packages/genbank/genbank.html>

Lesson 3.3.3

DDBJ DATAMODEL

Objective

3.3.3.1 Introduction

3.3.3.2 Format Design

3.3.3.3 Key aspects of this feature table design

3.3.3.4 Feature Table Terminology

3.3.3.5 DDBJ Format

Summary

Model Questions

References

Objective

- To understand the various features of DDBJ
- To know the feature table details
- To know and analyze DDBJ information more effectively

3.3.3.1 Introduction

Nucleic acid sequences provide the fundamental starting point for describing and understanding the structure, function, and development of genetically diverse organisms. The GenBank, EMBL, and DDBJ nucleic acid sequence databanks have from their inception used tables of sites and features to describe the roles and locations of higher order sequence domains and elements within the genome of an organism. In February, 1986, GenBank and EMBL began a collaborative effort (joined by DDBJ in 1987) to devise a common feature table format and common standards for annotation practice.

3.3.3.2 Format Design

The format design is based on a tabular approach and consists of the following items:

Feature key - a single word or abbreviation indicating functional group

Location - instructions for finding the feature

Qualifiers - auxiliary information about a feature

3.3.3.3 Key aspects of this feature table design

- Feature keys allow specific annotation of important sequence features.
- Related features can be easily specified and retrieved. Feature keys are arranged hierarchically, allowing complex and compound features to be expressed. Both location operators and the feature keys show feature relationships even when the features are not contiguous. The hierarchy of feature keys allows broad categories of biological functionality, such as rRNAs, to be easily retrieved.
- Generic feature keys provide a means for entering new or undefined features. A number of "generic" or miscellaneous feature keys have been added to permit annotation of features that cannot be adequately described by existing feature

keys. These generic feature keys will serve as an intermediate step in the identification and addition of new feature keys. The syntax has been designed to allow the addition of new feature keys as they are required.

- More complex locations (fuzzy and alternate ends, for example) can be specified. Each end point of a feature may be specified as a single point, an alternate set of possible end points, a base number beyond which the end point lies, or a region which contains the end point.
- Features can be combined and manipulated in many different ways. The location field can contain operators or functional descriptors specifying what must be done to the sequence to reproduce the feature. For example, a series of exons may be “join”ed into a full coding sequence.
- Standardized qualifiers provide precision and parsibility of descriptive details. A combination of standardized qualifiers and their controlled-vocabulary values enable free-text descriptions to be avoided.
- The nature of supporting evidence for a feature can be explicitly indicated. Features, such as open reading frames or sequences showing sequence similarity to consensus sequences, for which there is no direct experimental evidence can be annotated. Therefore, the feature table can incorporate contributions from researchers doing computational analysis of the sequence databases. However, all features that are supported by experimental data will be clearly marked as such.
- The table syntax has been designed to be machine parsible. A consistent syntax allows machine extraction and manipulation of sequences coding for all features in the table.

3.3.3.4 Feature Table Terminology

The format and wording in the feature table use common biological research terminology whenever possible. For example, an item in the feature table such as:

Key	Location/Qualifiers
CDS	23..400 /product="alcohol dehydrogenase" /gene="adhI"

might be read as:

The feature CDS is a coding sequence beginning at base 23 and ending at base 400, has a product called ‘alcohol dehydrogenase’ and is coded for by a gene called “adhI”.

A more complex description:

Key	Location/Qualifiers
CDS	join(544..589,688..>1032) /product="T-cell receptor beta-chain"

which might be read as:

This feature, which is a partial coding sequence, is formed by joining elements indicated to form one contiguous sequence encoding a product called T-cell receptor beta-chain.

The following sections contain detailed explanations of the feature table design showing conventions for each component of the feature table, examples of how the format might be implemented, a description of the exact column placement of all the data items and examples of complete sequence entries that have been annotated using the new format. The last section of this document describes known limitations of the current feature table design.

This document defines the syntax and vocabulary of the feature table. The syntax is sufficiently flexible to allow expression of a single biological entity in numerous ways. In such cases, the annotation staffs at the databases will propose conventions for standard means of denoting the entities. This feature table format is shared by GenBank, EMBL and DDBJ. Comments, corrections, and suggestions may be submitted to any of the database staffs. New format specifications will be added as needed.

Qualifier values

Since qualifiers convey many different types of information, there are several value formats:

1. Free text
2. Controlled vocabulary or enumerated values
3. Citation or reference numbers
4. Sequences
5. Feature labels

Citation or reference numbers

The citation or published reference number (as enumerated in the entry 'REFERENCE' or 'RN' data item) should be enclosed in square brackets (e.g., [3]) to distinguish it from other numbers.

Sequences

Literal sequence of nucleotide bases e.g., join(12..45,"atgcatt",988..1050) in location descriptors has become illegal starting from implementation of version 2.1 of the Feature Table Definition Document (December 15, 1998)

Qualifier examples

Key	Location/Qualifiers
source	1..1509 /organism="Mus musculus" /strain="CD1" /mol_type="genomic DNA"
promoter	<1..9 /gene="ubc42"
mRNA	join(10..567,789..1320) /gene="ubc42"
CDS	join(54..567,789..1254) /gene="ubc42" /product="ubiquitin conjugating enzyme" /function="cell division control"

Feature labels

The /label= qualifier takes as its value a feature label. Feature labels follow the same naming conventions as other feature table components (e.g., keys and qualifiers). While feature labels are optional, attaching a label to a feature allows it to be referred to unambiguously. For example, the feature label can be used to refer unambiguously to a coding region that exists in a different entry to the exons of which it is comprised.

Location**Purpose**

The location indicates the region of the presented sequence which corresponds to a feature.

Feature table Format

The examples below show the preferred sequence annotations for a number of commonly occurring sequence types. These examples may not be appropriate in all cases but should be used as a guide whenever possible. This section describes the columnar format used to write this feature table in “flat-file” form for distributions of the database.

Format examples

Feature table format example (EMBL):

```

source      1..1859
            /db_xref="taxon:3899"
            /organism="Trifolium repens"
            /tissue_type="leaves"
            /clone_lib="lambda gt10"
            /clone="TRE361"
            /mol_type="genomic DNA"
CDS         14..1495
            /db_xref="MENDEL:11000"
            /db_xref="SWISS-PROT:P26204"
            /note="non-cyanogenic"
            /EC_number="3.2.1.21"
            /product="beta-glucosidase"
            /protein_id="CAA40058.1"
            /translation="MDFIVAIFALFVISSFTITSTNAVEASTLLDIGNLSR.....
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1      10      20      30      40      50      60      70      79

```

Feature table format example (GenBank):

```

source      1..8959
            /organism="Homo sapiens"
            /db_xref="taxon:9606"
            /mol_type="genomic DNA"
gene        212..8668
            /gene="NF1"
CDS         212..8668

```

```

/gene="NF1"
/note="putative"
/codon_start=1
/product="GAP-related protein"
/protein_id="AAA59924.1"

```

```

/translation="MAAHRPVEWVQAVVSRFDEQLPIKTGQQNTHTKVSTE.....

```

```

-----+-----+-----+-----+-----+-----+-----+-----
1      10      20      30      40      50      60      70      79

```

Feature table format example (DDBJ):

```

source      1..2136
             /clone="pK28"
             /organism="Rattus norvegicus"
             /strain="Sprague-Dawley"
             /tissue_type="kidney"
             /mol_type="genomic DNA"
mRNA        19..2128
CDS         31..1212
             /codon_start=1
             /evidence=not_experimental
             /function="Dual specificity protein tyrosine/threonine
             kinase"
             /product="MAP kinase kinase"
             /protein_id="BAA02603.1"
             /translation="MPKKKPTPIQLNPAPDGSAVNGTSSAETNLEALQKKL.....

```

```

-----+-----+-----+-----+-----+-----+-----+-----
1      10      20      30      40      50      60      70      79

```

Definition of line types

The feature table consists of a header line, which contains the column titles for the table, and the individual feature entries. Each feature entry is composed of a feature descriptor line and qualifier and continuation lines, if needed. The feature descriptor line contains the feature's name, key, and location. If the location cannot be contained on the first line of the feature descriptor, it is continued on a continuation line immediately following the descriptor line. If the feature requires further attributes, feature qualifier lines immediately follow the corresponding feature descriptor line (or its continuation). Qualifier information that cannot be contained on one line continues on the following continuation lines as necessary.

Thus, there are 4 types of feature table lines:

Line type	Content	#/entry	#/feature
-----	-----	-----	-----
Header	Column titles	1*	N/A
Feature descriptor	Key and location	1 to many*	1

Feature qualifiers	Qualifiers and values	N/A	0 to many
Continuation lines	Feature descriptor or qualifier continuation	0 to many	0 to many

Example of sequence annotation

The examples below show the preferred sequence annotations for a number of commonly occurring sequence types. These examples may not be appropriate in all cases but should be used as a guide whenever possible.

Eukaryotic gene

```
source      1..1509
            /organism="Mus musculus"
            /strain="CD1"
            /mol_type="genomic DNA"
promoter    <1..9
            /gene="ubc42"
mRNA        join(10..567,789..1320)
            /gene="ubc42"
CDS         join(54..567,789..1254)
            /gene="ubc42"
            /product="ubiquitin conjugating enzyme"
            /function="cell division control"
```

```
/translation="MVSSFLLAEYKNLIVNPSEHFKISVNEDNLTEGPPDTLYQKIDTVLLSVISLLN
EPNPDSPANVDAAKSYRKYLYKEDLESYPMEKSLDECSAEDIEYFKNVPVNVLPVPSDDYED
EEMEDGTYILTYDDEDEEEDDEEMDDE"
```

```
exon        10..567
            /gene="ubc42"
            /number=1
intron      568..788
            /gene="ubc42"
            /number=1
exon        789..1320
            /gene="ubc42"
            /number=2
polyA_signal 1310..1317
            /gene="ubc42"
```

3.3.3.5. DDBJ Format

```
LOCUS      LISOD 756 bp DNA linear BCT 30-JUN-1993
DEFINITION Listeria ivanovii sod gene for superoxide dismutase.
ACCESSION  X64011 S78972
VERSION    X64011.1 GI:44010
KEYWORDS   sod gene; superoxide dismutase.
SOURCE     Listeria ivanovii
           ORGANISM Listeria ivanovii
             Bacteria; Firmicutes; Bacillales; Listeriaceae; Listeria.
```

REFERENCE 1 (bases 1 to 756)
AUTHORS Haas,A. and Goebel,W.
TITLE Cloning of a superoxide dismutase gene from *Listeria ivanovii* by functional complementation in *Escherichia coli* and characterization of the gene product
JOURNAL Mol. Gen. Genet. 231 (2), 313-322 (1992)
MEDLINE 92140371

REFERENCE 2 (bases 1 to 756)
AUTHORS Kreft,J.
TITLE Direct Submission
JOURNAL Submitted (21-APR-1992) J. Kreft, Institut f. Mikrobiologie, Universitaet Wuerzburg, Biozentrum Am Hubland, 8700 Wuerzburg, FRG

FEATURES Location/Qualifiers
source 1..756
/organism="Listeria ivanovii"
/strain="ATCC 19119"
/db_xref="taxon:1638"
/mol_type="genomic DNA"
RBS 95..100
/gene="sod"
gene 95..746
/gene="sod"
CDS 109..717
/gene="sod"
/EC_number="1.15.1.1"
/codon_start=1
/transl_table=11
/product="superoxide dismutase"
/db_xref="GI:44011"
/protein_id="CAA45406.1"
/db_xref="SWISS-PROT:P28763"

/translation="MTYELPKLPYTYDALEPNFDKETMEIH YTKHHNIYVTKLNEAVSGHAELASKP
GEELVANLDSVP E EIRGAVRNHGGGHANHTLFWSSLSPNGGGAPTGNLKAAIESEFGTFDE
FKEKFNA AAAARFGSGWAWLVVNNGKLEIVSTANQDSPLSEGKTPVLGLDVWEHAYYLKFKQ
NRRPEYIDTFWNVINWDERNKRFDAAK"
terminator 723..746
/gene="sod"

BASE COUNT 247 a 136 c 151 g 222 t
ORIGIN
1 cgttatttaa ggtgttcat agttctatgg aaatagggtc tataccttc
gccttacaat
61 gtaatttctt

//

7.3.2 Feature key reference manual

The following manual has been organized according to the following format:

Feature Key	the feature key name
Definition	the definition of the key
Mandatory qualifiers	qualifiers required with the key; if there are no mandatory qualifiers, this field is omitted.
Optional qualifiers	optional qualifiers associated with the key
Organism scope	valid organisms for the key; if the scope is any organism, this field is omitted.
Molecule scope	valid molecule types; if the scope is any molecule type, this field is omitted.
References	citations of published reports, usually supporting the feature consensus sequence
Comment	comments and clarifications
Abbreviations:	
accnum	an entry primary accession number
<amino_acid>	abbreviation for amino acid
<base_range>	location descriptor for a simple range of bases
<bool>	Boolean truth value. Valid values are yes and no
<evidence_value>	value indicating the nature of supporting evidence.
Feature_label	the feature label (follows naming conventions for all feature table components)
<integer>	unsigned integer value
<location>	general feature location descriptor
<modified_base>	abbreviation for modified nucleotide base
[number]	integer representing number of citation in entry's reference list
<repeat_type>	value indicating the organization of a repeated sequence.
"text"	any text or character string. Since the string is delimited by double quotes, double quotes may only appear as part of the string if they appear in pairs.
Feature Key	attenuator
Definition	1) region of DNA at which regulation of termination of transcription occurs, which controls the expression of some bacterial operons; 2) sequence segment located between the promoter and the first structural gene that causes partial termination of transcription
Optional qualifiers	/allele="text" /citation=[number] /db_xref="<database>:<identifier>" /evidence=<evidence_value>

```

/experiment="text"
/gene="text"
/inference="TYPE[ (same species)][:EVIDENCE_BASIS]"
/label=feature_label
/locus_tag="text" (single token)
/map="text"
/note="text"
/old_locus_tag="text" (single token)
/operon="text"
/phenotype="text"

```

Organism scope prokaryotes
Molecule scope DNA
Feature Key C_region
Definition constant region of immunoglobulin light and heavy chains, and T-cell receptor alpha, beta, and gamma chains; includes one or more exons depending on the particular chain

Optional qualifiers /allele="text"
/citation=[number]
/db_xref="<database>:<identifier>"
/evidence=<evidence_value>
/experiment="text"
/gene="text"
/inference="TYPE[(same species)][:EVIDENCE_BASIS]"
/label=feature_label
/locus_tag="text" (single token)
/map="text"
/note="text"
/old_locus_tag="text" (single token)
/product="text"
/pseudo
/standard_name="text"

Parent Key CDS
Organism scope eukaryotes

Feature Key CAAT_signal

Definition CAAT box; part of a conserved sequence located about 75 bp up-stream of the start point of eukaryotic transcription units which may be involved in RNA polymerase binding; consensus=GG(C or T)CAATCT [1,2].

Optional qualifiers /allele="text"
/citation=[number]
/db_xref="<database>:<identifier>"
/evidence=<evidence_value>

/experiment="text"
 /gene="text"
 /inference="TYPE[(same species)][:EVIDENCE_BASIS]"
 /label=feature_label
 /locus_tag="text" (single token)
 /map="text"
 /note="text"
 /old_locus_tag="text" (single token)

Organism scope eukaryotes and eukaryotic viruses

Molecule scope DNA

References [1] Efstratiadis, A. et al. Cell 21, 653-668 (1980)
 [2] Nevins, J.R. "The pathway of eukaryotic mRNA formation"
 Ann Rev Biochem 52, 441-466 (1983)

Feature Key CDS

Definition coding sequence; sequence of nucleotides that corresponds with the sequence of amino acids in a protein (location includes stop codon); feature includes amino acid conceptual translation.

Optional qualifiers /allele="text"
 /citation=[number]
 /codon=(seq:"codon-sequence",aa:<amino_acid>)
 /codon_start=<1 or 2 or 3>
 /db_xref="<database>:<identifier>"
 /EC_number="text"
 /evidence=<evidence_value>
 /exception="text"
 /experiment="text"
 /function="text"
 /gene="text"
 /inference="TYPE[(same species)][:EVIDENCE_BASIS]"
 /label=feature_label
 /locus_tag="text" (single token)
 /map="text"
 /note="text"
 /number=unquoted text (single token)
 /old_locus_tag="text" (single token)
 /operon="text"
 /product="text"
 /protein_id="<identifier>"
 /pseudo
 /ribosomal_slippage
 /standard_name="text"
 /translation="text"
 /transl_except=(pos:<base_range>,aa:<amino_acid>)
 /transl_table =<integer>

Comment	<p>/trans_splicing</p> <p>/codon_start has valid value of 1 or 2 or 3, indicating the offset at which the first complete codon of a coding feature can be found, relative to the first base of that feature;</p> <p>/transl_table defines the genetic code table used if other than the universal genetic code table; genetic code exceptions outside the range of the specified tables are reported in /codon or /transl_except qualifiers</p> <p>/protein_id consists of a stable ID portion (3+5 format with 3 position letters and 5 numbers) plus a version number after the decimal point; when the protein sequence encoded by the CDS changes, only the version number of the /protein_id value is incremented; the stable part of the /protein_id remains unchanged and as a result will permanently be associated with a given protein;</p>
Feature Key	conflict
Definition	independent determinations of the “same” sequence differ at this site or region;
Mandatory qualifiers	/citation=[number]
	Or
	/compare=[accession-number.sequence-version]
Optional qualifiers	<p>/allele="text"</p> <p>/db_xref="<database>:<identifier>"</p> <p>/evidence=<evidence_value></p> <p>/experiment="text"</p> <p>/gene="text"</p> <p>/inference="TYPE[(same species)][[:EVIDENCE_BASIS]]"</p> <p>/label=feature_label</p> <p>/locus_tag="text" (single token)</p> <p>/map="text"</p> <p>/note="text"</p> <p>/old_locus_tag="text" (single token)</p> <p>/replace="text"</p>
Comment	<p>use /replace="" to annotate deletion, e.g.</p> <p>conflict 4..5</p> <p>/replace=""</p>
Feature Key	D-loop
Definition	<p>displacement loop; a region within mitochondrial DNA in which a short stretch of RNA is paired with one strand of DNA, displacing the original partner DNA strand in this region; also used to describe the displacement of a region of one strand of duplex DNA by a single stranded invader</p>

in the reaction catalyzed by RecA protein

Optional qualifiers /allele="text"
 /citation=[number]
 /db_xref="<database>:<identifier>"
 /evidence=<evidence_value>
 /experiment="text"
 /gene="text"
 /inference="TYPE[(same
 species)][:EVIDENCE_BASIS]"
 /label=feature_label
 /locus_tag="text" (single token)
 /map="text"
 /note="text"
 /old_locus_tag="text" (single token)

Molecule scope DNA
 Feature Key D_segment
 Definition Diversity segment of immunoglobulin heavy chain,
 and T-cell receptor beta chain;

Optional qualifiers /allele="text"
 /citation=[number]
 /db_xref="<database>:<identifier>"
 /evidence=<evidence_value>
 /experiment="text"
 /gene="text"
 /inference="TYPE[(same
 species)][:EVIDENCE_BASIS]"
 /label=feature_label
 /locus_tag="text" (single token)
 /map="text"
 /note="text"
 /old_locus_tag="text" (single token)
 /product="text"
 /pseudo
 /standard_name="text"

Parent Key CDS
 Organism scope eukaryotes
 Feature Key enhancer
 Definition a cis-acting sequence that increases the
 utilization of some) eukaryotic promoters, and
 can function in either orientation and in any
 location (upstream or downstream)
 relative to the promoter;

Optional qualifiers /allele="text"
 /bound_moiety="text"

```

/citation=[number]
/db_xref="<database>:<identifier>"
/evidence=<evidence_value>
/experiment="text"
/label=feature_label
/gene="text"
/inference="TYPE[ (same
species)][:EVIDENCE_BASIS]"
/locus_tag="text" (single token)
/map="text"
/note="text"
/old_locus_tag="text" (single token)
/standard_name="text"

```

Organism scope eukaryotes and eukaryotic viruses
Feature Key exon
Definition region of genome that codes for portion of
spliced mRNA, rRNA and tRNA; may contain 5'UTR,
all CDSs and 3' UTR;

Optional qualifiers /allele="text"
/citation=[number]
/db_xref="<database>:<identifier>"
/EC_number="text"
/evidence=<evidence_value>
/experiment="text"
/function="text"
/gene="text"
/inference="TYPE[(same
species)][:EVIDENCE_BASIS]"
/label=feature_label
/locus_tag="text" (single token)
/map="text"
/note="text"
/number=unquoted text (single token)
/old_locus_tag="text" (single token)
/product="text"
/pseudo
/standard_name="text"

Feature key gap
Definition gap in the sequence
Mandatory qualifiers /estimated_length=unknown or <integer>
Optional qualifiers /experiment="text"
/inference="TYPE[(same
species)][:EVIDENCE_BASIS]"
/map="text"
/note="text"

Comment the location span of the gap feature for an unknown gap is 100 bp, with the 100 bp indicated as 100 “n”s in the sequence. Where estimated length is indicated by an integer, this is indicated by the same number of “n”s in the sequence. No upper or lower limit is set on the size of the gap.

Feature Key GC_signal

Definition GC box; a conserved GC-rich region located upstream of the start point of eukaryotic transcription units which may occur in multiple copies or in either orientation;
consensus=GGGCGG;

Optional qualifiers /allele="text"
/citation=[number]
/db_xref="<database>:<identifier>"
/evidence=<evidence_value>
/experiment="text"
/gene="text"
/inference="TYPE[(same species)][:EVIDENCE_BASIS]"
/label=feature_label
/locus_tag="text" (single token)
/map="text"
/note="text"
/old_locus_tag="text" (single token)

Organism scope eukaryotes and eukaryotic viruses

Feature Key gene

Definition region of biological interest identified as a gene and for which a name has been assigned;

Optional qualifiers /allele="text"
/citation=[number]
/db_xref="<database>:<identifier>"
/evidence=<evidence_value>
/experiment="text"
/function="text"
/gene="text"
/inference="TYPE[(same species)][:EVIDENCE_BASIS]"
/label=feature_label
/locus_tag="text" (single token)
/map="text"
/note="text"
/old_locus_tag="text" (single token)
/operon="text"

```

/product="text"
/pseudo
/phenotype="text"
/standard_name="text"
/trans_splicing

```

Comment the gene feature describes the interval of DNA that corresponds to a genetic trait or phenotype; the feature is, by definition, not strictly bound to it's positions at the ends; it is meant to represent a region where the gene is located.

Feature Key iDNA

Definition intervening DNA; DNA which is eliminated through any of several kinds of recombination;

Optional qualifiers /allele="text"
/citation=[number]
/db_xref="<database>:<identifier>"
/evidence=<evidence_value>
/experiment="text"
/function="text"
/gene="text"
/inference="TYPE[(same species)][[:EVIDENCE_BASIS]]"
/label=feature_label
/locus_tag="text" (single token)
/map="text"
/note="text"
/number=unquoted text (single token)
/old_locus_tag="text" (single token)
/standard_name="text"

Molecule scope DNA

Comment e.g., in the somatic processing of immunoglobulin genes.

Feature Key intron

Definition a segment of DNA that is transcribed, but removed from within the transcript by splicing together the sequences(exons) on either side of it;

Optional qualifiers /allele="text"
/citation=[number]
/cons_splice=(5'site:<bool>,3'site:<bool>)
/db_xref="<database>:<identifier>"
/evidence=<evidence_value>
/experiment="text"
/function="text"
/gene="text"
/inference="TYPE[(same

species)][:EVIDENCE_BASIS]"

/label=feature_label

/locus_tag="text" (single token)

/map="text"

/note="text"

/number=unquoted text (single token)

/old_locus_tag="text" (single token)

/pseudo

/standard_name="text"

Comment cons_splice is used only when one of the intron's
Splice sites does not match the GT...AG consensus.

Feature Key J_segment

Definition joining segment of immunoglobulin light and heavy
chains, and T-cell receptor alpha, beta, and
gamma chains;

Optional qualifiers /allele="text"

/citation=[number]

/db_xref="<database>:<identifier>"

/evidence=<evidence_value>

/experiment="text"

/gene="text"

/inference="TYPE[(same species)][:EVIDENCE_BASIS]"

/locus_tag="text" (single token)

/map="text"

/note="text"

/old_locus_tag="text" (single token)

/product="text"

/pseudo

/standard_name="text"

Parent Key CDS

Organism scope eukaryotes

Feature Key LTR

Definition long terminal repeat, a sequence directly
repeated at both ends of a defined sequence, of
the sort typically found in retroviruses;

Optional qualifiers /allele="text"

/citation=[number]

/db_xref="<database>:<identifier>"

/evidence=<evidence_value>

/experiment="text"

/function="text"

/gene="text"

/inference="TYPE[(same
species)][:EVIDENCE_BASIS]"

/label=feature_label
 /locus_tag="text" (single token)
 /map="text"
 /note="text"
 /old_locus_tag="text" (single token)
 /standard_name="text"

Feature Key mat_peptide
Definition mature peptide or protein coding sequence; coding sequence for the mature or final peptide or protein product following post-translational modification; the location does not include the stop codon (unlike the corresponding CDS);

Optional qualifiers /allele="text"
 /citation=[number]
 /db_xref="<database>:<identifier>"
 /EC_number="text"
 /evidence=<evidence_value>
 /experiment="text"
 /function="text"
 /gene="text"
 /inference="TYPE[(same species)][[:EVIDENCE_BASIS]]"
 /label=feature_label
 /locus_tag="text" (single token)
 /map="text"
 /note="text"
 /old_locus_tag="text" (single token)
 /product="text"
 /pseudo
 /standard_name="text"

Feature Key misc_binding
Definition site in nucleic acid which covalently or non-Covalently binds another moiety that cannot be described by any other binding key (primer_bind or protein_bind);

Mandatory qualifiers /bound_moiety="text"

Optional qualifiers /allele="text"
 /citation=[number]
 /db_xref="<database>:<identifier>"
 /evidence=<evidence_value>
 /experiment="text"
 /function="text"
 /gene="text"
 /inference="TYPE[(same

	<pre> species)][:EVIDENCE_BASIS]" /label=feature_label /locus_tag="text" (single token) /map="text" /note="text" /old_locus_tag="text" (single token) </pre>
Comment	note that the key RBS is used for ribosome binding sites
Feature Key	misc_difference
Definition	feature sequence is different from that presented in the entry and cannot be described by any other Difference key (conflict, unsure, old_sequence, variation, or modified_base);
Optional qualifiers	<pre> /allele="text" /citation=[number] /clone="text" /compare=[accession-number.sequence-version] /db_xref="<database>:<identifier>" /evidence=<evidence_value> /experiment="text" /gene="text" /inference="TYPE[(same species)][:EVIDENCE_BASIS]" /label=feature_label /locus_tag="text" (single token) /map="text" /note="text" /old_locus_tag="text" (single token) /phenotype="text" /replace="text" /standard_name="text" </pre>
Comment	the misc_difference feature key should be used to describe variability that arises as a result of genetic manipulation (e.g. site directed mutagenesis);
	<pre> use /replace="" to annotate deletion, e.g. misc_difference 412..433 /replace="" </pre>
Feature Key	misc_feature
Definition	region of biological interest which cannot be Described by any other feature key; a new or rare feature;

Optional qualifiers /allele="text"
 /citation=[number]
 /db_xref="<database>:<identifier>"
 /evidence=<evidence_value>
 /experiment="text"
 /function="text"
 /gene="text"
 /inference="TYPE[(same
 species)][:EVIDENCE_BASIS]"
 /label=feature_label
 /locus_tag="text" (single token)
 /map="text"
 /note="text"
 /number=unquoted text (single token)
 /old_locus_tag="text" (single token)
 /phenotype="text"
 /product="text"
 /pseudo
 /standard_name="text"

Comment this key should not be used when the need is
 merely to mark a region in order to comment on it
 or to use it in another feature's location

Feature Key misc_recomb
 Definition site of any generalized, site-specific or
 Replicative recombination event where there is a
 breakage and reunion of duplex DNA that cannot be
 described by other recombination keys or
 qualifiers of source key (/insertion_seq,
 /transposon, /proviral);

Optional qualifiers /allele="text"
 /citation=[number]
 /db_xref="<database>:<identifier>"
 /evidence=<evidence_value>
 /experiment="text"
 /gene="text"
 /inference="TYPE[(same
 species)][:EVIDENCE_BASIS]"
 /label=feature_label
 /locus_tag="text" (single token)
 /map="text"
 /note="text"
 /old_locus_tag="text" (single token)
 /organism="text"
 /standard_name="text"

Molecule scope DNA

Comment	if no /organism is provided with misc_recomb, this suggests that only one organism (same as in SOURCE) is involved in the recombination event
Feature Key	misc_RNA
Definition	any transcript or RNA product that cannot be defined by other RNA keys (prim_transcript, precursor_RNA, mRNA, 5'clip, 3'clip, 5'UTR, 3'UTR, exon, CDS, sig_peptide, transit_peptide, mat_peptide, intron, polyA_site, rRNA, tRNA, scRNA, and snRNA);

Optional qualifiers /allele="text"
 /citation=[number]
 /db_xref="<database>:<identifier>"
 /evidence=<evidence_value>
 /experiment="text"
 /function="text"
 /gene="text"
 /inference="TYPE[(same species)][:EVIDENCE_BASIS]"
 /label=feature_label
 /locus_tag="text" (single token)
 /map="text"
 /note="text"
 /old_locus_tag="text" (single token)
 /operon="text"
 /product="text"
 /pseudo
 /standard_name="text"
 /trans_splicing

Feature Key	misc_signal
Definition	any region containing a signal controlling or Altering gene function or expression that cannot be described by other signal keys (promoter, CAAT_signal, TATA_signal, -35_signal, -10_signal, GC_signal, RBS, polyA_signal, enhancer, attenuator, terminator, and rep_origin).

Optional qualifiers /allele="text"
 /citation=[number]
 /db_xref="<database>:<identifier>"
 /evidence=<evidence_value>
 /experiment="text"
 /function="text"
 /gene="text"
 /inference="TYPE[(same species)][:EVIDENCE_BASIS]"

/label=feature_label
 /locus_tag="text" (single token)
 /map="text"
 /note="text"
 /old_locus_tag="text" (single token)
 /operon="text"
 /phenotype="text"
 /standard_name="text"

Feature Key misc_structure
 Definition any secondary or tertiary nucleotide structure or conformation that cannot be described by other Structure keys (stem_loop and D-loop);

Optional qualifiers /allele="text"
 /citation=[number]
 /db_xref="<database>:<identifier>"
 /evidence=<evidence_value>
 /experiment="text"
 /function="text"
 /gene="text"
 /inference="TYPE[(same species)][[:EVIDENCE_BASIS]]"
 /label=feature_label
 /locus_tag="text" (single token)
 /map="text"
 /note="text"
 /old_locus_tag="text" (single token)
 /standard_name="text"

Feature Key modified_base

Definition the indicated nucleotide is a modified nucleotide and should be substituted for by the indicated molecule (given in the mod_base qualifier value)
 Mandatory qualifiers /mod_base=<modified_base>

Optional qualifiers /allele="text"
 /citation=[number]
 /db_xref="<database>:<identifier>"
 /evidence=<evidence_value>
 /experiment="text"
 /frequency="text"
 /gene="text"
 /inference="TYPE[(same species)][[:EVIDENCE_BASIS]]"
 /label=feature_label
 /locus_tag="text" (single token)
 /map="text"

/note="text"
 /old_locus_tag="text" (single token)

Comment value is limited to the restricted vocabulary for modified base abbreviations;

Feature Key mRNA
 Definition messenger RNA; includes 5'untranslated region (5'UTR), coding sequences (CDS, exon) and 3'untranslated region (3'UTR);

Optional qualifiers /allele="text"
 /citation=[number]
 /db_xref="<database>:<identifier>"
 /evidence=<evidence_value>
 /experiment="text"
 /function="text"
 /gene="text"
 /inference="TYPE[(same species)][[:EVIDENCE_BASIS]]"
 /label=feature_label
 /locus_tag="text" (single token)
 /map="text"
 /note="text"
 /old_locus_tag="text" (single token)
 /operon="text"
 /product="text"
 /pseudo
 /standard_name="text"
 /trans_splicing

Feature Key N_region
 Definition extra nucleotides inserted between rearranged Immunoglobulin segments.

Optional qualifiers /allele="text"
 /citation=[number]
 /db_xref=":"
 /evidence=<evidence_value>
 /experiment="text"
 /gene="text"
 /inference="TYPE[(same species)][[:EVIDENCE_BASIS]]"
 /label=feature_label
 /locus_tag="text" (single token)
 /map="text"
 /note="text"
 /old_locus_tag="text" (single token)

/product="text"
 /pseudo
 /standard_name="text"

Parent Key CDS
 Organism scope eukaryotes
 Feature Key old_sequence
 Definition the presented sequence revises a previous version
 of the sequence at this location;

Mandatory qualifiers /citation=[number]

Or

/compare=[accession-number.sequence-version]

Optional qualifiers /allele="text"

/db_xref="<database>:<identifier>"

/evidence=<evidence_value>

/experiment="text"

/gene="text"

/inference="TYPE[(same
 species)][:EVIDENCE_BASIS]"

/label=feature_label

/locus_tag="text" (single token)

/map="text"

/note="text"

/old_locus_tag="text" (single token)

/replace="text"

Comment use /replace="" to annotate deletion, e.g.
 old_sequence 12..15
 /replace=""

Feature Key operon

Definition region containing polycistronic transcript
 containing genes that encode enzymes that are
 in the same metabolic pathway and regulatory
 sequences

Mandatory qualifiers /operon="text"

Optional qualifiers /allele="text"

/citation=[number]

/db_xref="<database>:<identifier>"

/evidence=<evidence_value>

/experiment="text"

/function="text"

/inference="TYPE[(same
 species)][:EVIDENCE_BASIS]"

/label=feature_label

/map="text"

/note="text"

/phenotype="text"
 /pseudo
 /standard_name="text"

Feature Key oriT
Definition origin of transfer; region of a DNA molecule where transfer is initiated during the process of conjugation or mobilization

Optional qualifiers /allele="text"
 /bound_moiety="text"
 /citation=[number]
 /db_xref="<database>:<identifier>"
 /direction=value
 /evidence=<evidence_value>
 /experiment="text"
 /gene="text"
 /inference="TYPE[(same species)][:EVIDENCE_BASIS]"
 /label=feature_label
 /locus_tag="text" (single token)
 /map="text"
 /note="text"
 /old_locus_tag="text" (single token)
 /rpt_family="text"
 /rpt_type=<repeat_type>
 /rpt_unit="text" or <base_range>
 /rpt_unit_range=<base_range>
 /rpt_unit_seq="text"
 /standard_name="text"

Molecule Scope DNA
Comments rep_origin should be used for origins of replication;
 /direction has legal values RIGHT, LEFT and BOTH, however only RIGHT and LEFT are valid when used in conjunction with the oriT feature;
 origins of transfer can be present in the chromosome; plasmids can contain multiple origins of transfer

Feature Key polyA_signal
Definition recognition region necessary forendonucleases
 Cleavage of an RNA transcript that is followed by polyadenylation; consensus=AATAAA [1];

Optional qualifiers /allele="text"
 /citation=[number]
 /db_xref="<database>:<identifier>"
 /evidence=<evidence_value>

```

/experiment="text"
/gene="text"
/inference="TYPE[ (same
                species)][:EVIDENCE_BASIS]"
/label=feature_label
/locus_tag="text" (single token)
/map="text"
/note="text"
/old_locus_tag="text" (single token)

```

Modified and unusual Amino Acids

Abbreviation	Amino acid
-----	-----
Aad	2-Aminoadipic acid
bAad	3-Aminoadipic acid
bAla	beta-Alanine, beta-Aminopropionic acid
Abu	2-Aminobutyric acid
4Abu	4-Aminobutyric acid, piperidinic acid
Acp	6-Aminocaproic acid
Ahe	2-Aminoheptanoic acid
Aib	2-Aminoisobutyric acid
bAib	3-Aminoisobutyric acid
Apm	2-Aminopimelic acid
Dbu	2,4-Diaminobutyric acid
Des	Desmosine
Dpm	2,2'-Diaminopimelic acid
Dpr	2,3-Diaminopropionic acid
EtGly	N-Ethylglycine
EtAsn	N-Ethylasparagine
Hyl	Hydroxylysine
aHyl	allo-Hydroxylysine
3Hyp	3-Hydroxyproline
4Hyp	4-Hydroxyproline
Ide	Isodesmosine
alle	allo-Isoleucine
MeGly	N-Methylglycine, sarcosine
Melle	N-Methylisoleucine
MeLys	6-N-Methyllysine
MeVal	N-Methylvaline
Nva	Norvaline
Nle	Norleucine
Orn	Ornithine
OTHER	(requires /note=)

Summary

The overall goal of the feature table design is to provide an extensive vocabulary for describing features in a flexible framework for manipulating them. The Feature Table documentation represents the shared rules that allow the three databases to exchange data on a daily basis.

The range of features to be represented is diverse, including regions which:

- * perform a biological function,
- * affect or are the result of the expression of a biological function,
- * interact with other molecules,
- * affect replication of a sequence,
- * affect or are the result of recombination of different sequences,
- * are a recognizable repeated unit,
- * have secondary or tertiary structure,
- * exhibit variation, or have been revised or corrected.

Model Questions

1. What is DDBJ? How is it useful for biological analysis?
2. Describe feature table format?
3. What are feature table keys? Explain a few?

References

1. <http://www.ddbj.nig.ac.jp/>
2. Bioninformatics A Practical Guide to the Analysis of Genes and Proteins by Andreas D Baxevanis, B.F. Francis Ouellette.

Author:-

Asha Smitha.B.,
Centre For Biotechnology,
Acharya Nagarjuna University

Lesson 3.3.4**PDB DATAMODEL****Contents**

3.3.4.1	Objective
3.3.4.2	Introduction
3.3.4.3	Biological Unit Description in PDB and mmCIF Files
3.3.4.4	Record Format
3.3.4.5	Types of Records
3.3.4.6	Order of Records
3.3.4.7	Summary
3.3.4.8	Model Questions
3.3.4.9	Referenc

3.3.4.1 Objective

- To study the data model of PDB
- To understand the various record types and formats

3.3.4.2 Introduction

The thousands of proteins in the cells of an organism have intricate and disparate shapes and perform an enormous array of functions—from catalyzing reactions to forming rigid structures, from recognizing infectious agents to transmitting neural impulses. No wonder that biologists have been eager to understand the details of their structures. Solving the first protein structure, in 1957, took 22 years, but now, in the year 2002, the process has become increasingly automated, and nearly 20,000 protein structures are known. For the last 30 years the community of protein biologists has catalogued new structures in the Protein Data Bank, a freely accessible database. The PDB even includes the structures of complex entities, like entire viruses and the ribosome. Plans are underway to fill out the database with enough unique structures to provide a catalog of practically all possible protein formations. The progress in understanding the structural details of life has practical benefits, including rational drug design and the possibility of molecular engineering.

The current release has 34891 entries and was indexed 18-Nov-2005. The Protein Data Bank (PDB) is an archive of experimentally determined three-dimensional structures of biological macromolecules, serving a global community of researchers, educators, and students. The archives contain atomic coordinates, bibliographic citations, primary and secondary structure information, as well as crystallographic structure factors and NMR experimental data. The PDB Newsletter and CD ROM are published quarterly.

The PDB is supported by a combination of Federal Government Agency funds and user fees. Support is provided by the U.S. National Science Foundation, the U.S. Public Health Service, National Institutes of Health, National Center for Research Resources,

National Institutes of General Medical Sciences, National Library of Medicine, and the U.S. Department of Energy under contract DE-AC02-76CH00016 and user fees. The PDB is the single worldwide repository for the processing and distribution of 3-D structure data of large molecules of proteins and nucleic acids. New structures are released each Wednesday by 1:00am Pacific time.

The Protein Data Bank (PDB) is the central worldwide repository for three-dimensional (3D) structure data of biological macromolecules. The Research Collaboratory for Structural Bioinformatics (RCSB) has completely redesigned its resource for the distribution and query of 3D structure data. The re-engineered site is currently in public beta test at <http://pdbeta.rcsb.org>. The new site expands the functionality of the existing site by providing structure data in greater detail and uniformity, improved query and enhanced analysis tools. A new key feature is the integration and searchability of data from over 20 other sources covering genomic, proteomic and disease relationships. The current capabilities of the re-engineered site, which will become the RCSB production site at <http://www.pdb.org> in late 2005, are described.

Introduction to Biological Units and the PDB Archive

When crystallographic structures are deposited in the PDB, the primary coordinate file generally contains one asymmetric unit - a concept that has applicability only to crystallography, but is important to understanding the process in obtaining the functional biological molecule.

This page provides descriptions of the terms asymmetric unit and biological molecule, describes where information about the biological unit can be found in PDB and mmCIF coordinate files, and explains how the biological unit files in the PDB have been derived.

3.3.4.3 Biological Unit Description in PDB and mmCIF Files

PDB Format Coordinate Files

In PDB format files, information about the biological unit is given in Remarks 300 and 350, although in older files other Remarks may be used.

Protein Data Bank Contents Guide:

The Protein Data Bank (PDB) is an archive of experimentally determined three-dimensional structures of biological macromolecules, serving a global community of researchers, educators, and students. The archives contain atomic coordinates, bibliographic citations, primary and secondary structure information, as well as crystallographic structure factors and NMR experimental data.

Entries conforming to this format description have the following remark within them:

```
REMARK 4 XXXX COMPLIES WITH FORMAT V. 2.1, 25-OCT-1996
```

Entries released after October 25, 1996 will comply with this format. Conversion of older entries to this format will begin in the fall of 1996.

The PDB is supported by a combination of Federal Government Agency funds and user fees. Support is provided by the U.S. National Science Foundation, the U.S. Public Health Service, National Institutes of Health, National Center for Research Resources, National Institutes of General Medical Sciences, National Library of Medicine, and the U.S. Department of Energy under contract DE-AC02-76CH00016.

Purpose of this Document

The PDB Contents Guide gives a complete and concise description of the contents of PDB coordinate entry files. This document will be helpful to several communities, assisting depositors in preparing their entries for deposition, guiding software and information resource developers, and helping users of PDB to understand the contents of coordinate entries. Finally, this format description is crucial in the effort to produce CIF-compliant data files from PDB entries.

Changes to PDB Format and to the Contents Guide

When a change is made to PDB format, the format version number, as found in the entry and in this Contents Guide, will be incremented to the next whole number. Changes to the format of PDB coordinate entry files will follow the Format Change Policy presented below and will be detailed in this Contents Guide. Beginning January 1997, the format of all PDB entries will be compliant with the current version of this Contents Guide.

Changes to the Contents Guide will be listed at the beginning in the What's New section and denoted by a fractional increase in the document version number. These changes may be of the following kind.

- * Correction of typographical errors.
- * Changes to the language for clarity.
- * Addition or changes to the examples for better representation of format issues.
- * Addition of new rules (these do not change the format but help to clarify the semantics).
- * Addition of tokens to specification lists, such as in COMPND and SOURCE records, that are needed to more fully describe the structure and its biological source.
- * Enhancements to the refinement and experimental details templates in the REMARK records. These remarks are currently being reviewed by several people in the community, and PDB expects to increase the level of detail archived, such as for NMR studies.
- * Addition of new sections that enhance and expand the document (these may include topics such as PDB to mmCIF cross references or insertion of relevant sections from the PDB Deposition Form).

The PDB format has been in use since the late 1970's. A number of groups including the mmCIF Committee have been looking at ways to upgrade both the file content and the interchange format used by PDB. This is clearly needed due to changes in the data that PDB archives, the size of the database itself, and finally, to allow PDB to use more up-to-date methods for representing and storing biological data.

The PDB plans to be prudent and deliberate in making changes to the current PDB files in order to minimize the need to change existing programs. In particular, we will explore ways and means of ensuring that programs which read the current ATOM/HETATM records can continue to do so in the foreseeable future.

3.3.4.4 Record Format

Every PDB file may be broken into a number of lines terminated by an end-of-line indicator. Each line in the PDB entry file consists of 80 columns. The last character in each PDB entry should be an end-of-line indicator.

Each line in the PDB file is self-identifying. The first six columns of every line contain a record name, left-justified and blank-filled. This must be an exact match to one of the stated record names.

The PDB file may also be viewed as a collection of record types. Each record type consists of one or more lines.

Each record type is further divided into fields.

Each record type is detailed in this document. The description of each record type includes the following sections:

- * Overview
- * Record Format
- * Details
- * Verification/Validation/Value Authority Control
- * Relationship to Other Record Types
- * Example
- * Known Problems

For records that are fully described in fixed column format, columns not assigned to fields *must be left blank*.

3.3.4.5 Types of Records

It is possible to group records into categories based upon how often the record type appears in an entry.

Single

There are records which may only appear one time (without continuations) in a file. Listed alphabetically, these are:

RECORD TYPE	DESCRIPTION

CRYST1	Unit cell parameters, space group, and Z.
END	Last record in the file.
HEADER	First line of the entry, contains PDB ID code, classification, and date of deposition.
MASTER	Control record for bookkeeping.
ORIGXn	Transformation from orthogonal coordinates to the submitted coordinates (n = 1, 2, or 3).
SCALEn	Transformation from orthogonal coordinates to fractional crystallographic coordinates (n = 1, 2, or 3).

It is an error for a duplicate of any of these records to appear in an entry.

Single Continued

There are records that conceptually exist only once in an entry, but the information content may exceed the number of columns available. These records are therefore continued on subsequent lines. Listed alphabetically, these are:

RECORD TYPE	DESCRIPTION
AUTHOR	List of contributors.
CAVEAT	Severe error indicator. Entries with this record must be used with care.
COMPND	Description of macromolecular contents of the entry.
EXPDTA	Experimental technique used for the structure determination.
KEYWDS	List of keywords describing the macromolecule.
OBSLTE	Statement that the entry has been removed from distribution and list of the ID code(s) which replaced it.
SOURCE	Biological source of macromolecules in the entry.
SPRSDE	List of entries withdrawn from release and replaced by current entry.
TITLE	Description of the experiment represented in the entry.

The second and subsequent lines contain a continuation field which is a right-justified integer. This number increments by one for each additional line of the record, and is followed by a blank character.

Multiple

Most record types appear multiple times, often in groups where the information is not logically concatenated but is presented in the form of a list. Many of these record types have a custom serialization that may be used not only to order the records, but also to connect to other record types. Listed alphabetically, these are:

RECORD TYPE	DESCRIPTION
ANISOU	Anisotropic temperature factors.
ATOM	Atomic coordinate records for standard groups.
CISPEP	Identification of peptide residues in cis conformation.
CONNECT	Connectivity records.
DBREF	Reference to the entry in the sequence database(s).
HELIX	Identification of helical substructures.
HET	Identification of non-standard groups or residues (heterogens)
HETSYN	Synonymous compound names for heterogens.
HYDBND	Identification of hydrogen bonds.
LINK	Identification of inter-residue bonds.
MODRES	Identification of modifications to standard residues.
MTRIXn	Transformations expressing non-crystallographic Symmetry (n = 1, 2, or 3). There may be multiple sets of these records.

REVDAT	Revision date and related information.
SEQADV	Identification of conflicts between PDB and the named Sequence database.
SEQRES	Primary sequence of backbone residues.
SHEET	Identification of sheet substructures.
SIGATM	Standard deviations of atomic parameters.
SIGUIJ	Standard deviations of anisotropic temperature factors.
SITE	Identification of groups comprising important sites.
SLTBRG	Identification of salt bridges
SSBOND	Identification of disulfide bonds.
TURN	Identification of turns.
TVECT	Translation vector for infinite covalently connected structures.

Multiple Continued

There are records that conceptually exist multiple times in an entry, but the information content may exceed the number of columns available. These records are therefore continued on subsequent lines. Listed alphabetically, these are:

RECORD TYPE	DESCRIPTION
FORMUL	Chemical formula of non-standard groups.
HETATM	Atomic coordinate records for heterogens.
HETNAM	Compound name of the heterogens.

The second and subsequent lines contain a continuation field which is a right-justified integer. This number increments by one for each additional line of the record, and is followed by a blank character.

Grouping

There are three record types used to group other records. Listed alphabetically, these are:

RECORD TYPE	DESCRIPTION
ENDMDL	End-of-model record for multiple structures in a single coordinate entry.
MODEL	Specification of model number for multiple structures in a single coordinate entry.
TER	Chain terminator.

The MODEL/ENDMDL records surround groups of ATOM, HETATM, SIGATM, ANISOU, SIGUIJ, and TER records. TER records indicate the end of a chain.

Other

The remaining record types have a detailed inner structure. Listed alphabetically, these are:

RECORD TYPE	DESCRIPTION
JRNL	Literature citation that defines the coordinate set.
REMARK	General remarks, some are structured and some are free form.

3.3.4.6 Order of Records

All records in a PDB coordinate entry must appear in a defined order. Mandatory record types are present in all entries. When mandatory data are not provided, the record name must appear in the entry with a NULL indicator. Optional items become mandatory when certain conditions exist. Record order and existence are described in the following table:

RECORD TYPE	EXISTENCE	CONDITIONS IF OPTIONAL
HEADER	Mandatory	
OBSLTE	Optional	Mandatory in withdrawn entries.
TITLE	Mandatory	
CAVEAT	Optional	Mandatory if structure is deemed incorrect by an outside editorial board.
COMPND	Mandatory	
SOURCE	Mandatory	
KEYWDS	Mandatory	
EXPDTA	Mandatory	
AUTHOR	Mandatory	
REVDAT	Mandatory	
SPRSDE	Optional	Mandatory if a replacement entry.
JRNL	Optional	Mandatory if a publication describes the experiment.
REMARK 1	Optional	
REMARK 2	Mandatory	
REMARK 3	Mandatory	
REMARK N	Optional	Mandatory under certain conditions, as noted in the remark descriptions.
DBREF	Optional	Mandatory for each peptide chain with a length greater than ten (10) residues, and for nucleic acid entries that exist in the Nucleic Acid Database (NDB).
SEQADV	Optional	Mandatory if sequence conflict exists.
SEQRES	Optional	Mandatory if ATOM records exist.

MODRES	Optional	Mandatory if modified group exists within the coordinates.
HET	Optional	Mandatory if non-standard group Other than water appears in the entry.
HETNAM	Optional	Mandatory if non-standard group Other than water appears in the entry.
HETSYN	Optional	
FORMUL	Optional	Mandatory if non-standard group Or water appears.
HELIX	Optional	
SHEET	Optional	
TURN	Optional	
SSBOND	Optional	Mandatory if disulfide bond is present.
LINK	Optional	
HYDBND	Optional	
SLTBRG	Optional	
CISPEP	Optional	
SITE	Optional	
CRYST1	Mandatory	
ORIGX1 ORIGX2 ORIGX3	Mandatory	
SCALE1 SCALE2 SCALE3	Mandatory	
MTRIX1 MTRIX2 MTRIX3	Optional	Mandatory if the complete Asymmetric unit must be generated from the given coordinates using non-crystallographic symmetry.
TVECT	Optional	
MODEL	Optional	Mandatory if more than one model is present in the entry.
ATOM	Optional	Mandatory if standard residues exist.
SIGATM	Optional	
ANISOU	Optional	
SIGUIJ	Optional	
TER	Optional	Mandatory if ATOM records exist.
HETATM	Optional	Mandatory if non-standard group appears.
ENDMDL	Optional	Mandatory if MODEL appears.
CONNECT	Optional	Mandatory if non-standard group appears.

MASTER Mandatory
 END Mandatory

Note that a PDB file existing outside of the PDB official release may contain locally-defined records beginning with "USER". The PDB reserves the right to add new record types (not beginning with "USER"), so programs which read PDB entries should be prepared to read (and ignore) other record types. PDB will follow standard procedures whenever format changes are proposed.

Sections of an Entry

The following table lists the various sections of a PDB coordinate entry and the records comprising them:

SECTION	DESCRIPTION	RECORD TYPE
Title	Summary descriptive remarks	HEADER, OBSLTE, TITLE, CAVEAT, COMPND, SOURCE, KEYWDS, EXPDTA, AUTHOR, REVDAT, SPRSDE, JRNL
Remark	Bibliography, refinement, annotations	REMARKs 1, 2, 3 and others
Primary structure	Peptide and/or nucleotide sequence and the relationship between the PDB sequence and that found in the sequence database(s)	DBREF, SEQADV, SEQRES MODRES
Heterogen	Description of non-standard groups	HET, HETNAM, HETSYN, FORMUL
Secondary structure	Description of secondary structure	HELIX, SHEET, TURN
Connectivity annotation	Chemical connectivity	SSBOND, LINK, HYDBND, SLTBRG, CISPEP
Miscellaneous features	Features within the macromolecule	SITE
Crystallographic	Description of the crystallographic cell	CRYST1
Coordinate transformation	Coordinate transformation operators	ORIGXn, SCALEn, MTRIXn, TVECT
Coordinate	Atomic coordinate data	MODEL, ATOM, SIGATM, ANISOU,

SIGUIJ, TER, HETATM, ENDMDL

Connectivity	Chemical connectivity	CONNECT
Bookkeeping	Summary information, end-of-file marker	MASTER, END

Field Formats

Each record type is presented in a table which contains the division of the records into fields by column number, defined data type, field name or a quoted string which must appear in the field, and field definition. Any column not specified must be left blank. Each field contains an identified data type which can be validated by a program. These are:

DATA TYPE	DESCRIPTION
-----------	-------------

AChar	An alphabetic character (A-Z, a-z).
Atom	Atom name which follow the naming rules in Appendix 3.
Character	Any non-control character in the ASCII character set or a space.
Continuation	A two-character field that is either blank (for the first record of a set) or contains a two digit number right-justified and blank-filled which counts continuation records starting with 2. The continuation number must be followed by a blank.
Date	A 9 character string in the form dd-mmm-yy where DD is the day of the month, zero-filled on the left (e.g., 04); MMM is the common English 3-letter abbreviation of the month; and YY is a year in the 20th century. This must represent a valid date.
IDcode	A PDB identification code which consists of 4 characters, the first of which is a digit in the range 0 - 9; the remaining 3 are alpha-numeric, and letters are upper case only. Entries with a 0 as the first character do not contain coordinate data.
Integer	Right-justified blank-filled integer value.
Token	A sequence of non-space characters followed by a colon and a space.
List	A String that is composed of text separated with commas.
LString	A literal string of characters. All spacing is significant and must be preserved.

LString(n)	An LString with exactly n characters.
Real(n,m)	Real (floating point) number in the FORTRAN format Fn.m.
Record name	The name of the record: 6 characters, left-justified and blank-filled.
Residue name	One of the standard amino acid or nucleic acids, as listed below, or the non-standard group designation as defined in the HET dictionary. Field is right-justified.
SList	A String that is composed of text separated with semi-colons.
Specification	A String composed of a token and its associated value separated by a colon.
Specification list	A sequence of Specifications, separated by semi-colons.
String	A sequence of characters. These characters may have arbitrary spacing, but should be interpreted as directed below.
String(n)	A String with exactly n characters.
SymOP	An integer field of from 4 to 6 digits, right-justified, of the form nnnMMM where nnn is the symmetry operator number and MMM is the translation vector.

To interpret a String, concatenate the contents of all continued fields together, collapse all sequences of multiple blanks to a single blank, and remove any leading and trailing blanks. This permits very long strings to be properly reconstructed. The above information about field formats is repeated as Appendix 6.

Residue Names

Standard residue names used in PDB entries:

RESIDUE TYPE	RESIDUE NAME

Amino acids	ALA, ARG, ASN, ASP, CYS, GLN, GLU, GLY, HIS, ILE, LEU, LYS, MET, PHE, PRO, SER, THR, TRP, TYR, VAL, ASX, GLX
Nucleic acids	A, C, G, T, U, I, +A, +C, +G, +T, +U, +I
Other	UNK (unknown)

Title Section

This section contains records used to describe the experiment and the biological macromolecules present in the entry: HEADER, OBSLTE, TITLE, CAVEAT, COMPND, SOURCE, KEYWDS, EXPDTA, AUTHOR, REVDAT, SPRSDE, JRNL, and REMARK records.

HEADER

Overview

The HEADER record uniquely identifies a PDB entry through the idCode field. This record also provides a classification for the entry. Finally, it contains the date the coordinates were deposited at the PDB.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	"HEADER"	
11 - 50	String(40)	classification	Classifies the molecule(s)
51 - 59	Date	depDate	Deposition date. This is the date the coordinates were received by the PDB
63 - 66	IDcode	idCode	This identifier is unique within PDB

Details

* The classification string is left-justified and exactly matches one of a collection of strings. See the class list available from the WWW site. In the case of macromolecular complexes, the classification field must present a class for each macromolecule present. Due to the limited length of the classification field, strings must sometimes be abbreviated. In these cases, the full terms are given in KEYWDS.

* Classification may be based on function, metabolic role, molecule type, cellular location, etc. In the case of a molecule having a dual function, both may be presented here.

OBSLTE**Overview**

OBSLTE appears in entries which have been withdrawn from distribution. This record acts as a flag in an entry which has been withdrawn from the PDB's full release. It indicates which, if any, new entries have replaced the withdrawn entry. The format allows for the case of multiple new entries replacing one existing entry.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	"OBSLTE"	
9 - 10	Continuation	continuation	Allows concatenation of multiple records.
12 - 20	Date	repDate	Date that this entry was replaced.
22 - 25	IDcode	idCode	ID code of this entry.
32 - 35	IDcode	rIdCode	ID code of entry that replaced this one.
37 - 40	IDcode	rIdCode	ID code of entry that replaced this one.
42 - 45	IDcode	rIdCode	ID code of entry that replaced this one.
47 - 50	IDcode	rIdCode	ID code of entry that replaced this one.

Details

- It is PDB policy that only the primary author who submitted an entry has the authority to withdraw it. All withdrawn entries are available for research purposes. PDB should be contacted in cases where the withdrawn data are desired.
-

TITLE**Overview**

The TITLE record contains a title for the experiment or analysis that is represented in the entry. It should identify an entry in the PDB in the same way that a title identifies a paper.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION

1 - 6	Record name	"TITLE "	
9 - 10	Continuation	continuation	Allows concatenation of multiple records.
11 - 70	String	title	Title of the experiment.

CAVEAT**Overview**

CAVEAT warns of severe errors in an entry. Use caution when using an entry containing this record.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION

1 - 6	Record name	"CAVEAT"	
9 - 10	Continuation	continuation	Allows concatenation of multiple records.
12 - 15	IDcode	idCode	PDB ID code of this entry.
20 - 70	String	comment	Free text giving the reason for the CAVEAT.

COMPND**Overview**

The COMPND record describes the macromolecular contents of an entry. Each macromolecule found in the entry is described by a set of token: value pairs, and is referred to as a COMPND record component. Since the concept of a molecule is difficult to specify exactly, PDB staff may exercise editorial judgment in consultation with depositors in assigning these names.

For each macromolecular component, the molecule name, synonyms, number assigned by the Enzyme Commission (EC), and other relevant details are specified.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	"COMPND"	
9 - 10	Continuation	continuation	Allows concatenation of multiple records.
11 - 70	Specification list	compound	Description of the molecular components.

SOURCE**Overview**

The SOURCE record specifies the biological and/or chemical source of each biological molecule in the entry. Sources are described by both the common name and the scientific name, e.g., genus and species. Strain and/or cell-line for immortalized cells are given when they help to uniquely identify the biological entity studied.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	"SOURCE"	
9 - 10	Continuation	continuation	Allows concatenation of multiple records.
11 - 70	Specification list	srcName	Identifies the source of the macromolecule in a token: value format.

KEYWDS**Overview**

The KEYWDS record contains a set of terms relevant to the entry. Terms in the KEYWDS record provide a simple means of categorizing entries and may be used to generate index files. This record addresses some of the limitations found in the classification field of the HEADER record. It provides the opportunity to add further annotation to the entry in a concise and computer-searchable fashion.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	"KEYWDS"	
9 - 10	Continuation	continuation	Allows concatenation of records if necessary.
11 - 70	List	keywds	Comma-separated list of keywords relevant to the entry.

EXPDTA**Overview**

The EXPDTA record presents information about the experiment.

The EXPDTA record identifies the experimental technique used. This may refer to the type of radiation and sample, or include the spectroscopic or modeling technique. Permitted values include:

ELECTRON DIFFRACTION
 FIBER DIFFRACTION
 FLUORESCENCE TRANSFER
 NEUTRON DIFFRACTION
 NMR
 THEORETICAL MODEL
 X-RAY DIFFRACTION

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	"EXPDTA"	
9 - 10	Continuation	continuation	Allows concatenation of multiple records.
11 - 70	SList	technique	The experimental technique(s) with optional comment describing the sample or experiment.

AUTHOR**Overview**

The AUTHOR record contains the names of the people responsible for the contents of the entry.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	"AUTHOR"	
9 - 10	Continuation	continuation	Allows concatenation of multiple records.
11 - 70	List	authorList	List of the author names, separated by commas.

REVDAT**Overview**

REVDAT records contain a history of the modifications made to an entry since its release.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	"REVDAT"	
8 - 10	Integer	modNum	Modification number.
11 - 12	Continuation	continuation	Allows concatenation of multiple records.
14 - 22	Date	modDate	Date of modification (or release for new entries). This is not repeated on continuation lines.

24 - 28	String(5)	modId	Identifies this particular modification. It links to the archive used internally by PDB. This is not repeated on continuation lines.
32	Integer	modType	An integer identifying the type of modification. In case of revisions with more than one possible modType, the highest value applicable will be assigned.

SPRSDE**Overview**

The SPRSDE records contain a list of the ID codes of entries that were made obsolete by the given coordinate entry and withdrawn from the PDB release set. One entry may replace many. It is PDB policy that only the principal investigator of a structure has the authority to withdraw it.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	"SPRSDE"	
9 - 10	Continuation	continuation	Allows for multiple ID codes.
12 - 20	Date	sprsdeDate	Date this entry superseded the listed entries. This field is not copied on continuations.
22 - 25	IDcode	idCode	ID code of this entry. This field is not copied on continuations.
32 - 35	IDcode	sIdCode	ID code of a superseded entry.
37 - 40	IDcode	sIdCode	ID code of a superseded entry.
42 - 45	IDcode	sIdCode	ID code of a superseded entry.
47 - 50	IDcode	sIdCode	ID code of a superseded entry.

JRNL**Overview**

The JRNL record contains the primary literature citation that describes the experiment which resulted in the deposited coordinate set. There is at most one JRNL reference per entry. If there is no primary reference, then there is no JRNL reference. Other references are given in REMARK 1.

PDB is in the process of linking and/or adding all references to CitDB, the literature database used by the Genome Data Base (available at URL <http://gdbwww.gdb.org/gdb-bin/genera/genera/citation/Citation>).

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	"JRNL "	

13 - 70 LString text See Details below.

REMARK

Overview

REMARK records present experimental details, annotations, comments, and information not included in other records. In a number of cases, REMARKs are used to expand the contents of other record types. A new level of structure is being used for some REMARK records. This is expected to facilitate searching and will assist in the conversion to a relational database.

The very first line of every set of REMARK records is used as a spacer to aid in reading.

COLUMNS	DATA TYPE	FIELD	DEFINITION
---------	-----------	-------	------------

1 - 6	Record name	"REMARK"	
8 - 10	Integer	remarkNum	Remark number. It is not an error for remark n to exist in an entry when remark n-1 does not.
12 - 70	LString	empty	Left as white space in first line of each new remark.

Primary Structure Section

The primary structure section of a PDB file contains the sequence of residues in each chain of the macromolecule. Embedded in these records are chain identifiers and sequence numbers that allow other records to link into the sequence.

DBREF

Overview

The DBREF record provides cross-reference links between PDB sequences and the corresponding database entry or entries. A cross reference to the sequence database is mandatory for each peptide chain with a length greater than ten (10) residues. For nucleic acid entries a DBREF record pointing to the Nucleic Acid Database (NDB) is mandatory when the corresponding entry exists in NDB.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION
---------	-----------	-------	------------

1 - 6	Record name	"DBREF "	
8 - 11	IDcode	idCode	ID code of this entry.
13	Character	chainID	Chain identifier.
15 - 18	Integer	seqBegin	Initial sequence number of the PDB sequence segment.
19	AChar	insertBegin	Initial insertion code of the PDB sequence segment.
21 - 24	Integer	seqEnd	Ending sequence number of the PDB sequence segment.
25	AChar	insertEnd	Ending insertion code of the PDB sequence segment.

27 - 32	LString	database	Sequence database name. "PDB" when a corresponding sequence database entry has not been identified.
34 - 41	LString	dbAccession	Sequence database accession code. For GenBank entries, this is the NCBI gi number.
43 - 54	LString	dbIdCode	Sequence database identification code. For GenBank entries, this is the accession code.
56 - 60	Integer	dbseqBegin	Initial sequence number of the database segment.
61	AChar	idbnsBeg	Insertion code of initial residue of the segment, if PDB is the reference.
63 - 67	Integer	dbseqEnd	Ending sequence number of the database segment.

BioMagResBank	BMRB
BLOCKS	BLOCKS
European Molecular Biology Laboratory	EMBL
GenBank	GB
Genome Data Base	GDB
Nucleic Acid Database	NDB
PROSITE	PROSIT
Protein Data Bank	PDB
Protein Identification Resource	PIR
SWISS-PROT	SWS
TREMBL	TREMBL

SEQADV

Overview

The SEQADV record identifies conflicts between sequence information in the ATOM records of the PDB entry and the sequence database entry given on DBREF. Please note that these records were designed to identify differences and not errors. No assumption is made as to which database contains the correct data. PDB may include REMARK records in the entry that reflect the depositor's view of which database has the correct sequence.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	"SEQADV"	
8 - 11	IDcode	idCode	ID code of this entry.
13 - 15	Residue name	resName	Name of the PDB residue in conflict.
17	Character	chainID	PDB chain identifier.
19 - 22	Integer	seqNum	PDB sequence number.

23	AChar	iCode	PDB insertion code.
25 - 28	LString	database	Sequence database name.
30 - 38	LString	dbIdCode	Sequence database accession number.
40 - 42	Residue name	dbRes	Sequence database residue name.
44 - 48	Integer	dbSeq	Sequence database sequence number.

SEQRES

Overview

SEQRES records contain the amino acid or nucleic acid sequence of residues in each chain of the macromolecule that was studied.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	"SEQRES"	
9 - 10	Integer	serNum	Serial number of the SEQRES record for the current chain. Starts at 1 and increments by one each line. Reset to 1 for each chain.
12	Character	chainID	Chain identifier. This may be any single legal character, including a blank which is used if there is only one chain.
14 - 17	Integer	numRes	Number of residues in the chain. This value is repeated on every record.
20 - 22	Residue name	resName	Residue name.

MODRES

Overview

The MODRES record provides descriptions of modifications (e.g., chemical or post-translational) to protein and nucleic acid residues. Included are a mapping between residue names given in a PDB entry and standard residues.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	"MODRES"	
8 - 11	IDcode	idCode	ID code of this entry.
13 - 15	Residue name	resName	Residue name used in this entry.
17	Character	chainID	Chain identifier.
19 - 22	Integer	seqNum	Sequence number.
23	AChar	iCode	Insertion code.
25 - 27	Residue name	stdRes	Standard residue name.
30 - 70	String	comment	Description of the residue modification.

Secondary Structure Section

The secondary structure section of a PDB file describes helices, sheets, and turns found in protein and polypeptide structures.

HELIX

Overview

HELIX records are used to identify the position of helices in the molecule. Helices are both named and numbered. The residues where the helix begins and ends are noted, as well as the total length.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	"HELIX "	
8 - 10	Integer	serNum	Serial number of the helix. This starts at 1 and increases incrementally.
12 - 14	LString(3)	helixID	Helix identifier. In addition to a serial number, each helix is given an alphanumeric character helix identifier.
16 - 18	Residue name	initResName	Name of the initial residue.
20	Character	initChainID	Chain identifier for the chain containing this helix.
22 - 25	Integer	initSeqNum	Sequence number of the initial residue.
26	AChar	initIcode	Insertion code of the initial residue.
28 - 30	Residue name	endResName	Name of the terminal residue of the helix.
32	Character	endChainID	Chain identifier for the chain containing this helix.
34 - 37	Integer	endSeqNum	Sequence number of the terminal residue.
38	AChar	endIcode	Insertion code of the terminal residue.
39 - 40	Integer	helixClass	Helix class (see below).
41 - 70	String	comment	Comment about this helix.
72 - 76	Integer	length	Length of this helix.

SHEET

Overview

SHEET records are used to identify the position of sheets in the molecule. Sheets are both named and numbered. The residues where the sheet begins and ends are noted.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	"SHEET "	
8 - 10	Integer	strand	Strand number which starts at 1 for each strand within a sheet and increases by one.
12 - 14	LString(3)	sheetID	Sheet identifier.
15 - 16	Integer	numStrands	Number of strands in sheet.
18 - 20	Residue name	initResName	Residue name of initial residue.
22	Character	initChainID	Chain identifier of initial residue in strand.
23 - 26	Integer	initSeqNum	Sequence number of initial residue in strand.
27	AChar	initICode	Insertion code of initial residue in strand.
29 - 31	Residue name	endResName	Residue name of terminal residue.
33	Character	endChainID	Chain identifier of terminal residue.
34 - 37	Integer	endSeqNum	Sequence number of terminal residue.
38	AChar	endICode	Insertion code of terminal residue.
39 - 40	Integer	sense	Sense of strand with respect to previous strand in the sheet. 0 if first strand, 1 if parallel, -1 if anti-parallel.
42 - 45	Atom	curAtom	Registration. Atom name in current strand.
46 - 48	Residue name	curResName	Registration. Residue name in current strand.
50	Character	curChainId	Registration. Chain identifier in current strand.

TURN**Overview**

The TURN records identify turns and other short loop turns which normally connect other secondary structure segments.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	"TURN "	
8 - 10	Integer	seq	Turn number; starts with 1 and increments by one.
12 - 14	LString(3)	turnId	Turn identifier
16 - 18	Residue name	initResName	Residue name of initial residue in turn.
20	Character	initChainId	Chain identifier for the chain containing this turn.

21 - 24	Integer	initSeqNum	Sequence number of initial residue in turn.
25	AChar	initICode	Insertion code of initial residue in turn.
27 - 29	Residue name	endResName	Residue name of terminal residue of turn.
31	Character	endChainId	Chain identifier for the chain containing this turn.
32 - 35	Integer	endSeqNum	Sequence number of terminal residue of turn.
36	AChar	endICode	Insertion code of terminal residue of turn.
41 - 70	String	comment	Associated comment.

Miscellaneous Features Section

The miscellaneous features section describes features in the molecule such as the active site. Other features may be described in the remarks section but are not given a specific record type so far.

SITE

Overview

The SITE records supply the identification of groups comprising important sites in the macromolecule.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION

1 - 6	Record name	"SITE "	
8 - 10	Integer	seqNum	Sequence number.
12 - 14	LString(3)	siteID	Site name.
16 - 17	Integer	numRes	Number of residues comprising site.
19 - 21	Residue name	resName1	Residue name for first residue comprising site.
23	Character	chainID1	Chain identifier for first residue comprising site.
24 - 27	Integer	seq1	Residue sequence number for first residue comprising site.
28	AChar	iCode1	Insertion code for first residue comprising site.
30 - 32	Residue name	resName2	Residue name for second residue comprising site.
34	Character	chainID2	Chain identifier for second residue comprising site.
35 - 38	Integer	seq2	Residue sequence number for second residue comprising site.

39	AChar	iCode2	Insertion code for second residue comprising site.
41 - 43	Residue name	resName3	Residue name for third residue comprising site.
45	Character	chainID3	Chain identifier for third residue comprising site.
46 - 49	Integer	seq3	Residue sequence number for third residue comprising site.
50	AChar	iCode3	Insertion code for third residue comprising site.
52 - 54	Residue name	resName4	Residue name for fourth residue comprising site.
56	Character	chainID4	Chain identifier for fourth residue comprising site.
57 - 60	Integer	seq4	Residue sequence number for fourth residue comprising site.
61	AChar	iCode4	Insertion code for fourth residue comprising site.

Coordinate Section

The Coordinate Section contains the collection of atomic coordinates as well as the MODEL and ENDMDL records.

MODEL

Overview

The MODEL record specifies the model serial number when multiple structures are presented in a single coordinate entry, as is often the case with structures determined by NMR.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION
---------	-----------	-------	------------

1 - 6	Record name	"MODEL "	
-------	-------------	----------	--

11 - 14	Integer	serial	Model serial number.
---------	---------	--------	----------------------

ATOM

Overview

The ATOM records present the atomic coordinates for standard residues. They also present the occupancy and temperature factor for each atom. Heterogen coordinates use the HETATM record type. The element symbol is always present on each ATOM record; segment identifier and charge are optional.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION
---------	-----------	-------	------------

1 - 6	Record name	"ATOM "	
-------	-------------	---------	--

7 - 11	Integer	serial	Atom serial number.
--------	---------	--------	---------------------

13 - 16	Atom	name	Atom name.
17	Character	altLoc	Alternate location indicator.
18 - 20	Residue name	resName	Residue name.
22	Character	chainID	Chain identifier.
23 - 26	Integer	resSeq	Residue sequence number.
27	AChar	iCode	Code for insertion of residues.
31 - 38	Real(8.3)	x	Orthogonal coordinates for X in Angstroms.
39 - 46	Real(8.3)	y	Orthogonal coordinates for Y in Angstroms.
47 - 54	Real(8.3)	z	Orthogonal coordinates for Z in Angstroms.
55 - 60	Real(6.2)	occupancy	Occupancy.
61 - 66	Real(6.2)	tempFactor	Temperature factor.
73 - 76	LString(4)	segID	Segment identifier, left-justified.
77 - 78	LString(2)	element	Element symbol, right-justified.
79 - 80	LString(2)	charge	Charge on the atom.

SIGATM**Overview**

The SIGATM records present the standard deviation of atomic parameters as they appear in ATOM and HETATM records.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION

1 - 6	Record name	"SIGATM"	
7 - 11	Integer	serial	Atom serial number.
13 - 16	Atom	name	Atom name.
17	Character	altLoc	Alternate location indicator.
18 - 20	Residue name	resName	Residue name.
22	Character	chainID	Chain identifier.
23 - 26	Integer	resSeq	Residue sequence number.
27	AChar	iCode	Insertion code.
31 - 38	Real(8.3)	sigX	Standard deviations of the stored coordinates (Angstroms).
39 - 46	Real(8.3)	sigY	Standard deviations of the stored coordinates (Angstroms).
47 - 54	Real(8.3)	sigZ	Standard deviations of the stored coordinates (Angstroms).
55 - 60	Real(6.2)	sigOcc	Standard deviation of occupancy.
61 - 66	Real(6.2)	sigTemp	Standard deviation of temperature factor.
73 - 76	LString(4)	segID	Segment identifier, left-justified.
77 - 78	LString(2)	element	Element symbol, right-justified.
79 - 80	LString(2)	charge	Charge on the atom.

ANISOU**Overview**

The ANISOU records present the anisotropic temperature factors.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	"ANISOU"	
7 - 11	Integer	serial	Atom serial number.
13 - 16	Atom	name	Atom name.
17	Character	altLoc	Alternate location indicator.
18 - 20	Residue name	resName	Residue name.
22	Character	chainID	Chain identifier.
23 - 26	Integer	resSeq	Residue sequence number.
27	AChar	iCode	Insertion code.

SIGUIJ**Overview**

The SIGUIJ records present the standard deviations of anisotropic temperature factors scaled by a factor of 10^{**4} (Angstroms**2).

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	"SIGUIJ"	
7 - 11	Integer	serial	Atom serial number.
13 - 16	Atom	name	Atom name.
17	Character	altLoc	Alternate location indicator.
18 - 20	Residue name	resName	Residue name.
22	Character	chainID	Chain identifier.
23 - 26	Integer	resSeq	Residue sequence number.
27	AChar	iCode	Insertion code.

TER**Overview**

The TER record indicates the end of a list of ATOM/HETATM records for a chain.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	"TER "	
7 - 11	Integer	serial	Serial number.
18 - 20	Residue name	resName	Residue name.
22	Character	chainID	Chain identifier.
23 - 26	Integer	resSeq	Residue sequence number.
27	AChar	iCode	Insertion code.

HETATM

Overview The HETATM records present the atomic coordinate records for atoms within "non-standard" groups. These records are used for water molecules and atoms presented in HET groups.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	"HETATM"	
7 - 11	Integer	serial	Atom serial number.
13 - 16	Atom	name	Atom name.
17	Character	altLoc	Alternate location indicator.
18 - 20	Residue name	resName	Residue name.
22	Character	chainID	Chain identifier.
23 - 26	Integer	resSeq	Residue sequence number.
27	AChar	iCode	Code for insertion of residues.
31 - 38	Real(8.3)	x	Orthogonal coordinates for X.
39 - 46	Real(8.3)	y	Orthogonal coordinates for Y.
47 - 54	Real(8.3)	z	Orthogonal coordinates for Z.
55 - 60	Real(6.2)	occupancy	Occupancy.
61 - 66	Real(6.2)	tempFactor	Temperature factor.
73 - 76	LString(4)	segID	Segment identifier; left-justified.

ENDMDL**Overview**

The ENDMDL records are paired with MODEL records to group individual structures found in a coordinate entry.

Record Format

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	"ENDMDL"	

Summary

The Protein Data Bank (PDB) is the central worldwide repository for three-dimensional (3D) structure data of biological macromolecules. The Research Collaboratory for Structural Bioinformatics (RCSB) has completely redesigned its resource for the distribution and query of 3D structure data.

For the last 30 years the community of protein biologists has catalogued new structures in the Protein Data Bank, a freely accessible database. The PDB even includes the structures of complex entities, like entire viruses and the ribosome. Plans are underway to fill out the database with enough unique structures to provide a catalog of practically all possible protein formations.

Model questions

- What is PDB and how did it originate?
- What are the various features of PDB?
- What are the various types of records in PDB?

Reference:

<http://www.rcsb.org/pdb>

http://www.rcsb.org/pdb/docs/format/pdbguide2.2/guide2.2_frame.html

Lesson 3.4.1

PROTEINS-PRIMARY STRUCTURE

Objective

- 3.4.1.1 Introduction
- 3.4.1.2 Properties of Protein
- 3.4.1.3 Working with proteins
- 3.4.1.4 Protein regulation
- 3.4.1.5 Primary Structure of Proteins
- 3.4.1.6 Amino-Terminal Sequence Determination
- 3.4.1.7 Protease Digestion
- 3.4.1.8 Carboxy-Terminal Sequence Determination
- 3.4.1.9 Summary
- 3.4.1.10 Model Questions
- 3.4.1.11 References

Objective

The lesson gives a general account on proteins structure and properties .It also deals with the charactets of primary structure and sequencing of peptides.

3.4.1.1 Introduction

A **protein** (in Greek $\pi\rho\omega\tau\varepsilon\iota\nu\eta$ = *first thread*) is a complex, high-molecular-weight organic compound that consists of amino acids joined by peptide bonds. Proteins are essential to the structure and function of all living cells and viruses. Many proteins are enzymes or subunits of enzymes. Other proteins play structural or mechanical roles, such as those that form the struts and joints of the cytoskeleton, serving as biological scaffolds for the mechanical integrity and tissue signalling functions. Still more functions filled by proteins include immune response and the storage and transport of various ligands. In nutrition, proteins serve as the source of amino acids for organisms that do not synthesize those amino acids natively.

Proteins are one of the classes of bio-macromolecules, alongside polysaccharides, lipids, and nucleic acids, that make up the primary constituents of living things. They are among the most actively-studied molecules in biochemistry, and were discovered by Jöns Jakob Berzelius in 1838.

Almost all natural proteins are encoded by DNA. DNA is transcribed to yield RNA, which serves as a template for translation by ribosomes.

3.4.1.2 Properties of Protein

a)Structure

Proteins are amino acid chains that fold into unique 3-dimensional structures. The shape into which a protein naturally folds is known as its native state, which is determined by its sequence of amino acids. Thus, proteins are their own polymers, with amino acids being the monomers. Biochemists refer to four distinct aspects of a protein's structure:

- Primary structure: the amino acid sequence

- Secondary structure: highly patterned sub-structures—alpha helix and beta sheet—or segments of chain that assume no stable shape. Secondary structures are locally defined, meaning that there can be many different secondary motifs present in one single protein molecule.
- Tertiary structure: the overall shape of a single protein molecule; the spatial relationship of the secondary structural motifs to one another
- Quaternary structure: the shape or structure that results from the union of more than one protein molecule, usually called subunit proteins subunits in this context, which function as part of the larger assembly or protein complex.

In addition to these levels of structure, proteins may shift between several similar structures in performing their biological function. In the context of these functional rearrangements, these tertiary or quaternary structures are usually referred to as "conformations," and transitions between them are called **conformational changes**. Proteins are separated into two groups: Complete and Incomplete. Incomplete proteins are from plants and do not include all 20 amino acids. Complete proteins come from an animal and include all 20 amino acids. You get protein from mostly everything you eat, but whether all the amino acids are in them depends on what the substance is.

The primary structure is held together by covalent peptide bonds, which are made during the process of translation. The secondary structures are held together by hydrogen bonds. The tertiary structure is held together primarily by hydrophobic interactions but hydrogen bonds, ionic interactions, and disulfide bonds are usually involved too.

The process by which the higher structures form is called protein folding and is a consequence of the primary structure. The mechanism of protein folding is not entirely understood. Although any unique polypeptide may have more than one stable folded conformation, each conformation has its own biological activity and only one conformation is considered to be the active, or native conformation.

The two ends of the amino acid chain are referred to as the carboxy terminus (C-terminus) and the amino terminus (N-terminus) based on the nature of the free group on each extremity.

3.4.1.3 Working with proteins

Proteins are sensitive to their environment. They may only be active in their native state, over a small pH range, and under solution conditions with a minimum quantity of electrolytes. A protein in its native state is often described as folded. A protein that is not in its native state is said to be denatured. Denatured proteins generally have no well-defined secondary structure. Many proteins denature and will not remain in solution in distilled water.

One of the more striking discoveries of the 20th century was that the native and denatured states in many proteins were interconvertible, that by careful control of solution conditions (by for example, dialyzing away a denaturing chemical), a denatured protein could be converted to native form. The issue of how proteins arrive at their native state is an important area of biochemical study, called the study of protein folding.

Through genetic engineering, researchers can alter the sequence and hence the structure, "targeting", susceptibility to regulation and other properties of a protein. The genetic sequences of different proteins may be spliced together to create "chimeric" proteins that possess properties of both. This form of tinkering represents one of the chief tools of cell and molecular biologists to change and to probe the workings of cells.

Another area of protein research attempts to engineer proteins with entirely new properties or functions, a field known as protein engineering. Protein-protein interactions can be screened for using two-hybrid screening.

3.4.1.4 Protein regulation

Various molecules and ions are able to bind to specific sites on proteins. These sites are called binding sites. They exhibit chemical specificity. The particle that binds is called a ligand. The strength of ligand-protein binding is a property of the binding site known as affinity.

Since proteins are involved in practically every function performed by a cell, the mechanisms for controlling these functions therefore depend on controlling protein activity. Regulation can involve a protein's shape or concentration. Some forms of regulation include:

- Allosteric modulation: When the binding of a ligand at one site on a protein affects the binding of ligand at another site.
- Covalent modulation: When the covalent modification of a protein affects the binding of a ligand or some other aspect of the protein's function.

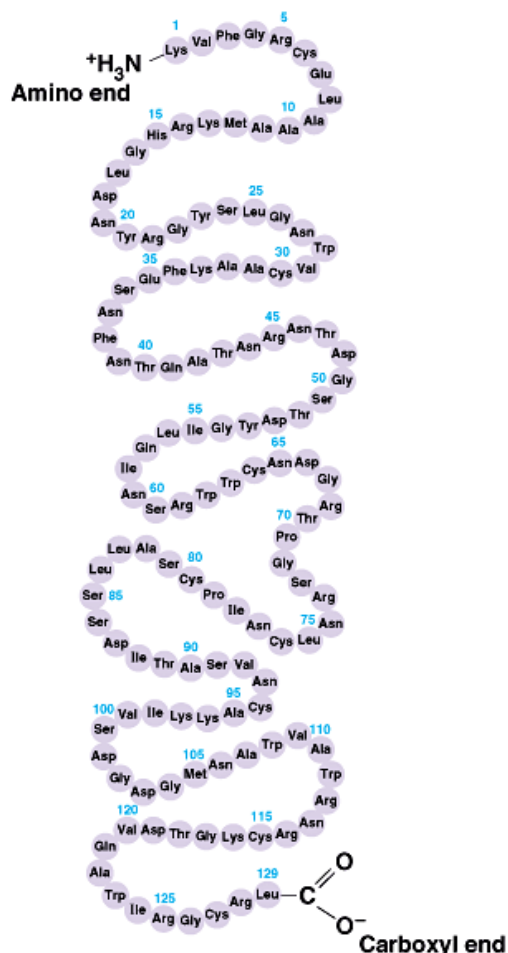
3.4.1.5 Primary Structure of Proteins

As described earlier proteins are the polymers containing large number of amino acids joined to each other by peptide bonds. For establishing the structure of a protein molecule, we will have to answer the following questions.

- (1) The nature of amino acids.
- (2) The number of each particular amino acid present in one molecule of the protein.
- (3) The sequence in which the various different amino acids are arranged in the molecule.
- (4) The shape of the peptide chain, i.e. Whether it is linear, cyclic, branched or arranged in the form of helix.
- (5) The forces with which the individual peptide chains are held together.
- (6) The way in which the individual peptide chains are arranged in definite manner in a macromolecule of an individual shape (folded, refolded).
- (7) The number of peptide chains and their arrangement in the natural protein.

The first three points constitute the primary structure, the point 4 constitutes the secondary structure, the point 5 and 6 constitute the tertiary structure; and 7 constitutes the quaternary structure of the protein molecule. The secondary or higher structure of proteins can accurately be determined only by X-ray analysis; although other physical methods like viscosity, light scattering, rotatory dispersion etc. also provide useful information.

The primary structure of peptides and proteins refers to the linear number and order of the amino acids present. The convention for the designation of the order of amino acids is that the N-terminal end (i.e. the end bearing the residue with the free α -amino group) is to the left (and the number 1 amino acid) and the C-terminal end (i.e. the end with the residue containing a free α -carboxyl group) is to the right.



Copyright © Pearson Education, Inc., publishing as Benjamin Cummings.

3.4.1.6 Amino-Terminal Sequence Determination

Prior to sequencing peptides it is necessary to eliminate disulfide bonds within peptides and between peptides. Several different chemical reactions can be used in order to permit separation of peptide strands and prevent protein conformations that are dependent upon disulfide bonds. The most common treatments are to use either **2-mercaptoethanol** or **dithiothreitol**. Both of these chemicals reduce disulfide bonds. To prevent reformation of the disulfide bonds the peptides are treated with **iodoacetic acid** in order to alkylate the free sulfhydryls.

There are three major chemical techniques for sequencing peptides and proteins from the N-terminus. These are the **Sanger**, **Dansyl chloride** and **Edman techniques**.

Sanger's Reagent: This sequencing technique utilizes the compound, 2,4-dinitrofluorobenzene (DNF) which reacts with the N-terminal residue under alkaline conditions. The derivatized amino acid can be hydrolyzed and will be labeled with a dinitrobenzene group that imparts a yellow color to the amino acid. Separation of the modified amino acids (DNP-derivative) by electrophoresis

and comparison with the migration of DNP-derivative standards allows for the identification of the N-terminal amino acid.

Dansyl chloride: Like DNF, dansyl chloride reacts with the N-terminal residue under alkaline conditions. Analysis of the modified amino acids is carried out similarly to the Sanger method except that the dansylated amino acids are detected by fluorescence. This imparts a higher sensitivity into this technique over that of the Sanger method.

Edman degradation: The utility of the Edman degradation technique is that it allows for additional amino acid sequence to be obtained from the N-terminus inward. Using this method it is possible to obtain the entire sequence of peptides. This method utilizes phenylisothiocyanate to react with the N-terminal residue under alkaline conditions. The resultant phenylthiocarbamyl derivatized amino acid is hydrolyzed in anhydrous acid. The hydrolysis reaction results in a rearrangement of the released N-terminal residue to a phenylthiohydantoin derivative. As in the Sanger and Dansyl chloride methods, the N-terminal residue is tagged with an identifiable marker, however, the added advantage of the Edman process is that the remainder of the peptide is intact. The entire sequence of reactions can be repeated over and over to obtain the sequences of the peptide. This process has subsequently been automated to allow rapid and efficient sequencing of even extremely small quantities of peptide.

3.4.1.7 Protease Digestion

Due to the limitations of the Edman degradation technique, peptides longer than around 50 residues can not be sequenced completely. The ability to obtain peptides of this length, from proteins of greater length, is facilitated by the use of enzymes, endopeptidases, that cleave at specific sites within the primary sequence of proteins. The resultant smaller peptides can be chromatographically separated and subjected to Edman degradation sequencing reactions.

Specificities of Several Endoproteases

Enzyme	Source	Specificity	Additional Points
Trypsin	Bovine pancreas	peptide bond C-terminal to R, K, but not if next to P	highly specific for positively charged residues
Chymotrypsin	Bovine pancreas	peptide bond C-terminal to F, Y, W but not if next to P	prefers bulky hydrophobic residues, cleaves slowly at N, H, M, L

Elastase	Bovine pancreas	peptide bond C-terminal to A, G, S, V, but not if next to P	
Thermolysin	<i>Bacillus thermoproteolyticus</i>	peptide bond N-terminal to I, M, F, W, Y, V, but not if next to P	prefers small neutral residues, can cleave at A, D, H, T
Pepsin	Bovine gastric mucosa	peptide bond N-terminal to L, F, W, Y, but when next to P	exhibits little specificity, requires low pH
Endopeptidase V8	<i>Staphylococcus aureus</i>	peptide bond C-terminal to E	

3.4.1.8 Carboxy-Terminal Sequence Determination

No reliable chemical techniques exist for sequencing the C-terminal amino acid of peptides. However, there are enzymes, exopeptidases, that have been identified that cleave peptides at the C-terminal residue which can then be analyzed chromatographically and compared to standard amino acids. This class of exopeptidases are called, carboxypeptidases.

Specificities of Several Exopeptidases

Enzyme	Source	Specificity
Carboxypeptidase A	Bovine pancreas	Will not cleave when C-terminal residue = R, K or P or if P resides next to terminal residue
Carboxypeptidase B	Bovine pancreas	Cleaves when C-terminal residue = R or K; not when P resides next to terminal residue
Carboxypeptidase	Citrus	All free C-terminal residues, pH optimum

C	leaves	= 3.5
Carboxypeptidase Y	Yeast	All free C-terminal residues, slowly at G residues

3.4.1.9 Summary

- Proteins are the polymers of amino acids. The amino acids in protein are linked by peptide bond. Many enzymes and hormones nothing but proteins. Proteins are amino acid chains that fold into unique 3-dimensional structures. The shape into which a protein naturally folds is known as its native state, which is determined by its sequence of amino acids. According to biochemists protein structure can be studied under different levels viz Primary structure: the amino acid sequence. Secondary structure: highly patterned sub-structures—alpha helix and beta sheet—or segments of chain that assume no stable shape. Tertiary structure: the overall shape of a single protein molecule; Proteins are sensitive to their environment. They may only be active in their native state, over a small pH range, and under solution conditions with a minimum quantity of electrolytes

3.4.1.10 Model Questions

- 1) Explain the primary structure of proteins
- 2) Explain different methods of N-terminal sequences of peptides

3.4.1.11 References

- 1) Principles of biochemistry by Nelson and Cox
- 2) Biochemistry by O.P Agarwal

Author
P. Jaganmohan Rao

Lesson 3.4.2**Secondary, Tertiary and Quaternary Structure of proteins****Structure**

- 3.4.2.1 Introduction**
- 3.4.2.2 Types of secondary structures**
- 3.4.2.3 Tertiary Structure**
- 3.4.2.4 Quaternary structure**
- 3.4.2.5 Symmetry in proteins**
- 3.4.2.6 Protein Stability**
- 3.4.2.7 Summary**
- 3.4.2.8 Model questions**
- 3.4.2.9 Reference books**

Objective

This lesson explains the secondary level of organization of proteins

3.4.2.1 Introduction:

The Secondary Structure is defined as the local conformation of its back bone. For proteins this has come to mean the specification of regular polypeptide back bone foldings "helices, pleated sheets & turns.

3.4.2.2 Types of secondary structures:

Two types of basic structures of poly peptides have been recognized which are as follows;

Helices:

Helices are the most striking elements of protein secondary structure. If a polypeptide chain is twisted by the same amount of carbon atoms, it assumes a helical conformation. A helix may be characterized by the number, n . Of peptide units per helical turn, and its pitch, P .

P = the distance the helix rises along its axis per turn.

The helix has Chirality's: i .e. it may be right handed or left handed. The glue that holds polypeptide helices another 20 structures together is as Hydrogen bonds.

The α - Helix:

Only one helical Polypeptide conformation has simultaneously allowed conformation angles & a favorable Hydrogen bonding pattern : the α - helix, a particularly rigid arrangement of the Polypeptide chain. Its discovery through model building by pauling in 1951.

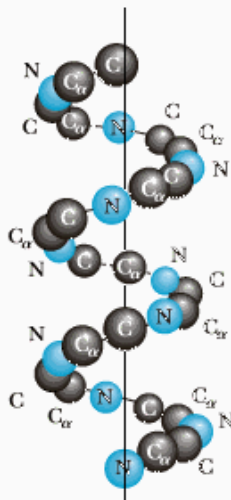
- Thus, all the main-chain Co and NN groups are 'H' bonded. Each residue is related to the next one by a rise of 1.5 \AA along the helix axis & a rotation of 100° , which gives 3.6 aminoacid residues per turn of helix. Thus, amino acids spaced three & four apart in the linear sequence are spatially quite close to one another in a helix.
- The pitch of the α -helix, which is equal to the product of the translation (1.5 \AA) & the number of residues per turn (3.6), is 5.4 \AA . The screw sense of a helix can be right-handed or left - handed the α - helices found in proteins are right handed.

- In α - helix structure along the 'H' bond vanderwaals forces are also present. These contacts across the helix, thereby maximizing their association energies.
Eg: 75% of Myoglobin & hemoglobin is α - helix. Single standard α - helices are usually less than 45 A⁰ long.
- The α - helix is a common secondary structural element of both fibrous & globular proteins.

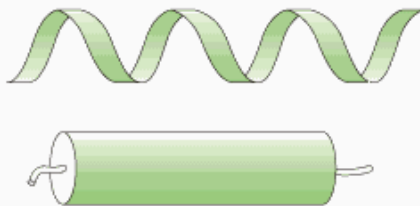
In globular proteins, α - helices have an average span of ~ 12 residues. Which corresponds to over 3 helical turns & a length of 18 A⁰. However, helices with as many as 53 residues have been found. The cytoskeleton (internal scaffolding) of cells is rich in intermediate filaments, which also are two-stranded α - helical coiled coils.

α - Helix

Only the N-C _{α} -C backbone is represented. The vertical line is the helix axis.

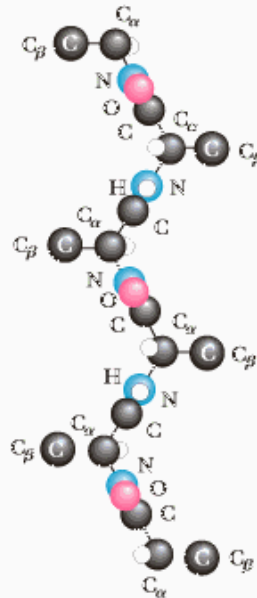


"Shorthand" α -helix

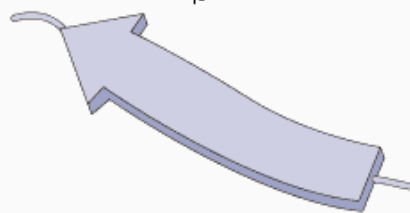


β - Strand

The N-C _{α} -C backbone as well as the C _{β} of R groups are represented here. Note that the amide planes are perpendicular to the image.



"Shorthand" β -strand



Copyright © 1999 by Harcourt Brace & Company

Other helices:

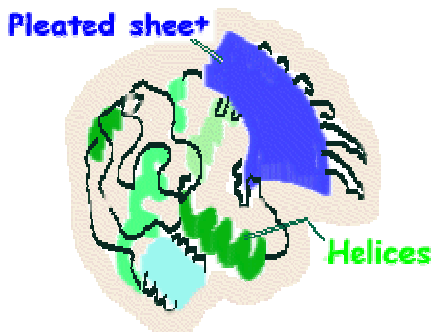
Along α - helices II helix also has a tightly forbidden conformation, has only rarely been observed & then only as segments of longer helices. It has wide & flat conformation results in an axial hole i.e. too small to admit water molecules but yet too

wide to allow vanderwaals associations across the helix axis, this greatly reduces its stability relative to more closely packed conformations.

Beta Structures:

In 1951, Pauling & Corey also postulated the existence of a different polypeptide secondary structure, the β pleated sheet.

- The β pleated sheet differs markedly from the rod like α helix.
- The polypeptide chain in the β -pleated sheet, called a β strand, is almost fully extended rather than being lightly coiled as in the α helix.
- The axial distance between adjacent amino acids is 3.5\AA
- Another difference is that the β pleated sheet is stabilized by Hydrogen bonds between NH & CO groups in different polypeptide strands.



The tertiary structure is the way the secondary structures fold onto themselves to form a protein or a subunit of a more complex protein.

©Bioinformatics Experimental Station, 1997, 1999

- β pleated sheets come in two varieties:
 - 1) The anti parallel β pleated sheet, in which neighboring hydrogen bonded polypeptide chain run in opposite directions.
 - 2) The parallel β pleated sheet in which the hydrogen bonded chains extend in the same Direction.

β Sheets are common structural motifs in proteins. In globular proteins, they consist of from 2 to as many as 15 polypeptide strands, the average being 6 strands, which have an aggregate width of $\sim 25\text{\AA}$. The polypeptide chains in a β sheet are known to be up to 15 residues long, with the average being 6 residues that have a length of $\sim 21\text{\AA}$. A 6 – stranded anti parallel β sheet for example jack bean protein concanavalin A.

- Parallel β sheets of less than 5 stands are rare. This observation suggests that parallel B sheets are less stable than anti parallel β sheets. Because the hydrogen bonds of parallel sheets are distorted in comparison to those of the anti parallel sheets. Mixed parallels ,anti parallel β sheet are common but occur with only $\sim 40\%$ of frequency that would be expected for the random mixing of strand directions.

The topology of the polypeptide strands in a β sheet can be quite complex; the connecting links of these assemblies often consist of long runs of polypeptide chain which frequently contain helices. The link connecting two consecutive anti parallel strands is topologically equivalent to a simple hairpin turn. However, tandem parallel strands must be linked by a cross-over connection i.e. out of the plane of the β sheet. Such cross over connections almost always have a right, handed helical sense, which is thought to be better fit the β sheet inherent right-handed twist.

Non repetitive Structures:

Helices & β sheets comprise around half of the average globular protein. The protein's remaining polypeptide segments are said to have a coil or loop conformation. Random coil. Which refers to the totally disordered & rapidly fluctuating set of conformations assumed by denatured proteins & other polymers in solution?

Fibrous Proteins:

Fibrous proteins are highly elongated molecules whose secondary structures are their dominant structural motifs. Many fibrous proteins such as those of skin, tendon, & bone function as structural materials that have a protective, connective, or supportive role in living organisms. Others such as muscle & ciliary proteins have motive functions.

Ex: α . Keratin, Silk fibrin, Collagen, Elastin

Silk Fibroin: A β pleated sheet

Insects & arachnids produce Silk to fabricate structure & such as cocoons, webs, nests & egg stalks. Most silk consist of the fibrous protein fibroin & a gummy amorphous protein names sericin that connects the fibroin fibers together.

Silk fibroin from the cultivated larvae of the moth Bombyx Mori exhibits an X-ray diffraction patterns indicating that its polypeptide chains form anti parallel β pleated sheets in which the chains extend parallel to the fiber axis. Sequence studies have shown that long stretches of chain are comprised of six residues repeat

(- Gly - Ser - Gly - Ala - Gly - Ala -)_n

Silk fibers are strong but only slightly extensible because appreciable stretching would require breaking the covalent bonds of its nearly fully extended polypeptide chains. Fibers are flexible because neighboring β sheets associate only through relatively weak vanderwaals forces.

Globular Proteins:

These proteins comprise a nightly diverse group of substances that, in their native state, exist as compact Spherical Molecule.

■ Enzymes are globular proteins as are transport & receipt or proteins most of our detailed structural knowledge of proteins & thus to a large extent of their function has resulted from X-ray crystal structure determination of globular proteins & more recently, from their Nuclear Magnetic Resonance structure determination is doing.

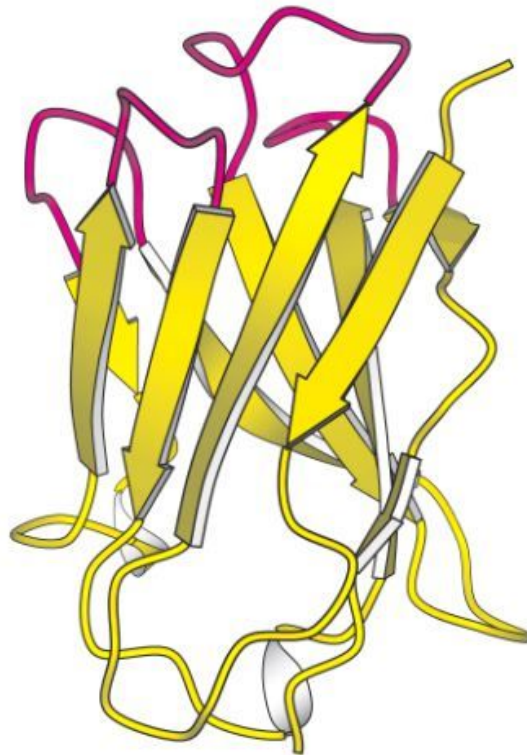
Almost all proteins of > 60 residues contain one or more loops of 6 to 16 residues that are not components of helices or β sheets & whose end to end distances are <10Å. Such Ω (o mega) 100 PS. Which may contain reverse turns, are compact globular entities because their side chains tend to fill in their internal cavities? Since Ω 100 PS

are almost invariably located on the protein surface, they may have an imp role in biological recognition processes.

3.4.2.3 Tertiary Structure:

Fibrous proteins are highly elongated molecules whose secondary structures are their dominant structural motifs many fibrous proteins, such as those of Skin, tendon & bone function as structural materials that have protective, connective or supportive role in living organisms. Others such as muscle & ciliary proteins have motive function.

X- ray crystallographic studies have revealed the detailed three dimensional structures of more than 300 proteins, as will be discussed in subsequent chapters. We begin here with a preview of Myoglobin, the first protein to be seen in atomic detail.
Myoglobin:



It is a Oxygen carrier in muscles is a single polypeptide chain of 153 amino acids and has a mass of 18Kd. The capacity of Myoglobin to bind oxygen depends on the presence of hence, a non polypeptide prosthetic (helper) group consisting of proto porphyrin and a central iron atom. Myoglobin is an extremely compact molecule. Its overall dimensions are 45 x 35 x 25 A an order of magnitude less than if it were bully stretched out. Myoglobin is built primarily of α helices, of which there are eight. About 70% of the main chain is bolded into α helices, and much of the rest of the

chain forms turns between helices. Four of the turns contain proline, which tends to disrupt α helices because of steric hindrance by its rigid bimembered ring.

The folding of the main chain of Myoglobin, like that of other proteins, is complex and devoid of symmetry. However, a unifying principle emerges from the distribution of side chains. The striking fact is that the interior consists almost entirely of nonpolar residues such as leucine, valine, methionine & phenylalanine. Polar residues such as aspartate, glutamate, lysine & arginine are absent from the inside of Myoglobin. The only polar residues inside are two histidines, which play critical roles in the binding of haem oxygen. The outside of Myoglobin, on the other hand, consists of both polar & nonpolar residues. The space-filling model also shows that there is very little empty space inside.

The polypeptide chain therefore folds spontaneously so that its hydrophobic side chains are buried & its polar, charged chains are on the surface.

An unpaired peptide NH or CO markedly prefer water to a nonpolar milieu. The secret of burying a segment of main chain in a hydrophobic environment is to pair all the NH & CO groups by hydrogen bonding. This pairing is neatly accomplished in an α -helix or β sheets. Van der Waals bonds between tightly packed hydrocarbon side chains also contribute to the stability of proteins.

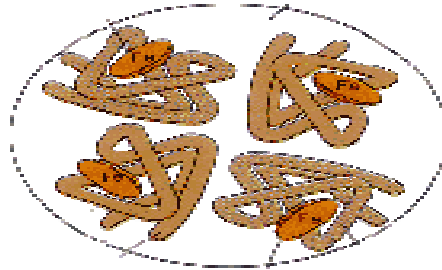
Another example is Fibronectin - A. It is a digestive enzyme present in pancreas. It has a complex 3-dimensional structure. It contains both α helices & β strands & it contains 4 disulfide linkages. The diameter is 20 Å.

3.4.2.4 Quaternary structure:

Proteins because of their multiple polar and non-polar groups stick to other proteins. Svedberg discovered that some proteins are composed of more than one polypeptide chain. These polypeptide subunits associate in a geometrically specific manner. The spatial arrangement of these subunits is known as Quaternary structure. In large assemblies of proteins, such as collagen fibrils, the advantages of subunit construction over the synthesis of one huge polypeptide chain are analogous to those of using pre-fabricated components in constructing a building. In case of enzymes, increasing a protein size tends to better fix the three-dimensional positions of the groups forming the enzyme's active site.

The multi-subunit protein may consist of identical or non-identical polypeptide chains. For eg: Hemoglobin, has the subunit composition $\alpha_2\beta_2$ this will refer to proteins with identical subunits as oligomers and to these identical subunits as protomers. A protomer may therefore consist of one polypeptide chain or several unlike polypeptide chains. In this sense, hemoglobin is a dimer of $\alpha\beta$ protomers. The contact regions between the subunits closely resemble the interior of a single subunit protein.

3.4.2.5 Symmetry in proteins:



In the vast majority of oligomeric proteins, the protomers are symmetrically arranged. Proteins can not have inversion or mirror symmetry, however, because such symmetry operations convert chiral L-residues to D-residues. Thus proteins can only have rotational symmetry. Various types of rotational symmetries are there. Those are

1. Cyclic symmetry:

It is simplest type of rotational symmetry. Sub units are related by a single axis of rotation. Object with 2,3,.....,n rotational axes are said to have C_2, C_3, \dots, C_n symmetry respectively. C_2 symmetry is the most common symmetry in proteins. Higher cyclic symmetries are relatively rare.

2. Dihedral symmetry:

Dihedral symmetry (D_h) is a more complicated type of rotational symmetry, is generated when an N-fold rotation axis and a two fold rotation axis intersect at right angles. An oligomer with D_h symmetry consists of $2n$ protomers. The D_2 symmetry is by far, the most common type of dihedral symmetry in proteins.

3. Other rotational symmetry:

The other rotational symmetries are tetrahedron symmetry (T), a cube or octahedron (O) and have 60 equivalent positions respectively. The subunit arrangements in the protein coat of spherical viruses are based on icosahedral symmetry for e.g.: that the 600 KD E.Coli glutamine synthetase has D_6 symmetry.

4. Helical symmetry:

Some protein oligomers have helical symmetry. The chemically identical sub units in a helix are not strictly equivalent because, for instance those at the end of the helix have a different environment than those in the middle. Nevertheless, the surroundings of all subunits in a long helix, except those near its ends are sufficiently similar that the sub units are said to be quasiaequivalent.

3.4.2.6 Protein Stability:

Native proteins are only marginally stable entities under physiological conditions. The free energy required to denature them is 0.45 /mol of amino acid residue so that 100 residue proteins are typically stable by around 40 KJ/Mole only. For stability of proteins there are many bonds they are

- Electrostatic forces
- Hydrogen bonds
- Hydrophobic bonds
- Disulfide bonds

1) Electrostatic forces:

Molecules are collection of electrically charged particles & hence, their interactions are determined by laws of classical electrostatics. The forces present are *Ionic interaction*: The association of two ionic protein groups of opposite charge is known as ion pair or salt bridge. This ion pairs contribute little stability towards a protein's native structure.

Vanderwaals forces: The noncovalent associations between electrically neutral molecules, collectively known as vanderwaals forces, arise from electrostatic interactions among permanent & induced dipoles. These forces are responsible for numerous interactions of varying strengths between non-bonded neighboring atoms.

- In the low dielectric constant core of a protein dipole, dipole interactions (vanderwaals forces) significantly influence protein folding.

Hydrogen bonding:

It is a electrostatic interaction between a weakly acid donor group (D-H) & an accept or atom that bears a lone pair of electrons.

- Hydrogen bonds, which have association energies in the range -12 to 30 KJ/mole, are much more directional than are vanderwaals although less so than are covalent bonds. The D...A distance is normally in the range 2.7 to 3.1 Å. This hydrogen bonding has major influence on the structures of proteins.
- The internal hydrogen bonds of proteins provide a structural basis for its native folding pattern.

Hydrophobic bonds:

The hydrophobic effect is the name given to those influences that cause non polar substance to minimize their contacts water and amphipathic molecules. Such as soaps & detergents to form micelles in aqueous solutions. Hydrophobic interactions must be an important determinant of protein structures.

Disulfide bonds:

The function of disulfide bond is to stabilize the protein three dimensional structures. Almost all proteins with disulfide bonds are secreted to more oxidized extra cellular destinations where their disulfide bonds are effective in stabilizing protein structure.

3.4.2.7 Summary

Protein structure can be explained at four levels viz primary, secondary tertiary and quaternary, where primary structure explains linear structure of poly peptide and quaternary structure is highly folded confirmation. The secondary structure of proteins exists in different forms like α helix and β pleated sheets. The helix is generally right handed .The helices are stabilized by various forces like electrostatic forces, disulfide bonds etc

3.4.2.8 Model questions

- 1) Write in detail about the different levels of organization of proteins structure
- 2) Write a short note on protein diversity

3.4.2.9 Reference books

- 1) Principles of biochemistry by Nelson and Cox
- 2) Biochemistry by O.P Agarwal

Lesson 3.4.3**DNA STRUCTURE****Contents****3.4.3.1 Introduction****3.4.3.2 Griffith's experiment****3.4.3.3 Structure****3.4.3.4 Secondary structure / double helical structure****Summary****Model Questions****References****Objective**

The objective of this lesson is to know the DNA structure and its organisation

3.4.3.1 Introduction:

In 1865, Mendel showed that genes transmitted genetic information. The classical genetics of early twentieth century showed that the genetic material must perform three essential functions.

1. Replication – genotypic function
2. Gene expression – phenotypic function
3. Mutation – evolutionary function

Other early genetic studies established a precise correlation between the patterns of transmission of genes and the behaviour of chromosomes during sexual reproduction, providing strong evidence that genes are usually located on chromosomes.

Chromosomes are composed of two types of large organic molecules (macromolecules) called proteins and nucleic acids. The nucleic acids are of two types : deoxyribonucleic acid (DNA) and Ribonucleic acid (RNA). During the 1940s and early 1950s, the results of elegant experiments clearly established that the genetic information is stored in nucleic acids, not in proteins. In most organisms, the genetic information is encoded in the structure of DNA. However, in many small viruses, the genetic information is encoded in RNA.

Proof that genetic information is stored in DNA

Several lines of indirect evidences suggested that DNA harbors the genetic information of living organisms.

For example:

- Most of the Cell's DNA is located in the chromosomes, whereas RNA and proteins are also abundant in cytoplasm.
- A precise correlation exists between the amount of DNA per cell and the no of sets of chromosomes per cell.
- Most somatic cells of diploid organisms contain twice the amount of DNA as the haploid germ cells (gametes) of the same species.
- The molecular composition of the DNA is the same (with rare exceptions) in all the cells of an organism, where as the composition of RNA and proteins is highly variable from one cell type to another.

- DNA is more stable than RNA or proteins.

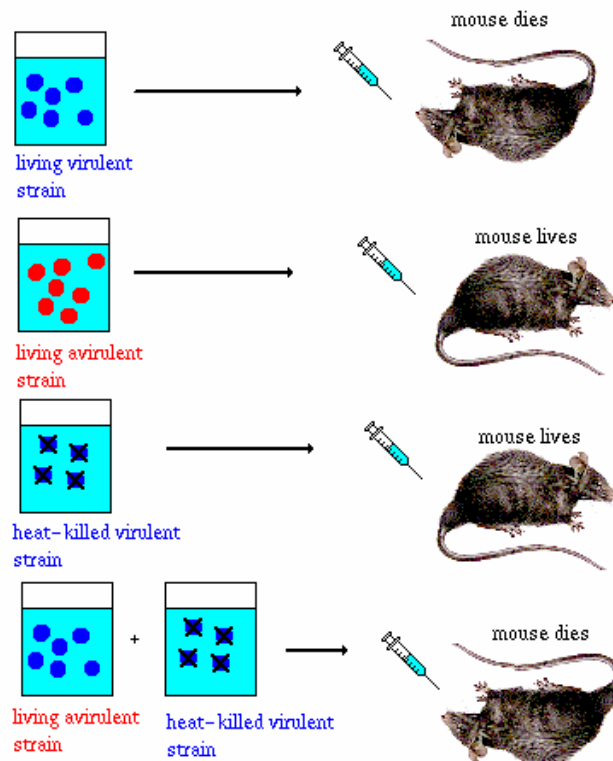
Transformation

It involves the uptake of naked DNA molecules from one bacterium (the donor) by other bacterium (the recipient). It was discovered by Frederick Griffith in 1928 which laid foundation for the identification of DNA as the genetic material in the bacterium *streptococcus pneumoneae*.

3.4.3.2 Griffith's experiment

The wild-type organism is a spherical cell surrounded by mucous coat called capsule. They form large, glistening, smooth colonies. These cells are virulent, capable of causing lethal infections upon injection into mice.

A certain mutant strain of *S.pneumoniae* has lost the ability to form a capsule. As a result, it forms small, rough colonies and is avirulent. Fig.



When virulent strains were injected into mice, they died. The mice were alive when both avirulent and heat-killed virulent strains were injected independently. However, the mice died when a heat-killed virulent strain + avirulent strain was injected. Griffith called this conversion of avirulent strains to virulent strains as transformation. This transformation was not transient, the ability to make a capsule and therefore to kill host cells, once conferred upon the avirulent bacteria, was passed to their descendants as a heritable trait. In other words, the gene for virulence, missing in avirulent cells, was

somehow restored during transformation. It means that the transforming substance in the heat killed bacteria was probably the gene for virulence itself.

DNA as the transforming principle

In 1944, Avery McLeod and McCarty showed that the transforming principle was DNA.

- First, they removed the protein from the extract with organic solvents and found that it still transformed.
- Trypsin, chymotrypsin which destroy protein had no effect on transformation. Neither did Ribonuclease which destroy RNA.
- On the otherhand they found that the enzyme Dnase which breakdown DNA, destroyed the transforming ability of the virulent cell extract.

Finally, direct physico-chemical analysis showed the purified transforming substances to be DNA.

Ultracentrifugation

The material with transforming activity sedimented rapidly suggesting a very high molecular weight, characteristic of DNA.

Electrophoresis

Transforming activity had a relatively high mobility, also characteristic of DNA.

UV-absorption spectrophotometry

Its absorption spectrum matched that of DNA, i.e, maximum absorption at 260nm.

Elementary chemical analysis

This yielded an average Nitrogen / Phosphorous ratio of 1.67, equal to that of DNA which is rich in both elements.

Genetic transformation has been demonstrated in several other genera including Haemophilus,

Bacillus
Salmonella
Sheptococcus
Rhizobium
Neisseria and in

Higher organisms

Drosophila
phestia insects
Bombyx

Mice and humans cells cultured invitro. Even in these species, all cells in a given population are not capable of active uptake of DNA.

Only competent cells, which possess a so called competence factor, are capable of serving as recipients in transformation. Competance of bacteria is not a permanent feature but occurs only at certain times in life cycle. Competence is commonly observed towards the ends of the 'log' phase of growth just before the stationary phage.

There are two theories to explain development of competence.

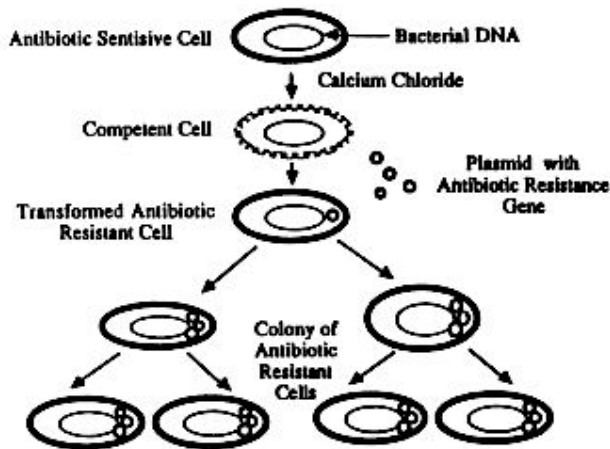
1. Structure of cellwall is critical. It permits uptake of DNA only during the restricted competence phase and its permeability to macromolecules may change with growth conditions.
2. Competence results from the synthesis of specific receptor sites on the surface of the cell. This view is supported by the fact that synthesis of new proteins is necessary for development of competence. Inhibition of proteins or RNA synthesis inhibits the transformation.

Stages in transformation

1. DNA comes into contact with the bacterial cell surface as a result of random collision. The binding becomes irreversible after a very short period (5-6 seconds).
2. Permanently bound DNA penetrates the bacterium. Ds DNA is converted into ss DNA by the action of exonuclease. Penetrating DNA must have a minimum length of about 750 basepairs.
3. ssDNA is stabilized by a competence-specific protein. ssDNA migrates from the periphery of the cell to the chromosome DNA.
4. The homologous portion synapses with the recipient chromosome. The unsynapsed DNA is out by means of nuclease action.
5. The transformation heteroduplex undergoes replication to form transformation homoduplexes. One of these is a normal duplex, while the other is transformed duplex. The clone produced from the transformed duplex is the transformed. The normal duplex will give rise to a non-transformed duplex.

Fig.

Figure 2. Transformation of Bacteria with Plasmid DNA.



Proof that DNA is the Genetic Material in T₂ Bacteriophage

Finally in 1952, A.D. Hershey and Martha chase performed an experiment to prove that DNA was the genetic material. Their experiment involved a bacteriophage, called T₂ that infects the bacterium E.coli. During infection, the phage genes enter the host cell and direct the synthesis of new phage particles.

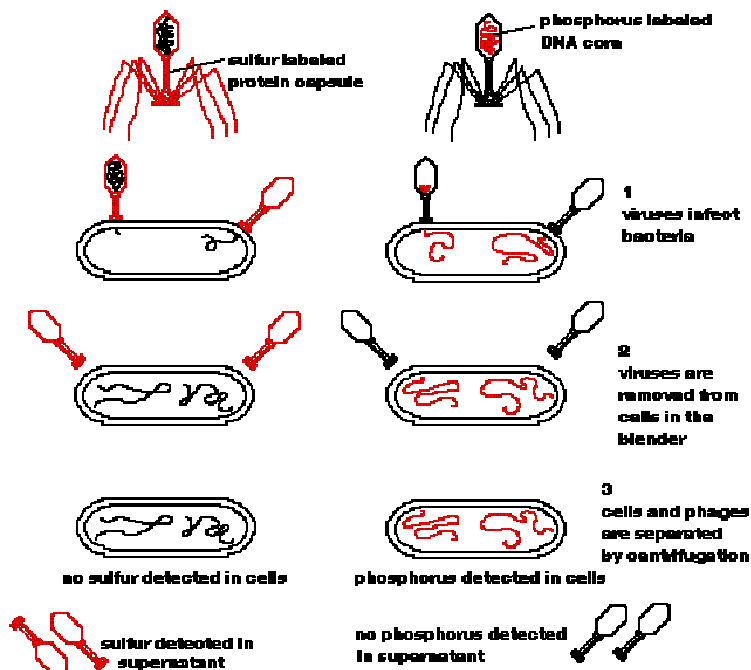
Their experiment depended on radioactive labels on the DNA and protein, a different label for each. ³²p for DNA, ³⁵s for protein, DNA is rich in phosphorous while phage protein has none, and protein contains sulfur but DNA does not.

Hershey and chase allowed the labeled phages to attach by their tails to bacteria and inject their genes into their hosts. Then they removed the empty phage coats by homogenizing in a blender.

Since the phage genes must enter the cell, they reasoned that the type of label found in the infected cell would indicate the nature of the genes. From the above experiment the conclusion was that the genes are made of DNA.

Fig.

The Hershey-Chase Blender Experiment



Deoxyribonucleic acid

Friedrich Miescher, in 1869 had isolated a previously unidentified macromolecular substance, to which he gave the name nuclein. Nuclein was later renamed nucleic acid. Development of DNA-specific staining techniques by Feulgen

and Rossenbeck in 1924 enabled Feulgen to demonstrate in 1937 that most of the DNA content of a cell is located in the nucleus.

Nucleic acids are macromolecules present in all living cells, either in free state or in combination with proteins. Nucleic acids are polymers consisting of units called nucleotides. They are hence called polynucleotides.

Nucleotides

These are the compounds constituted by purine or pyrimidine bases, deoxyribose sugars and phosphoric acid.

Importance

1. Purine nucleotides act as the high energy sources: ATP, GTP.
2. Serve as monomeric precursors of RNA & DNA.
3. Play an important role in carbohydrate, fat, protein metabolism.
4. They also serve as chemical signals Ex: cAMP, cGMP.
5. Function as components of coenzymes FAD, NAD⁺ etc. and an important methyl donor, SAM.
6. also act as high energy intermediates such as UDP-glc & UDP-gal in carbohydrate metabolism and CDP-acyl glycerol in lipid synthesis.
- 7.

Nitrogenous bases

The bases are derivatives of two parent compounds: purines & pyrimidines. These are weakly basic.

Pyrimidine bases

Pyrimidine bases found in Nucleic acids are mainly three.

- Cytosine – found in both DNA and RNA
- Thymine – found in only DNA
- Uracil - found in only RNA

All the pyrimidine bases can exist in lactam form and lactim form. If the group is –NH-CO- it is called Lactam (keto) type, while the same if isomerizes to –N=C-OH, it is called lactim (enol) type. At the physiological pH, the lactam forms are predominant.

- chemically 2-oxy-4-amino pyrimidine
- is found in all nucleic acids except DNA of certain viruses.

Thymine

- Chemically, it is 2,4-dioxy – 5- methyl pyrimidine
- Also called as 5-methyl uracil
- Occurs only in DNA, however, minor amounts have recently been found in tRNA.

Uracil

- chemically it is 2,4-dioxy pyrimidine
- is confined to RNA only, not found in DNA

Purines

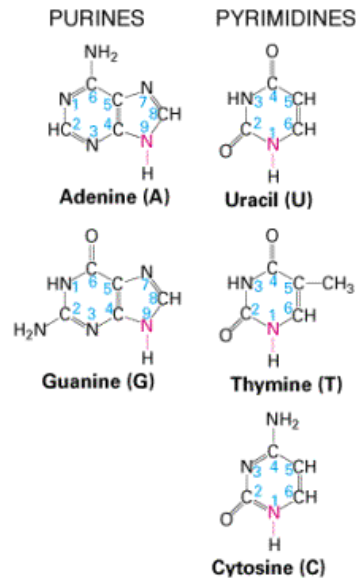
- Purine ring is more complex than pyrimidine
- It can be considered as the product of fusion of a pyrimidine ring with an imidazole ring.
- Adenine & Guanine are the two principal purines
- Found in both DNA & RNA.

Adenine

- chemically it is 6-amino purine

Guanine

- chemically it is 2-amino- 6-oxy purine

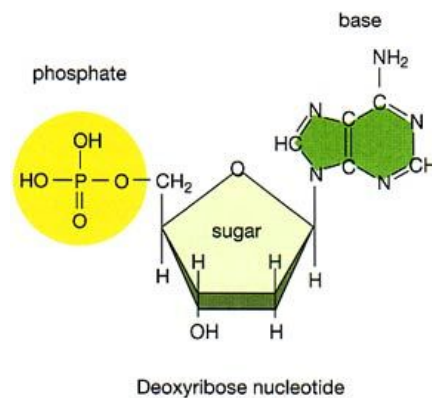
**Minor bases in DNA**

- 5-methyl cytosine occur in plants
- N⁶-methyl adenine in bacterial DNA
- 5-hydroxy methyl cytosine in bacteria infected with certain bacteriophages.

Sugar

2-deoxy B-D-ribose. The sugar is in its furanose form in nucleic acids. An important property of the pentoser is their capacity to form esters with phosphoric acid. The -OH group of the pentose, especially those at C₃ & C₅ are involved forming a 3', 5' - phosphodiester bond.

Fig.



Phosphoric acid

The molecular formula of phosphoric acid is H_3PO_4 . It contains 3 monovalent –OH groups and a divalent oxygen atom, all linked to the pentavalent phosphorous atom.

Nucleotides are the phosphoric acid esters of nucleosides. These occur either in the free form or as subunits in Nucleicacids.

Deoxyribonucleotides

- Deoxy adenylic acid
- Deoxy cytidylic acid
- Deoxy thymidylic acid
- Deoxy guanylic acid

Nucleosides are composed of a purine or pyrimidine base and a deoxyribose sugar.

The Nucleotides of DNA

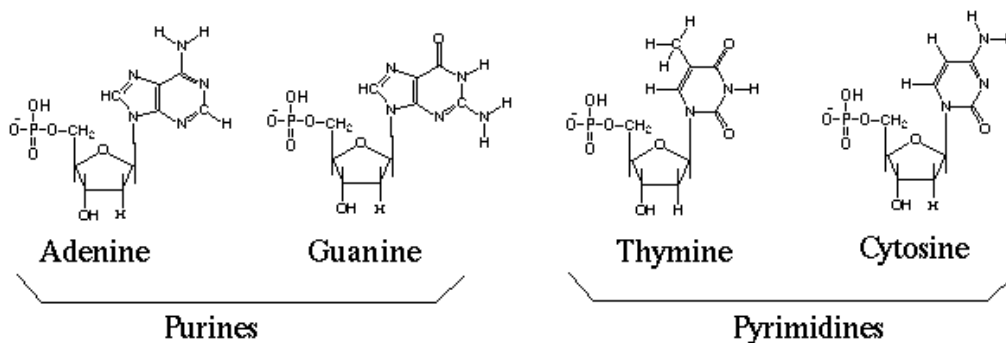


Fig.

The base is joined covalently, at N-1 of pyrimidine and N-9 of purines in an N-glycosyl linkage to the 1 carbon of the pentose.

The antiform is necessary for the proper positioning of the complementary purine and pyrimidine bases in the dsDNA.

Deoxyribonucleosides : Deoxy adenosine

- Deoxy Guanosine
- Deoxy Cytidine
- Deoxy Thymidine

Synthetic derivatives

Synthetic nucleobases, nucleosides, nucleotides are widely used in the medical science & clinical medicine. Changes in heterocyclic ring structure and sugar moiety, induces toxic effects when incorporated into cells and inhibits the activities of enzymes.

- 6-thioguanine
- 6-mercaptopurine
- 4- hydroxy pyrazole pyrimidine
- also called as allopurinol
- inhibitor of xanthine oxidase.
 - cytarabine (arabinosyl cytosine) – used in the chemotherapy of cancer & viral infections
 - Vidarabine (arabinosyl adenine)
 - Azathioprine – useful in organ transplantation

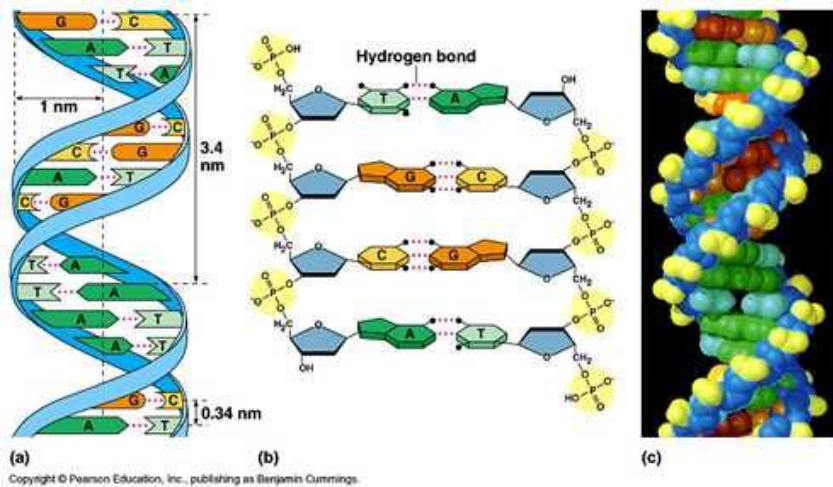
Nucleic acid

A nucleic acid is a polymer of a nucleotide monomer and can be considered as a polynucleotide.

The successive nucleotides in DNA are covalently linked through phosphate groups bridges, specifically the 5'-OH group of one nucleotide unit is joined to the 3'-OH group of the next nucleotide by a phosphodiester bond. Thus the back bone of nucleic acids consist of alternating phosphate and pentose residues, and the characteristic bases may be regarded as side groups joined to the backbone at regular intervals.

Each linear nucleic acid has a specific polarity and distinct 5' and 3' ends. The 5' end lacks a nucleotide at 5' position and the 3' end lacks a nucleotide at 3' position.

Fig.



The back bone of phosphate and sugar is hydrophilic, where as the bases is hydrophobic.

3.4.3.3 Structure

DNA structure contains hierarchial levels of complexity.

1. Primary structure → covalent structure of nucleotides forming a linear chain.
2. Secondary structure → any regular, stable structure taken up by some or all of the nucleotides.
3. Tertiary structure → The complex folding of large chromosomes with in the bacterial nucliod & eukaryotic chromatin.

A most important due to the structure of DNA came from the work of Erwin chargraff and his colleagues in the late 1940s. They concluded that:

- the base composition of DNA generally varies from one species to another.
- DNA specimens isolated from different tissues of the same species have the same base composition.
- The base composition of DNA in a given species doesnt change with the organism's age, nutritional state or changing environment.
- In all DNAs, reardless, of species, the no. of A residues is equal to the no. of T residues and the no. of G is equal to the no. of C residues, i.e., the no. of purines

= the no. of pyrimidines. ($A+G = T+C$). This is sometimes referred as Chargaff's rule and is the key for establishing 3-dimensional structure of DNA.

3.4.3.4 Secondary structure / double helical structure

Evolution

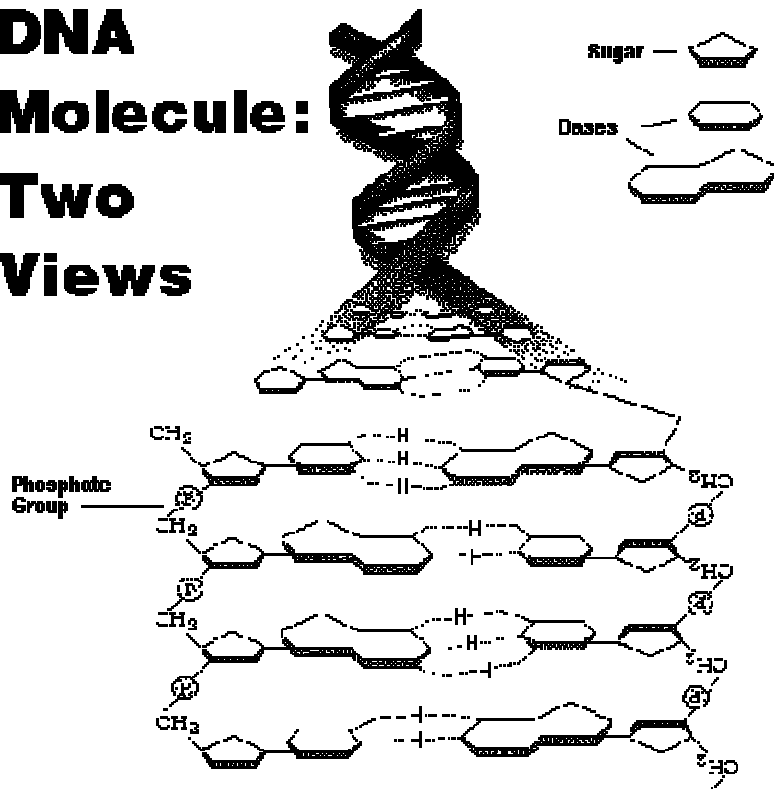
- W.T. Astbury was the first person to propose the 3-D-structure of DNA. By his X-ray crystallography study on DNA, he concluded that, because DNA has a high density, its polynucleotide was a stack of flat nucleotides, each of which was oriented perpendicularly to the long axis of the molecule & was placed 3-4 Å apart from each other.
- Continued crystallography studies by Wilkins & R. Franklin confirmed Astbury's 3-4 Å internucleotide distance and suggested a helical configuration for DNA molecule. They also suggest that, the helix is folded into many turns and each turn causes a vertical rise of 34 Å.
- Analytical studies also suggest that the polynucleotide chains were held together by H-bonding between the base residues.

Watson and Crick model of DNA

- In 1953, Watson & Crick postulated a three-dimensional model of DNA structure from the all available data.
- It consists of two helical DNA chains coiled around the same axis to form a right handed helix.
- They hydrophilic backbones of alternating deoxyribose & negatively charged phosphates are on the outside of the double helix, facing the surrounding water.
- The purine & pyrimidine bases of both strands are stacked inside the double helix with their hydrophobic & nearly planar ring structure very close together and as perpendicular to the long axis.
- The spatial relationship between these two strands creates a major groove and a minor groove between the two strands.
- The diameter of the helix is 20 Å, the bases are 3.4 Å apart along the helix axis. Each turn of the helix contains 10 nucleotides residues. Therefore the helical structure repeats at intervals of 34 Å.
- The two chains are held together by H-bonds between pairs of bases. Adenine always pairs with thymine by two H-bonds and guanine always pairs with cytosine by three

Fig.

DNA Molecule: Two Views



- The two chains or strands of the helix are antiparallel, their 5', 3'-phosphodiester bonds run in opposite directions.
- The two strands are complementary to each other. Where ever adenine appears in one strand, T is found in the other; similarly wherever G is found in one chain, C is found in the other.

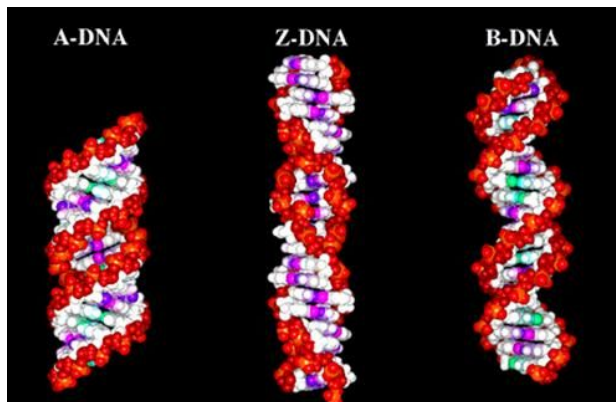
Different structural forms of DNA

DNA is remarkably flexible molecule. The Watson-Crick structure is also referred as B-form; which is the more stable, right handed DNA molecule. Many significant derivations from the W-C DNA structure are found in cellular DNA, and some (or) all of these may play an important role in DNA metabolism.

A-form

- it is a right handed helix
- it is favoured in many solutions that are relatively devoid of water.
- No. of base pairs per helical turn is 11.
- Rise per basepair is 2.3 Å
- Is shorter and have a greater diameter
- The reagents used to promote crystallization of DNA tend to dehydrate it, and this leads to a tendency for many DNAs to crystallize in the A-form.

Fig.

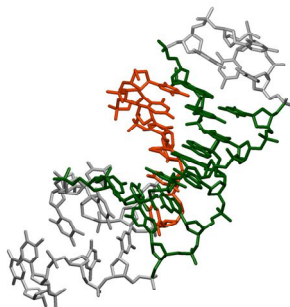
**Z-form**

- left-handed double helix
- 12-bp per helical turn
- rise per basepair is 3.8 Å
- DNA back bone takes zig-zag appearance.
- Sequences in which pyrimidine alternating with purine will give Z-forms.
- There is evidence for short stretches of Z-DNA both in prokaryotes & eu-karyotes.
- These Z-DNA tracts may play an undefined role in the regulation of gene expression or genetic recombination.

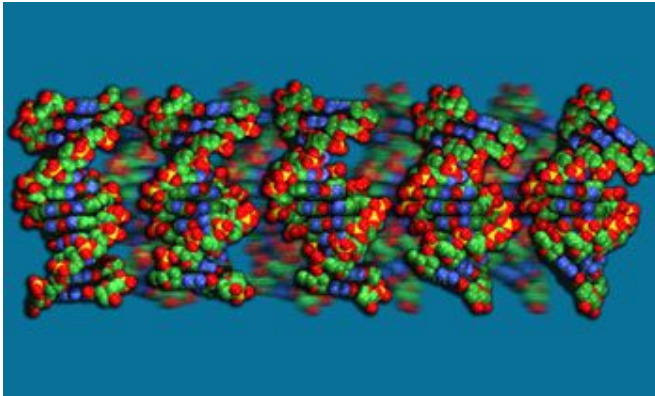
H-DNA

- triple helical DNA
- unusual structure
- occur in polypyrimidine / polypurine tracts
- pairing and interwinding of three strands.
- They form spontaneously only within long sequences containing only pyrimidines or only purines in one strand.
- Two of the 3 strands in the H-DNA triple helix contain pyrimidines and the third contains purines.
- Found within regions involved in the regulation of expression of a number of genes in eukaryotes.

Fig: H DNA



Unusual structures of DNA



These are formed during initiation of DNA metabolism (Replication, transcription, translation) and / or in regulation of gene expression.

Denaturation & Renaturation

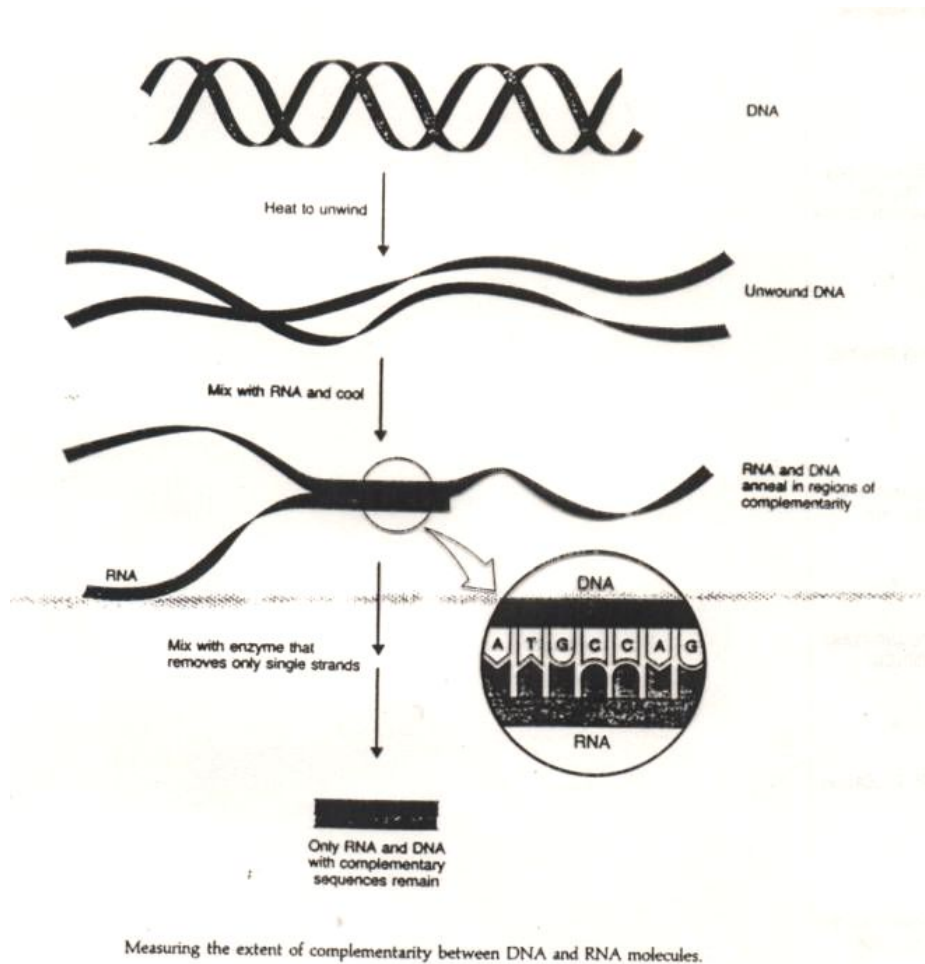
Solutions of carefully isolated, native DNA are highly viscous at pH 7.0 and room temperature (20-25°C). When such solution is subjected to extreme temperature or pH its viscosity decreases sharply. This is due to a process called denaturation.

Denaturation of helical DNA involves disruption of the H-bonds between the paired bases and the hydrophobic interactions between the stacked bases. As a result, the double helix unwinds to form two single strands, completely separate from each other along the entire length or part of the length (partial denaturation) of the molecule. No covalent bonds in DNA are broken.

When the temperature or pH is returned to normal biological range, the unwound segments of the two strands spontaneously rewind or anneal to yield the intact duplex. This is called renaturation. If the two strands are completely denatured, the process of renaturation occurs in two steps.

1. Slow first step, because two strands must find each other by random collision and form short segment of complementary double helix.

Fig.



2. The second step is much faster, the two strands 'Zipper' themselves together to form the double helix.

Careful, controlled denaturation cause, denaturation of A-T rich region, while the other length of DNA is double helix. This partial denaturation is required for initiation of DNA replication and transcription.

T_M

The transition of double helical DNA to single stranded DNA can be accomplished by heating a solution of DNA or by adding acid or alkali to ionize its bases. The unwinding of double helix is called melting, because it occurs abruptly at a certain temperature.

Melting temperature is the temperature at which half of the helical structure is lost. This transition indicates that the DNA is a highly cooperative structure, stabilized by the stacking of bases as well as by basepairing.

The T_M of a DNA molecule depends markedly on its base composition. DNA molecules rich in G-C bps have a higher T_M than those having an abundance of A-T

basepairs. The T_M of DNA from many species varies linearly with G-C content, rising from 77°C-100°C as the amount of G-C pairs increases from 20-75%.

Hybridization

The double helical DNA contains two complementary strands. If the separated strands from two different species are reannealed to form a double helix, there occurs a hybrid DNA. This process is called hybridization.

Ex: one strand from human DNA & one strand from mouse DNA can form a duplex (hybrid) DNA.

- Two DNAs from different species are completely denatured by heating.
- When mixed and slowly cooled, complementary DNA strands of each species will associate & anneal to form duplexes.
- If the homology is high, greater no. of hybrids are formed.
 - It is the basis for essential techniques in molecular genetics
 - Can be used to detect specific RNA
 - To detect sequence homology among different species
 - To detect evolutionary heritage.

Hybrid formation can be measured by different procedures such as chromatography or isopycnic centrifugation. Usually one of the DNAs is labeled with a radioisotope to simplify the measurement.

Tertiary structure

The DNA double helix can undergo coiling about its own axes to produce a supercoiled tertiary structure. Thus DNA can exist in forms other than a linear molecule. In bacterial, viral replicative forms, plasmids, mitochondrial, and chloroplast DNA, the ends of the DNA molecules are covalently joined to form a closed, circular duplex molecule. In the much larger eukaryote chromosomes, supercoiling arises when the DNA coils around histones.

The terms supercoiling, superhelicity and supertwisting are employed for the twisting of DNA duplex upon itself. This property of DNA is an integral feature of all chromosomes, whether circular or linear. It has been shown to be essential for the stages of replication, transcription and recombination.

Discovery of supercoiling in the polyoma virus

In 1965, Vinograd and his associates discovered that the genetic material of the polyoma virus was easily renatured after denaturation by heat. They also found that the individual strands of the DNA duplex did not separate from each other. It was therefore concluded that the two strands were covalently closed and intertwined with each other. Sedimentation studies of the polyoma DNA showed the existence of 3 components.

- a linear or broken open form of the genome.
- A twisted loop or supercoiled form
- A relaxed loop containing at least one nick or break in one of the strands.

If the two ends of the linear DNA are joined, a covalent relaxed circle is formed. If the supercoiled DNA undergoes a nick in one strand, a relaxed circle is formed. If it undergoes nicks in both strands, the linear form results.

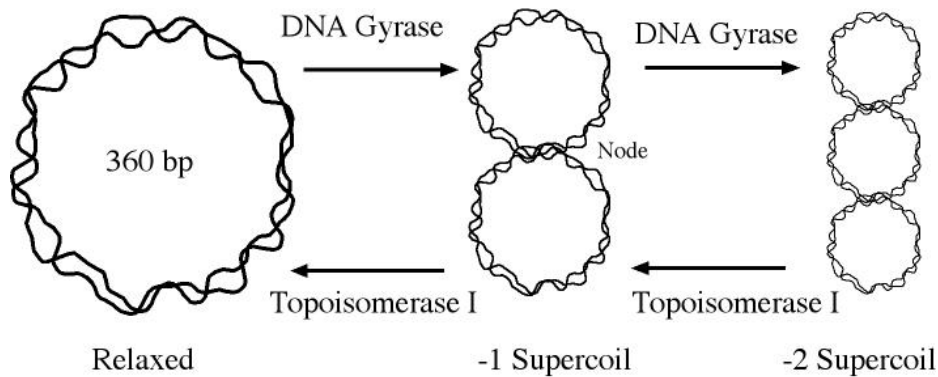
Negative and positive superhelices

The structure of the double helix which is thermodynamically favored contains one complete turn per 10 basepairs. For each turn of the helix, the strands cross twice.

This structure is found in linear DNA. The circular structure consists of fewer turns of the helix. This double helix characterized by a deficit of turns is called a negative superhelix. The deficit in turns can be accommodated by breaking of the H-bonds and the opening of the double helix over a small region. Another way of accommodating the strain is by the formation of the tertiary structure with supercoils. DNA from natural sources is negatively supercoiled. Usually there is one negative twist in the DNA double helix per 15 turns of the helix. In a different form of the superhelix structure, an excess of helical turns is present. This type of helical structure is called positive superhelix. In this, strain can be accommodated only through the formation of supercoils.

Fig.

DNA Supercoiling



Overwound DNA - positive supercoiling
Underwound DNA - negative supercoiling

Linking number : α or L

The linking number, also known as the linkage number, is a topological parameter which characterizes closed circular dsDNA. It specifies the no. of times the two complementary strands of DNA duplex twist around each other in the DNA circle. The linkage number can change only by breaking and resealing covalent bonds in DNA, as in the case of DNA topoisomerase treatment. For relaxed B-form DNA, the linking no is the no. of basepairs in the molecule divided by 10 conventionally, the linkage number is counted so that it is positive for each crossover in a right-handed helix.

Enzymatic activity altering DNA supercoiling

Topoisomerases

These are nicking-closing enzymes whose functions depend on supertwisting of DNA. They catalyze the breaking (nicking) and rejoining (closing) of phosphodiester bonds. This alters the topology of DNA without affecting its primary structure.

DNA topoisomerases have been isolated from viruses and from bacterial, plant & animal cells. The E.coli omega (ω) protein was the first topoisomerase discovered, and has been renamed as Eco DNA topoisomerase I. Topoisomerases convert or isomerise

one topological version of DNA into another by changing its linkage no. (the no. of times two DNA chains twist around each other). Topoisomerase action is implicated in replication, segregation of replicas, transcription, recombination and nucleosome assembly.

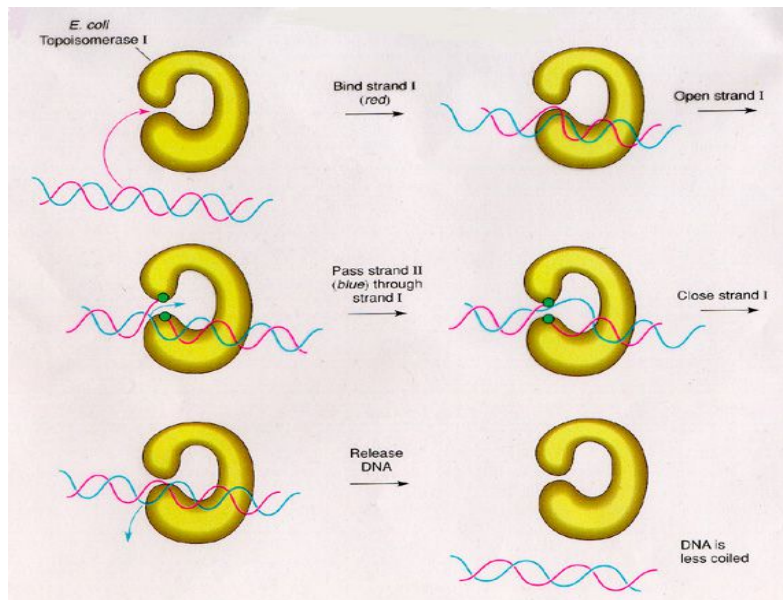
There are two classes of topoisomerases, type I (topo I) and type II (topo II, gyrase) with counterbalancing action.

Supercoiled dsDNA relaxed DNA

Topo I

- The *E. coli* type I topoisomerase is a monomer (100 Kda), encoded by the top A gene. It breaks and reseals one strand of DNA, changing the linkage no. in steps of 1. The enzyme binds to duplex DNA and unwinds the double helix locally. It then nicks one strand, and the free phosphate on the DNA becomes covalently attached to a tyrosine residue in the enzyme. Free rotation of the helix is prevented by the cut ends of the DNA remaining bound to the enzyme. The other strand is passed through the break, and the complex rotates, relieving a supercoil. The enzyme now ligates the cut ends. The linkage no. is increased by 1. The enzyme becomes separated from the DNA, which undergoes renaturation. The reaction does not require energy. The end result is DNA with one less negative supercoil.

Fig.



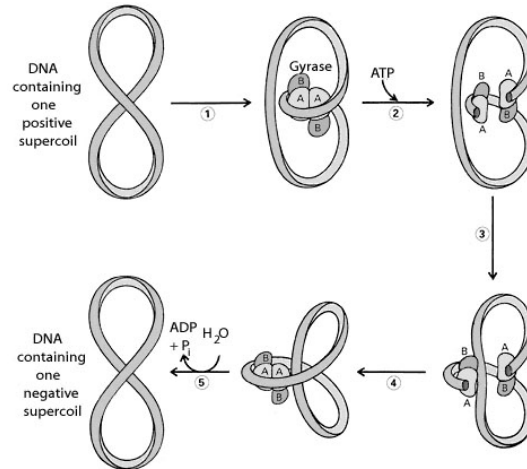
Topo II (gyrase)

E. coli type II topoisomerase is an A_2B_2 tetramer (mole wt 400 Kda), encoded by *gyr A* or *gyr B* gene. Each polypeptide has a molecular weight of 105 Kda. The eukaryote (Hela) type II isomerase is a dimer (mol. wt 309 Kda) each subunit of which has a molecular weight of 172 Kda.

Type II isomerase break and reseal both strands of DNA, changing the linking number in steps of two. The enzyme can cut a ds DNA molecule, pass another duplex

through the cut, and reseal the cut. This activity requires ATP. The effect of enzyme action is to change a positive supercoil into a negative supercoil.

Fig.



<http://138.192.68.68/bio/Courses/biochem2/DNA/DNAStructure.html>

SUMMARY:

The nucleic acids are of two types : Deoxyribonucleic acid (DNA) and Ribonucleic acid (RNA). DNA acts as the genetic material. DNA is located in the chromosomes. DNA is more stable than RNA or proteins. It was discovered by Frederick Griffith in 1928. In 1944, Avery McLeod and Madyn Mc Carty showed that the transforming principle was DNA. Finally in 1952, A.D. Hershey and Martha Chase performed an experiment to prove that DNA was the genetic material.

DNA consists of deoxy nucleotides i.e. Deoxy Adenylic acid, Deoxy Guanylic acid, Deoxy Cytidilic acid, Thymidilic acid. DNA exists in 3 structures those are

Primary structure is a covalent structure of nucleotides forming a linear chain. Secondary structure is a any regular, stable structure taken up by some or all of the nucleotides. Tertiary structure is a The complex folding of large chromosomes with in the bacterial nucleoid & eukaryotic chromatin. In DNA tertiary structure super coils are eliminate by DNA Gyrases.

Model Questions:

1. Explain the structure of DNA with reference to Watson and Crick?
2. Prove that DNA as the genetic material?

References

Molecularbiology of the Gene by Wattson
 Stryer's Biochemistry
 Biochemistry by Voet and Voet

S.Lavanya

Lesson-3.4.4

RNA Structure

Contents

3.4.4.1 INTRODUCTION

3.4.4.2 RNA structural organization

- 1) Primary structure
- 2) Secondary structure
- 3) Tertiary structure

3.4.4.3 TYPES OF RNA

- Summary
- Model questions
- References

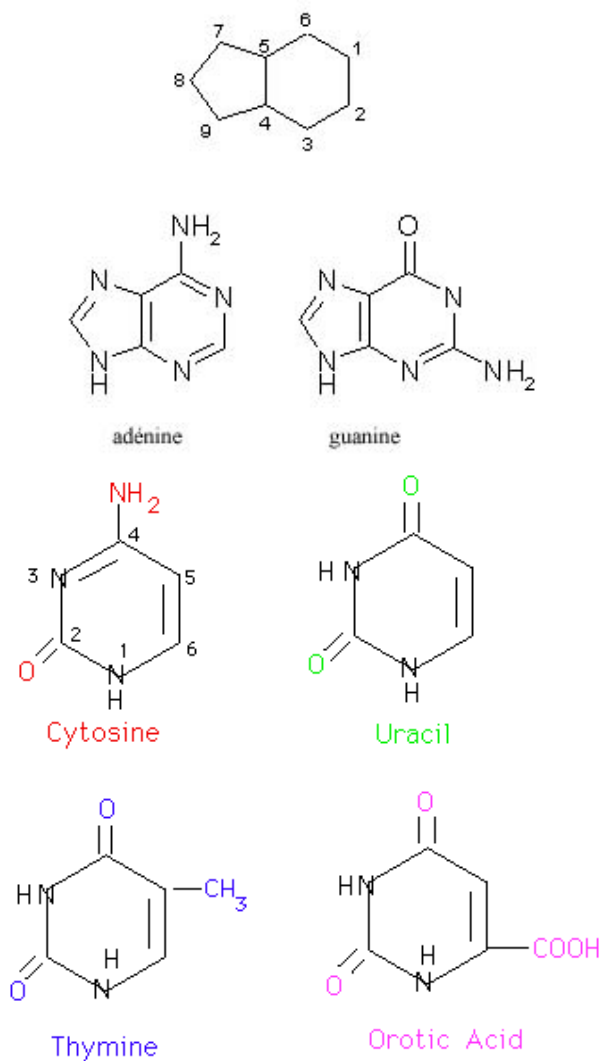
Objective

The objective of this lesson is to know the different levels of structural organization of RNA

3.4.4.1 INTRODUCTION

Nucleic acids were discovered in 1869 by *Johann Friedrich Miescher* (1844-1895), who called the material 'nuclein' since it was found in the nucleus. It was later discovered that prokaryotic cells, which do not have a nucleus, also contain nucleic acids. Nucleic acid biopolymers comprise the DNA and RNA molecules. The RNA world hypothesis proposes that the earliest forms of life relied on RNA both to carry genetic information (like DNA does now) and to catalyze biochemical reactions like an enzyme. According to this hypothesis, descendants of these early lifeforms gradually integrated DNA and proteins. In later forms the two types of molecules possess very different functional roles. In brief, DNA molecules (deoxyribonucleic acids) contain the genetic code, whereas the more versatile RNA molecules (ribonucleic acids) are involved in almost all crucial life processes and especially in the translation of the genetic code into proteins. The four ribonucleosides that incorporate the Purine bases, adenine (A) and guanine (G), and the pyrimidine bases, uracil (U) and cytosine (C) constitute the basic building blocks of a RNA polymeric chain

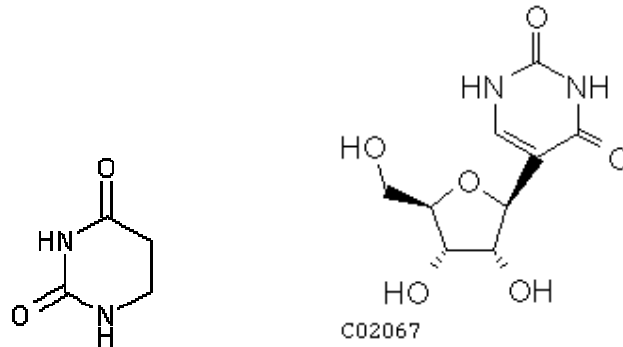
(Figure 1)



The first three are the same as those found in DNA, but uracil replaces thymine as the base complementary to adenine. This base is also a pyrimidine and is very similar to thymine. Uracil is energetically less expensive to produce than thymine, which may account for its use in RNA. In DNA, however, uracil is readily produced by chemical degradation of cytosine, so having thymine as the normal base makes detection and repair of such incipient mutations more efficient. Thus, uracil is appropriate for RNA, where quantity is important but lifespan is not, whereas thymine is appropriate for DNA where maintaining sequence with high fidelity is more critical.

There are also numerous modified bases found in RNA that serve many different roles. Pseudouridine and the DNA base thymidine and are found in various places, but most notably in the TΨC loop of every tRNA. There are nearly 100 other naturally occurring modified bases, many of which are not fully understood.

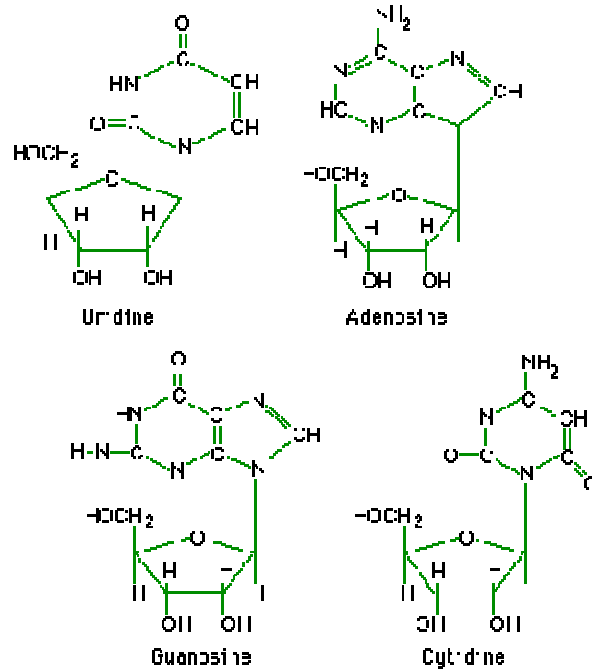
Modified bases fig



Dihydro uracil

Pseudo uridine

The nucleosides comprise a ribose sugar ring and a purine (A, G) or a pyrimidine (C, U) base. They are connected together by a phosphodiester linkage. The nucleoside and its phosphodiester unit are called a nucleotide.



Uracil

Adenosine

Guanosine

Cytidine

The bases of an RNA polymeric chain associate with the complementary bases of the same chain or of another RNA chain by forming purine–pyrimidine A U or G C Watson–Crick base pairs (see Figure 2). The association of two self-complementary strands results in the formation

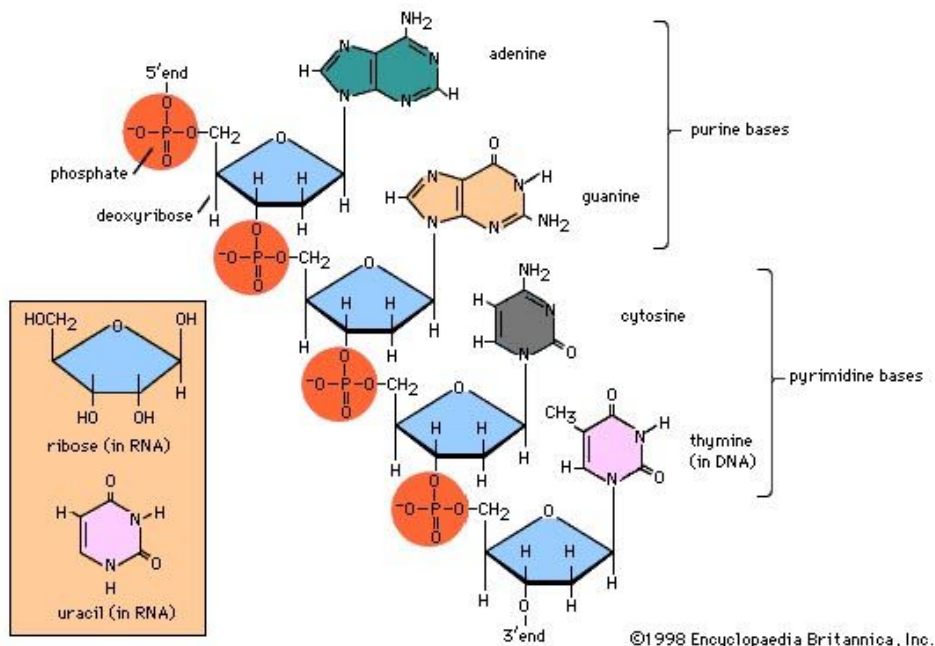
of a right-handed double helical structure. Although DNA and RNA molecules possess very distinct biological functions, chemically RNA differs from DNA only in two aspects: (a) the absence of a methyl group at position 5 of the uridine (U), and (b) the presence of a 2-hydroxyl group of the RNA ribose sugar. These two small chemical modifications account for the profound functional and structural differences observed between DNA and RNA molecules. In contrast to DNA but similar to proteins, single-stranded polynucleotide RNA chains can fold in a variety of complex 3D structures. This ability to form complex folds is exemplified by tRNA molecules which are constituted by a single chain of about 70

nucleotides. The analogy with proteins includes also the fact that some RNA molecules, called ribozymes, are able to perform biologically crucial catalytic reactions.

3.4.4.2 RNA structural organization

1) Primary structure

The first level of organization is thus the sequence of bases attached to the sugar–phosphate backbone.



2) RNA secondary structures

Primary structure is the seq of bases listed from the 5'to the 3' end of the molecule there are a few def of second structure but the simplest is this the structure lies flat on a sheet of paper with no strands overlapping. Tertiary structure req these sec structure to interact out of the plane finally qur structure are interactions b'n two separate molecules(RNA-RNA or RNA-protein)

Types of RNA Secondary structures

Secondary structural elements come in 5 types

Helices

Bulges

Internal loops

Hairpin loops

Multibranched loops or junctions

Pseudonots can be secondary or tertiary interactions

Helical or base paired regions contribute most of the stability to RNA sec structure through H bonding and base stacking. The Watson-Crick pairs, G-C and A-U, as well as some mismatches, such as G-U, stabilize a helix. Base stacking is a such an important stabilizing effect that a single base stacking on the 3¹ side of a helix (called a dangling end) can act as much stability to the structure as a base pair. Helices are on average 6 bp long in sec structure of small subunit rRNA a typical representation of helices in secondary structure is a ladder, where the rungs represent basepairs & the sides represent the sugar-phosphate backbone.

Internal loops form when there are bases that cannot pair on both sides of a helix. Internal loops can be either symmetric (the same number of unpaired bases on each side of the helix) or asymmetric(a different number of unpaired bases on each side of the helix). Two base internal loops are often called mismatches. Originally, studies of homopolymer loops suggested that these regions would destabilize secondary structure; recent studies on common small internal loops reveal increased stability due to base stacking and non Watson-Crick hydrogen bonding For eg , the loop E region of *Xenopus laevis* 5s rRNA has no unusual structures with many non Watson-Crick interactions. Internal loops are imr sites of RNA-Protein interaction in5srRNA, and proposed RNA-RNA tertiary and quarternary interaction in group 1 introns.

Bulges are regions having unpaired bases on only one side of a helix: they can bend RNA back bones. Bulges are important recognition sites for many regulatory and structural proteins. A single bulged "A" is a common motif in 5 s rRNA and is the guanosin binding site of group one intron ribozymes (a catalytic RNA) .an unusal 2¹-5¹ prime linkage at a bulged A is a product of mRNA intron splicing by group II ribozymes, as well as the cellular splicing machinery (small nuclear ribonucleo protein particles or sn RNPs)

Hair pin loops occur at the and of a helix when sugar phosphate

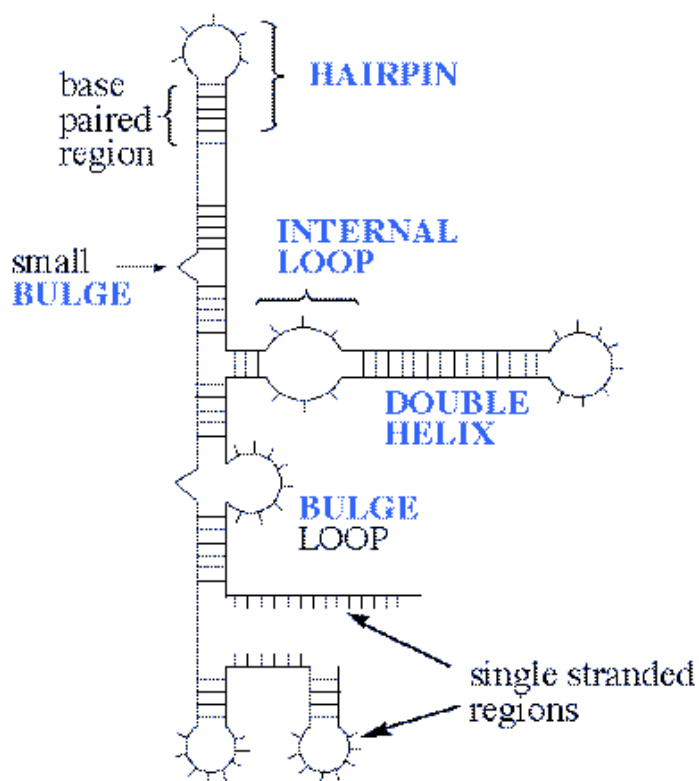
backbone folds back on it self to form, an open loop: tracing the back

bone reveals a hair pin like structure. Comparisons of small sub unit

rRNAs structures reveal an uneven distribution of hairpin loop sizes: four base loops are the most common. Many four base hair pin loops are either GGNRACor CUNCGG (loop bases underlined; N is any base, and r is either G or A) on though other sequences are possible. These special sequence tetra loop hair pins often

substitute for one another in rRNAs. Recent studies on the CUUCGG hair pin reveal an additional strained base pairs between the first and fourth bases in the loop, and an additional intra loop hydrogen bonds. Larger hairpin loops can pair into complex structures involving non Watson-Crick interactions. Hair pin loops are important for snRNA stability, RNA tertiary interactions, and protein binding sites.

Multi branched loops are junctions occur when 3 or more helices joined to form a closed loop. The crystal structure of tRNA has a four helix multi branched loop stabilized by helix-helix stacking as well as significant non Watson-Crick secondary and tertiary interactions. These interactions probably stabilize other multi branched loops. Three-helix multi branched loops are found in 5s rRNA and the hammerhead ribozyme; the former is recognized by TFIIA (Transcription factor A for polymerase III) ribosomal protein, and the latter is the site of phosphor diester cleavage.



Pseudoknots are special RNA interactions, they result from additional pairing in loop regions of the afore mentioned secondary structure. Pseudoknots can form either secondary structure (such as local folding back on a hairpin structure) or tertiary spanning several helical regions. Pseudoknots are

important structural features in the tRNA-like ends of viroids, 16s tRNA of *E.coli*, and the catalytic core of the group I introns. They are important recognition sites for regulatory proteins, and they cause translational frame shifting in many viruses.

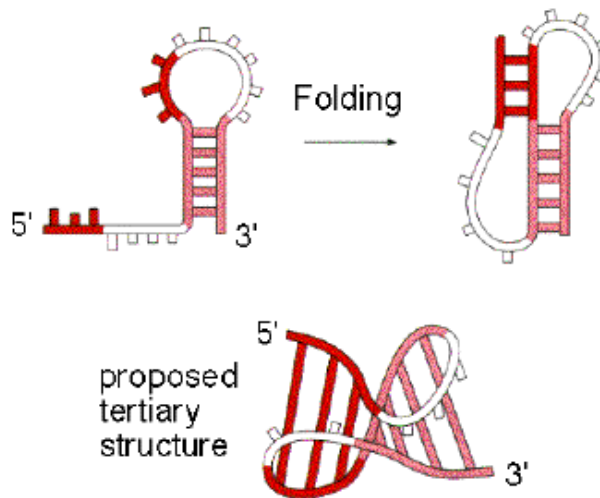
3) TERTIARY STRUCTURE:

In the next level of organization, the tertiary structure, the secondary structure elements are associated through numerous van der Waals contacts, specific hydrogen bonds via the formation of a small number of additional . Involving hairpin loops or internal bulges. The passing of energy levels between secondary and tertiary structures is reasonable

in large RNAs, considering the relative energies and the clear identification of the secondary structure elements. In some cases, it is even possible to cut RNA molecules into modular domains which can re-associate only through tertiary contacts.

RNA Structure: THE PSEUDOKNOT

5' GCG AUUUCUG ACCGC UUUUU UGUCAG 3'



4 RNA TERTIARY MOTIFS

RNA tertiary structure comprises those interactions involving (a) two helices, (b) two unpaired regions, or (c) one unpaired region and a double-stranded helix. The interactions between two helices are basically of two types: either two helices with a contiguous strand stack Watson-Crick pairs and/or unusual pairs

3.4.4.3 TYPES OF RNA

The 4 types of RNA

- tRNA (transfer RNA)
- mRNA
- rRNA
- snRNA

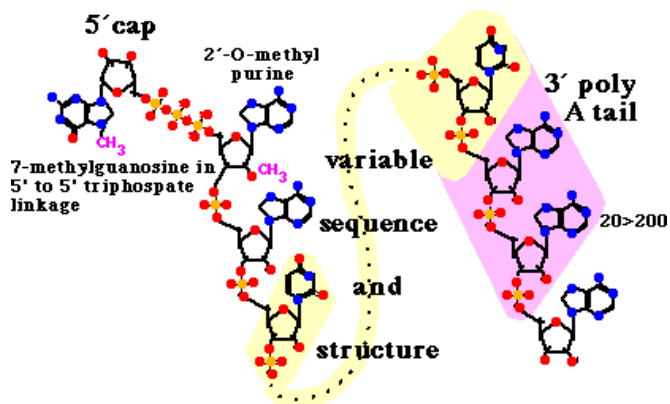
mRNA

Messenger or mRNA is a copy of the information carried by a gene on the DNA. The role of mRNA is to move the information contained in DNA to the translation machinery.

mRNA is heterogeneous in size and sequence. It always has a 5' cap composed of a 5' to 5' triphosphate linkage between two modified nucleotides: a 7-methylguanosine and a 2'-O-methyl purine. This cap serves to identify this RNA molecule as an mRNA to the translational machinery. In addition, most mRNA molecules contain a poly-Adenosine tail at the 3' end. Both the 5' cap and the 3' tail are added after the RNA is transcribed and contribute to the stability of the mRNA in the cell.

mRNA is not made directly in a eukaryotic cell. It is transcribed as heterogeneous nuclear RNA (hnRNA) in the nucleus. hnRNA contains introns and exons. The introns are removed by RNA splicing leaving the exons, which contain the information, joined together. In some cases, individual nucleotides can be added in the middle of the mRNA sequence by a process called RNA editing. In the figure the exons are represented as the region of variable sequence.

hnRNA and mRNA are never found free in the cell. Like DNA, they are bound by cations and proteins. These complexes are termed ribonucleoproteins or RNPs. The variability in sequence and structure means that no structure has been determined for a mRNA.



tRNA

tRNA is the information adapter molecule. It is the direct interface between amino-acid sequence of a protein and the information in DNA. Therefore it decodes the information in DNA. There are > 20 different tRNA molecules. All have between 75-95 nt.

All tRNA's from all organisms have a similar structure, indeed a human tRNA can function in yeast cells.

There are 4 arms and 3 loops. The acceptor, D, T pseudouridine C and anticodon arms, and D, T pseudouridine C and anticodon loops. Sometimes tRNA molecules have an extra or variable loop (this is shown in yellow in the adjacent figure).

tRNA is synthesized in two parts. The body of the tRNA is transcribed from a tRNA gene. The acceptor stem is the same for all tRNA molecules and is added after the body is synthesized. It is replaced often during lifetime of a tRNA molecule.

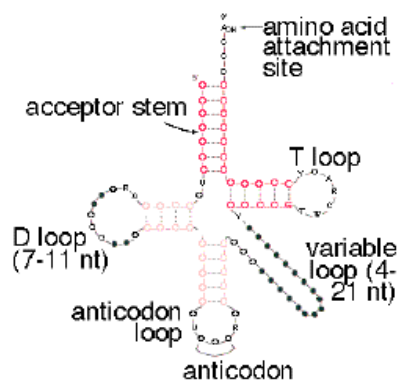
The adjacent image is a 3-D model of a yeast tRNA molecule which can code for ser. The model and the schematic above share the same color coding. You can rotate the molecule in the y axis to get better views of the structure.

Observe how the molecule is folded with the D and T pseudo-U C loops in contact, and with the acceptor stem and the anticodon loop at opposite ends.

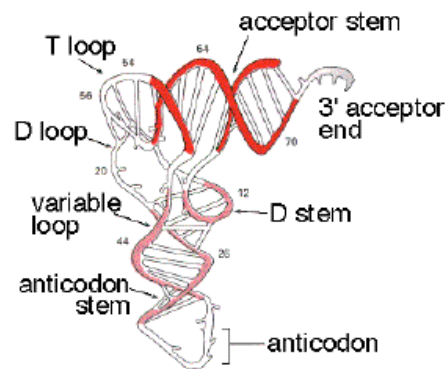
The acceptor stem is the site at which a specific amino acid is attached by an aminoacyl-tRNA synthase. The anticodon reads the information in a mRNA sequence by base pairing.

Notice how the overall gross structure of the helix resembles that of DNA. Observe that the phosphoryl groups (shown in orange) are not on the outside of the helix like they are in DNA but are located in the groove. bases are paired similarly to DNA. In this image the acceptor stem is on the left and the anticodon loop is at the bottom. The D loop is in front of the T pseudoU C loop at the top right.

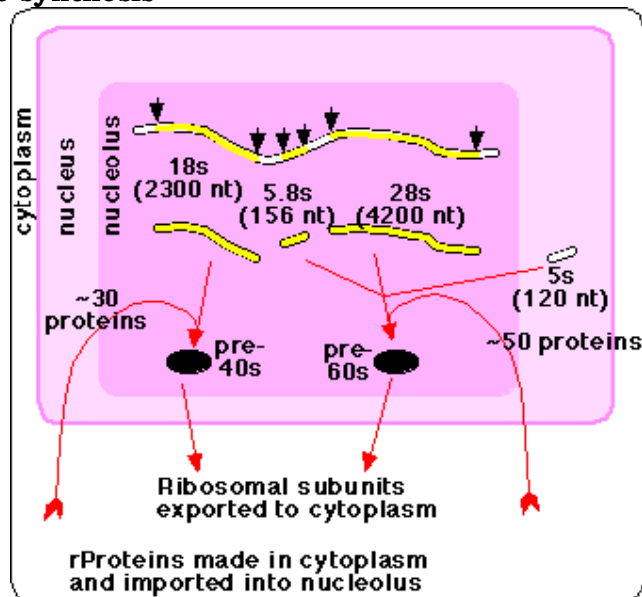
tRNA structure



**Cloverleaf 2^o
structure**



**L-shaped 3^o
structure**

rRNA and ribosome synthesis

Ribosomal RNA (rRNA) is a component of the ribosomes, the protein synthetic factories in the cell. Eukaryotic ribosomes contain four different rRNA molecules: 18 s, 5.8 s, 28 s, and 5 s rRNA. Three of the rRNA molecules are synthesized in the nucleolus, and one is synthesized elsewhere. rRNA molecules are extremely abundant. They make up at least 80% of the RNA molecules found in a typical eukaryotic cell.

Synthesis of the three nucleolar rRNA molecules is unusual because they are made on one primary transcript that is chopped up into three mature rRNA molecules. These rRNA molecules and the 5 s rRNA combine with the ribosomal proteins in the nucleolus to form pre 40 s and pre 60 s ribosomal subunits. These pre-subunits are exported to the nucleus where they mature and assume their role in protein synthesis.

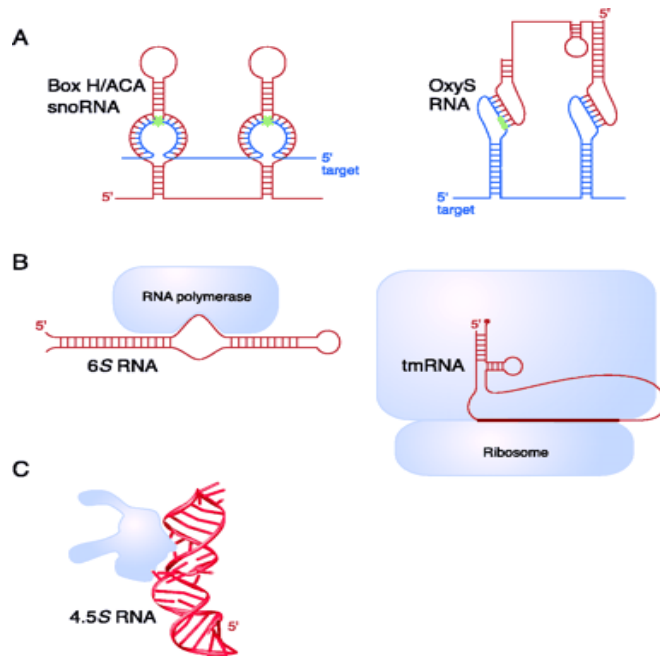
The rRNA molecules have several roles in protein synthesis. First, the 28 s rRNA has a catalytic role, it forms part of the peptidyl transferrase activity of the 60 s subunit. Second, 18s rRNA has a recognition role, involved in correct positioning of the mRNA and the peptidyl tRNA. Finally, the rRNA molecules have a structural role. They fold into three-dimensional shapes that form the scaffold on which the ribosomal proteins assemble. The model on the left shows a the three dimensional structure that the 5 s rRNA from the African frog, *Xenopus laevis* is thought to adopt.

Non-coding RNAs (ncRNA)

It has recently become clear that both prokaryotic and eukaryotic cells contain a large number of small RNA molecules with a diversity of functions in processes such as regulation of transcription, replication of eukaryotic chromosomes, RNA processing, RNA modification, RNA editing, mRNA stability and degradation, regulation of translation, and protein translocation.

The following diagram from the same review article illustrates some examples of ncRNAs in action. The top row shows two ncRNA molecules (red) interacting with target mRNA molecules. On the left, the interaction exposes bases that are modified. On the right the interaction covers up the ribosome binding site and prevents translation of the target

mRNA. The middle row shows two ncRNAs that mimic RNA structures found in the cell. On the left is a structure that mimics that of an RNA-polymerase open promoter complex; on the right is one that mimics both tRNA and mRNA (tmRNA). The bottom row shows the RNA-signal recognition particle complex which functions in directing protein synthesis through the endoplasmic reticulum.



Some specific examples of ncRNAs are:

telomere RNA

Telomerase, the enzyme that adds the telomere repeats to eukaryotic chromosomes contains an essential RNA template.

snRNAs

Small nuclear RNA (snRNA) is the name used to refer to a number of small RNA molecules found in the nucleus. These RNA molecules are important in a number of processes including RNA splicing (removal of the introns from hnRNA) and maintenance of the telomeres, or chromosome ends. They are always found associated with specific proteins and the complexes are referred to as small nuclear ribonucleoproteins (SNRNP) or sometimes as snurps.

Antibodies against snurps are found in a number of autoimmune diseases

snoRNA

small nucleolar RNA molecules are found in the nucleolus of eukaryotic cells. They are associated with protein particles, snoRNPs, and they have been demonstrated to define sites of nucleotide modifications in rRNA. In addition, a few snoRNAs may play a role in of pre-rRNA processing in the nucleolus.

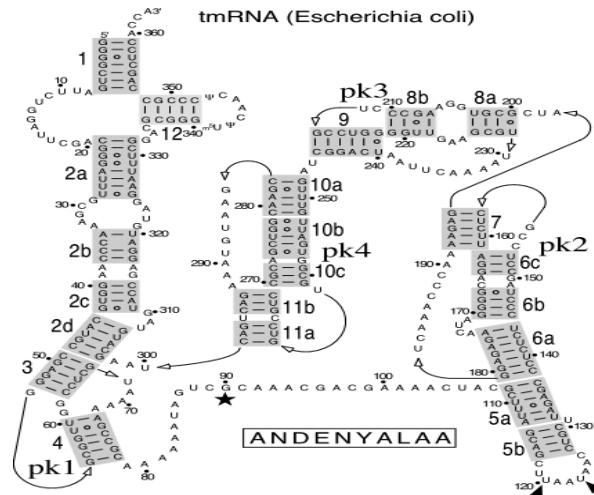
gRNA

guide RNA functions in the editing of certain mRNAs. RNA editing is found particularly in the mitochondria of plants and protozoa and also in chloroplasts. gRNA directs where and what changes can occur.

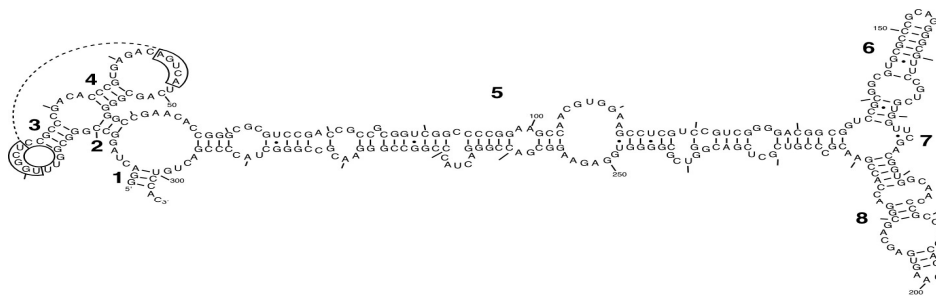
tmRNA

tmRNA has properties of tRNA and mRNA combined in a single molecule. It functions during protein synthesis to rescue ribosomes that have become "stuck" while translating mRNA molecules that have lost their stop codons.

I

**signal recognition particle RNA**

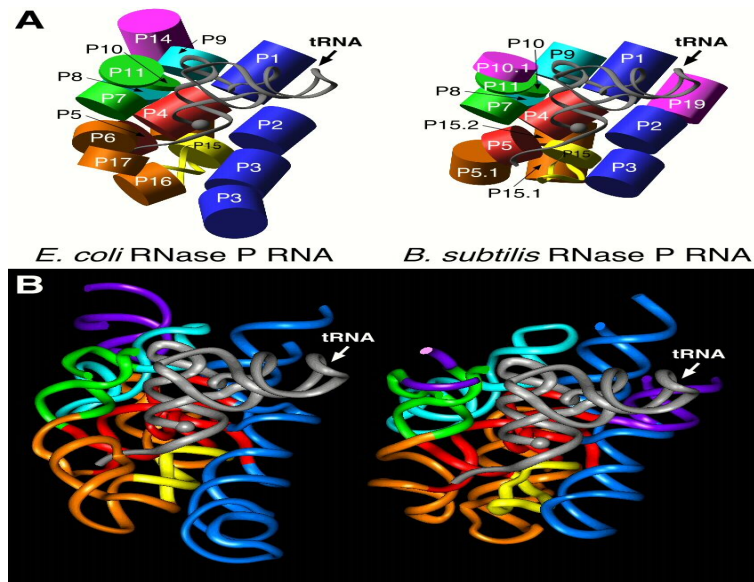
This 300 nt RNA is an integral component of **Signal Recognition Particles** which direct the secretion of newly-synthesized proteins through the endoplasmic reticulum.



Halobacterium halobium SRP RNA
(SRPDB, March 10, 2000)

M1 RNA

M1 RNA is the name given to the RNA component of **Ribonuclease P**, which functions in the processing of tRNA molecules in prokaryotes. M1 RNA is the catalytic component of the enzyme.



Summary:

RNA is ubiquitous in the cell and is important for many processes. The activity of RNA is determined by its structure, the way it is folded back on itself. Secondary structure modeling of RNA predicts, or otherwise determines, the pattern of Watson-Crick (WC), wobble and other, non-canonical pairings that occur when the RNA is folded. By the folding of RNA different secondary structures are formed. Further association of secondary structures results in tertiary structure. There are many different kinds of RNA. Ribosomal RNA (rRNA) is a crucial part of the ribosome which is found in all living cells and in organelles such as mitochondria and chloroplasts. Small nuclear RNAs (snRNA) form a vital part of spliceosomes that process mRNAs in eukaryotes. These are 2 examples of *structural* RNAs.

Model questions

- 1) Discuss in detail about various structural organizations of RNA?
- 2) Write about different types of RNAs?

References

Molecular biology and biotechnology Robert A. Meyers
Genes by Lewin

Author
K. Haritha
Lecturer
Centre for Biotechnology
Acharya Nagarguna University

Lesson 3.5.1

MOLECULAR MODELING AND SIMULATION STUDIES

- 3.5.1.1 Objective**
- 3.5.1.1.1 Elements of Molecular Modeling**
- 3.5.1.1.2 Components**
- 3.5.1.2 Force Fields**
- 3.5.1.3 Methods of Molecular modeling**
- 3.5.1.2.1 Energy due to Bond Stretching**
- 3.5.1.2.2 Energy due to Bond Angle Bending**
- 3.5.1.5 Self Assessment Questions**
- 3.5.1.6 Reference book:**
- 3.5.1.4 Summary**

3.5.1.1.1 Elements Of Molecular Modeling

Molecular modeling, also known as molecular mechanics, is a method to calculate the structure and energy of molecules based on nuclear motions. Electrons are not considered explicitly, but rather it is assumed that they will find their optimum distribution once the positions of the nuclei are known. This assumption is based on the Born-Oppenheimer approximation of the Schrödinger equation. The Born-Oppenheimer approximation states that nuclei are much heavier and move much more slowly than electrons. Thus, nuclear motions, vibrations and rotations can be studied separately from electrons; the electrons are assumed to move fast enough to adjust to any movement of the nuclei. In a very crude sense molecular modeling treats a molecule as a collection of weights connected with springs, where the weights represent the nuclei and the springs represent the bonds.

3.5.1.1.2 Components



Fig1 (9.3 pg 351 Bioinformatics Computing pearsons)

Every modeling and simulation system is composed of a model, a database, a simulation engine, and a visualization engine. The user and some form of feedback device, such as computer monitor, are normally considered key elements as well. The components of a simulation system vary typically in form, complexity and completeness (Fig 1).

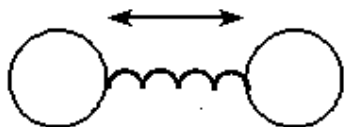
3.5.1.2 Force Fields

A *force field* is used to calculate the energy and geometry of a molecule. It is a collection of atom types (to define the atoms in a molecule), parameters (for bond lengths, bond angles, etc.) and equations (to calculate the energy of a molecule). In a force field a given element may have several atom types. For example, ethylbenzene contains both sp^3 -hybridized carbons and aromatic carbons. sp^3 -Hybridized carbons have a tetrahedral bonding geometry, while aromatic carbons have a trigonal bonding geometry. The C-C bond in the ethyl group differs from a C-C bond in the phenyl ring, and the C-C bond between the phenyl ring and the ethyl group differs from all other C-C bonds in ethylbenzene. The force field contains parameters for these different types of bonds. Some of these parameters are given below. The total energy of a molecule is divided into several parts called force potentials, or potential energy equations. Force potentials are calculated independently, and summed to give the total energy of the molecule. Examples of force potentials are the equations for the energies associated with bond stretching, bond bending, torsional strain and van der Waals interactions. These equations define the potential energy surface of a molecule.

$$E_{\text{TOTAL}} = E_{\text{STRETCH}} + E_{\text{BEND}} + E_{\text{S-B}} + E_{\text{TORSION}} + E_{\text{vdW}} + E_{\text{DP-DP}}$$

3.5.1.2.1 Energy due to Bond Stretching

Whenever a bond is compressed or stretched the energy goes up. The energy potential for bond stretching and compressing is described by an equation similar to Hooke's law for a spring, except a cubic term is added. This cubic term helps to keep the energy from rising too sharply as the bond is stretched.



$$E_s = 143.88 \frac{k_s}{2} (l - l_0)^2 (1 - 2(l - l_0))$$

k_s is the force constant in mdyne/Å

l_0 is the natural bond length in Å

l is the actual bond length in Å

143.88 converts the units to kcal/mol

For a typical alkane C-C bond $k_s = 4.4$ mdyne/Å and $l_0 = 1.523$ Å

3.5.1.2.2 Energy due to Bond Angle Bending

As angles are bent from their norm the energy increases. The potential function below works very well for bends of up to about 10 degrees. To handle special cases, such as cyclobutane, special atom types and parameters are used in the force field.



$$E_{\theta} = 0.21914k_{\theta}(\theta - \theta_0)^2 (1 + 7 \times 10^{-6}(\theta - \theta_0)^4)$$

k_{θ} is the force constant in mdyn/(Å rad²)

θ_0 is the natural bond angle in degrees

θ is the actual bond angle in degrees

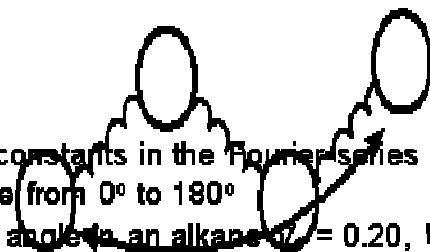
l_h and l are as above

0.21914 is the conversion factor

For an alkane C-C-C bond angle k_{θ} is 0.45 mdyn/(Å rad²) and θ is 109.5°

3.5.1.2.3 Energy due to Torsional Strain

Intramolecular rotations (rotations about torsion or dihedral angles) require energy. For example, it takes energy for cyclohexane to go from the chair conformation to the boat conformation. The torsion potential is a Fourier series that accounts for all 1-4 through-bond relationships.



V_1, V_2, V_3 are force constants in the Fourier series in kcal/mol

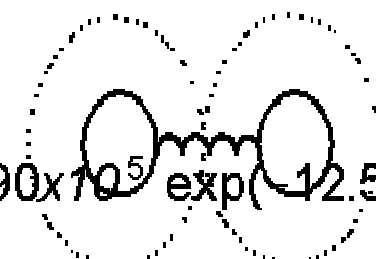
ω is the torsion angle from 0° to 180°

For a typical torsion angle in an alkane $V_1 = 0.20, V_2 = 0.27, V_3 = 0.093$

$$E_{\text{tor}} = \frac{V_1}{2}(1 + \cos\omega) + \frac{V_2}{2}(1 + \cos 2\omega) + \frac{V_3}{2}(1 + \cos 3\omega)$$

3.5.1.2.4 Energy due to van der Waals Interactions

The van der Waals radius of an atom is its effective size. As two non-bonded atoms are brought together the van der Waals attraction between them increases (a decrease in energy). When the distance between them equals the sum of the van der Waals radii the attraction is at a maximum. If the atoms are brought still closer together there is strong van der Waals repulsion (a sharp increase in energy).



$$E_{vdW} = \epsilon \left[2.90 \times 10^5 \exp\left(-12.50 \frac{r_0}{r_v}\right) - 2.25 \left(\frac{r_v}{r_0}\right)^6 \right]$$

ϵ is the energy parameter which sets the depth of the potential energy well

r_v is the sum of the van der Waals radii of the interacting atoms

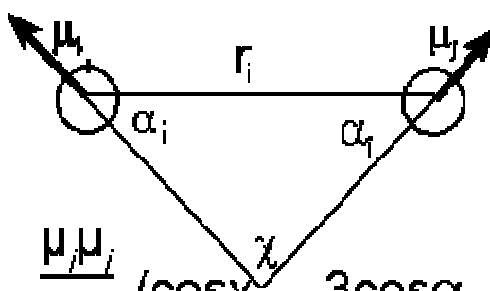
r_0 is the distance between the interacting centers

For carbon-carbon interactions ϵ is 0.044

For carbon-hydrogen interactions ϵ is 0.046

3.5.1.2.5 Energy due to Dipole-Dipole Interactions

In some force fields electrostatic interactions are accounted for by atomic point charges. In other force fields, such as MM2 and MMX, bond dipole moments are used to represent electrostatic contributions. One can readily see that the equation below stems from Coulomb's law. The energy is calculated by considering all dipole-dipole interactions in a molecule. If the molecule has a net charge (e.g., NH_4^+), charge-charge and charge-dipole calculations must also be carried out.



$$E_{dipole} = \frac{\mu_i \mu_j}{D(r_{ij})^3} (\cos \chi - 3 \cos \alpha_i \times \cos \alpha_j)$$

D is the dielectric constant of the solvent

χ is the angle between two the dipoles μ_i and μ_j

α_i α_j are the angles between the dipoles and a vector connecting the two dipoles

r_{ij} is the distance between the dipoles

3.5.1.3 Methods of Molecular modeling

Scientific study can be represented as manipulations of some observables (inputs) and measuring or calculating the system response in terms of values of some other variables (outputs). *Experimental studies* measure the fundamental properties of *real* systems. They provide basic information about all aspects of the Universe. *Scientific theory* tries to establish connections between the inputs and outputs which have been observed experimentally; sophisticated theories use an underlying state to connect outputs to inputs. The goal is of understanding, explanation, prediction. Real systems are usually too complex to study theoretically - we are unable to reliably calculate the outputs from the inputs. That is why we construct and study *models* - simplified representations of a system. Some models can be analyzed *analytically* - e.g. ideal gas, harmonic oscillator, hydrogen atom. Many models can only be studied *computationally or numerically* - e.g. liquids, glasses or many-electron systems in quantum mechanics. Finally, models of least detail for most complex systems use only *phenomenological* description - e.g. QSAR, protein structure prediction, medicine.

Simulations are intermediate between theory and experiment, and involve observation of connection between inputs and outputs for complex model systems. Simulations are not experiments, because they deal with models, and not with real systems. The information obtained from simulations is used to give insight and understanding of complex systems, to test existing theories and develop new ones.

In *molecular modeling* we encounter all types and sorts of models:

- structural (graphical, ball-and-stick, wire, ...)
- phenomenological (homology, secondary structure prediction, ...)
- mathematical (computer simulations)

The most general and detailed mathematical models use potential energy functions to describe the states of the model.

3.5.1.3.1 Quantum Mechanics

Quantum mechanics is one of the oldest mathematical formalisms of theoretical chemistry. In its purest form, quantum theory uses well known physical constants such as the velocity of light, values for the masses and charges of nuclear particles and differential equations to directly calculate molecular properties and geometries. This formalism is referred to as *ab initio* (from first principles) quantum mechanics.

The equation from which molecular properties can be derived is the Schrodinger equation,

$$\hat{H}\Psi = E\Psi$$

where E is energy of the system relative to one in which all atomic particles are separated to infinite distances, Ψ is the wavefunction which defines the Cartesian and spin coordinates of the atomic particles and H is the Hamiltonian operator which includes terms for both potential and kinetic energy. Unfortunately, the Schrodinger equation can be solved only for very small molecules such as hydrogen and helium. Approximations must be introduced in order to extend the utility of the method to polyatomic systems.

The first approximation attempts to differentiate nuclei and electrons. It assumes that nuclei are much heavier than electrons and move much more slowly so that molecular systems can be viewed as electrons moving in a field of fixed nuclei (the Born-Oppenheimer approximation). Solutions to the Schrodinger equation using this assumption lead to values of effective electronic energy which are dependent on relative nuclear coordinates. As the nuclei are moved to new coordinates and molecular energies are re-calculated, a quantitative description of molecular energy is derived. This description, which relates energy to geometry, is referred to as the potential energy surface for the molecule. The lowest point on this surface, with respect to energy, is the ground state energy (and its associated geometry) for the molecule.

The second approximation allows the wavefunction Ψ to be represented as the product of one-electron (or spin) orbitals. The functions that are used to describe these orbitals are referred to as basis functions. This formalism is referred to as the Linear Combination of Atomic Orbitals (LCAO) theory. Once the orbitals have been derived, the orbital coefficients (which define the energy of the system) are calculated. Hartree-Fock theory is used to accomplish this goal.

Hartree-Fock assumes that the energy of a set of molecular orbitals can be derived from the basis set functions which are used to define each orbital and a set of adjustable coefficients which are used to minimize the energy of the system. The energy calculation becomes an exercise in solving a set of $(N \times N)$ matrices to obtain optimal values for the orbital coefficients. Since this calculation requires a value for the coefficients in order to solve the equations, an iterative process is used in which an initial guess for the value of the coefficients is progressively refined until it provides consistent values. This method is referred to as the self-consistent-field (SCF) theory.

Quantum mechanics utilizes a set of mathematical descriptions to define a theoretical model for the behavior of molecules. The validity of these models can be gauged by comparing structures and properties derived from the model with experimental results. In general, *ab initio* methods are able to reproduce laboratory measurements for properties such as the heat of formation, ionization potential, UV/Visible spectra and molecular geometry.

Since quantum methods utilize the principles of particle physics to examine structure as a function of electron distribution, their use can be extended to the analysis of molecules as yet unsynthesized and chemical species which are difficult (or impossible) to isolate. Geometries and properties for transition states (where the electronic character of component atoms is shifting from that found in the starting material to that of the products) and excited states (where the electronic configuration of the molecule is temporarily perturbed by adding energy to the molecule) can only be calculated using quantum methods.

Ab initio quantum methods compute a number of solutions to a large number of equations. While recent publications have reported calculations on large molecules, the methods are generally limited to compounds containing between ten and twenty atoms due to the amount of computer time required for each calculation and the large amount of disk space needed to store intermediate data files. Physical/theoretical chemists have developed alternative approaches to computing structures and properties by simplifying

portions of the calculation to circumvent these limitations. These methods are referred to collectively as semiempirical quantum methods.

Semiempirical methods utilize approaches which are similar to *ab initio* methods, but several approximations are introduced to simplify the calculations. Rather than performing a full analysis on all electrons within the molecule, some electron interactions are ignored. These methods include the Huckel approach for aromatic compounds (in which the outer electrons in conjugated systems are treated, but the inner (or core) electrons are ignored) and the Neglect of Differential Overlap formalisms found in the CNDO (Complete Neglect of Differential Overlap) and INDO (Intermediate Neglect of Differential Overlap). In these methods, the more complex portions of the *ab initio* calculation are ignored or set to zero. Other semiempirical approaches replace complex portions of the calculation with parameters which are derived from experimental data.

While semiempirical methods require less computer resources than *ab initio* methods, they are still compute intensive. In general, calculations are routinely performed on compounds which contain up to 100 atoms. The chief drawback of the method is that its application is limited to systems for which appropriate parameters have been developed.

Application of quantum techniques in drug design are detailed in many of the primary journals. Recently, the computational group at Sandoz has reported the use of Gaussian-90 (an *ab initio* program) in the analysis of the electronic and tautomeric factors for a model of H2 receptor agonists. Scientists at The University of San Luis have examined conformational energy maps and electrostatic potentials derived from semiempirical calculations to differentiate between H2 receptor antagonists and H3 receptor agonists.

Quantum methods can also be used to examine the forces which determine protein and peptide stability. Sligar and Robinson have reported preparation of the first system which permits measurement of electrostatic effects on the stabilization of helical proteins. D.R. Ripoll and co-workers have utilized calculations from DELPHI (quantum calculations based on density functional theory) to develop a proposed mechanism for capture of substrate molecules by acetylcholinesterase. Workers at the University of Nancy have used high-level quantum calculations to define six new conformations for peptides. All of the proposed conformations were found in proteins whose structures were solved using X-ray crystallography.

Experimental chemists routinely work with compounds which range in size from several hundred atoms (drug candidates, monomers, agricultural chemicals, etc.) to several thousand atoms (proteins, nucleic acids, carbohydrates, polymers, etc.). The computational requirements for quantum mechanical approaches render these methods unusable for routine analysis of these types of compounds. Thus, a further simplification in the way molecular geometries and their associated properties are computed is required. This approach is the molecular mechanics or force field method.

3.5.1.3.2 Molecular mechanics is a mathematical formalism

Molecular mechanics attempts to reproduce molecular geometries, energies and other features by adjusting bond lengths, bond angles and torsion angles to equilibrium

values that are dependent on the hybridization of an atom and its bonding scheme (this atom description is referred to as the atom type). Rather than utilizing quantum physics, the method relies on the laws of classical Newtonian physics and experimentally derived parameters to calculate geometry as a function of steric energy. The general form of the force field equation is

$$E_{\text{pot}} = \sum E_{\text{bond}} + \sum E_{\text{ang}} + \sum E_{\text{tor}} + \sum E_{\text{oop}} + \sum E_{\text{nb}} + \sum E_{\text{el}}$$

E_{pot} is the total steric energy which is defined as the difference in energy between a real molecule and an ideal molecule. E_{bond} , the energy resulting from deforming a bond length from its natural value, is calculated using Hooke's equation for the deformation of a spring ($E = 1/2 K_b(b - b_0)^2$ where K_b is the force constant for the bond, b_0 is the equilibrium bond length and b is the current bond length). E_{ang} , the energy resulting from deforming a bond angle from its natural value, is also calculated from Hooke's Law. E_{tor} is the energy which results from deforming the torsion or dihedral angle. E_{oop} is the out-of-plane bending component of the steric energy. E_{nb} is the energy arising from non-bonded interactions and E_{el} is the energy arising from coulombic forces.

When the terms shown in the general form of the force field are expanded, the equation becomes

$$E_{\text{pot}} = \sum 1/2 K_b (b - b_0)^2 + \sum 1/2 K_\theta (\theta - \theta_0)^2 + \\ \sum 1/2 K_\phi (1 + \cos N\phi)^2 + \sum 1/2 K_\chi (\chi - \chi_0)^2 \\ \sum (B/r)^{12} - (A/r)^6 + \sum (qq/r)$$

The manner in which these terms are utilized to build a model is referred to as the functional form of the force field. The force constants

$$k_b \quad k_\theta \quad k_\phi \quad k_\chi$$

and equilibrium values

$$b_0 \quad \theta_0 \quad \phi_0 \quad \chi_0$$

are atomic parameters which are experimentally derived from X-ray, NMR, IR, microwave, Raman spectroscopy and *ab initio* calculations on a given class of molecules (alkanes, alcohols, etc). The energy of the atoms in a molecule is calculated and minimized using a variety of directional derivative techniques.

In contrast to *ab initio* methods, molecular mechanics is used to compute molecular properties which do not depend on electronic effects. These include geometry, rotational barriers, vibrational spectra, heats of formation and the relative stability of conformers. Since the calculations are fast and efficient, molecular mechanics can be used to examine systems containing thousands of atoms. However, unlike *ab initio* methods, molecular mechanics relies on experimentally derived parameters so that calculations on new molecular structures may be misleading.

Each of the methods described above are used to calculate the energy of a compound in a specific 3D orientation and to optimize the geometry as a function of energy (i.e.,

adjust the coordinates of each of the atoms and recompute the energy of the molecule until a minimum energy is obtained). Coupled with other numerical techniques, they also can be used to simulate the time-dependent behavior of molecules (molecular dynamics) and explore their conformational flexibility (conformational search).

Molecular dynamics combines energy calculations from force field methodology with the laws of Newtonian (as opposed to quantum) mechanics. The simulation is performed by numerically integrating Newton's equations of motion over small time steps (usually 10^{-15} sec or 1 fsec). The simulation is initialized by providing the location and assigning a force vector for each atom in the molecule. The acceleration of each atom is then calculated from the equation $a = F/m$ where m is the mass of the atom and F the negative gradient of the potential energy function (the mathematical description of the potential energy surface). The Verlet algorithm is used to compute the velocities of the atoms from the forces and atom locations. Once the velocities are computed, new atom locations and the temperature of the assembly can be calculated. These values then are used to calculate trajectories, or time dependent locations, for each atom. Over a period of time, these values can be stored on disk and played back after the simulation has completed to produce a "movie" of the dynamic nature of the molecule.

Molecular dynamics simulations have been used in a variety of biomolecular applications. The technique, when combined with data derived from Nuclear Magnetic Resonance (NMR) studies, has been used to derive 3D structures for peptides and small proteins in cases where X-ray crystallography was not practical. Additionally, structural, dynamic and thermodynamic data from molecular dynamics has provided insights into the structure-function relationships, binding affinities, mobility and stability of proteins, nucleic acids and other macromolecules that cannot be obtained from static models.

Paulsen and Ornstein used dynamics to explain substrate orientation and the product profile for the oxidation of thiocamphor bound to cytochrome P450. Karplus and Gelin demonstrated that conformational changes are required in hemoglobin to permit oxygen transport and discussed potential pathways for oxygen migration into the active site. This study is of interest because static models of hemoglobin, some of which are based on X-ray data, do not show sufficient entry space to permit oxygen molecules to diffuse into the active site. Dynamic motion of hinge regions in the protein is necessary to account for activity.

While molecular dynamics provides an excellent approach to searching regions of conformational space, it is not an exhaustive search. The active conformation of a molecule can be missed as the dynamics simulation skips over the hills and valleys of the potential energy surface. Since the active conformation at a receptor may not always be the minimum energy structure (defined as the structure with the 3D geometry that places the molecule at the lowest point on the potential energy hypersurface), it is important to examine all potentially accessible conformations.

For small molecules with a limited number of freely rotating bonds, this can be easily accomplished by driving each torsion angle stepwise over a 360 degree range. As an example, a graph of the conformationally dependent energy (shown along the Y-axis) of the molecule butane is shown in Figure 3.

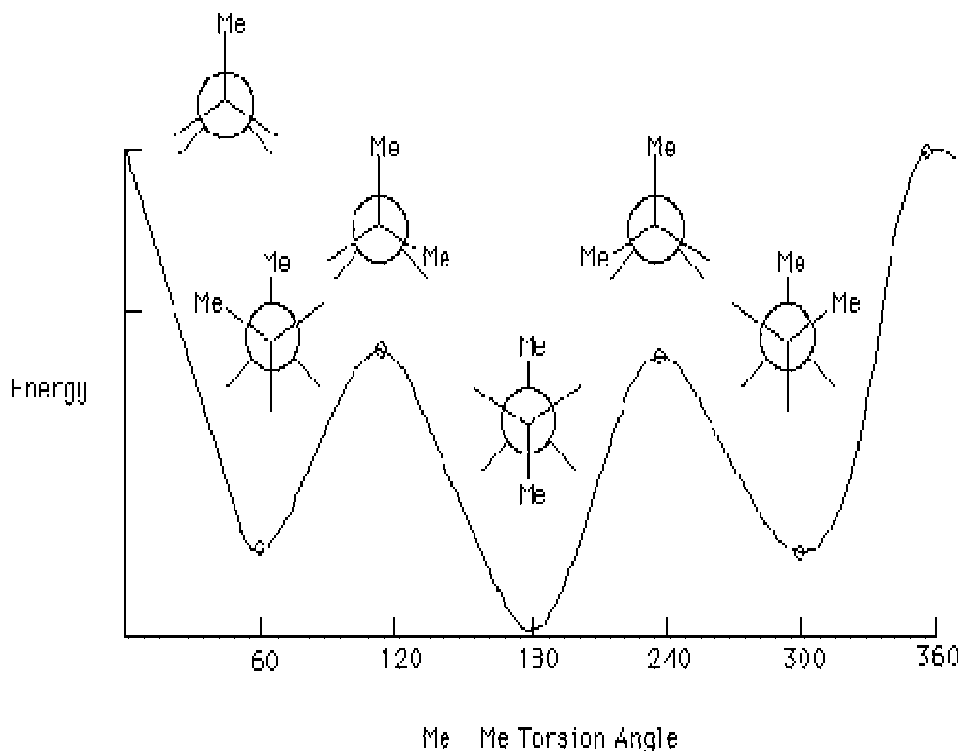


Figure3
Butane Conformers

Unfortunately, the number of conformations for a molecule (defined as the "non-identical arrangements of the atoms in a molecule obtainable by rotation about one or more single bonds") generated in this manner rises exponentially with the number of bonds rotated. The theoretical number of conformations for a molecule can be calculated by the formula.

$$\text{Number of conformers} = (360/\text{angle increment})^{(\# \text{ rotatable bonds})}$$

Thus a molecule with four rotatable bonds searched in a 60 degree increment will generate $(360/60)^4$ or 1,296 distinct conformations. If the energy for each structure is evaluated in one second, the search and evaluation will require approximately 22 minutes. Timing of this order of magnitude is acceptable for small and medium-sized molecules. The approach is not reasonable for proteins and peptides which contain hundreds or thousands of rotatable bonds. Conformational behavior for these types of molecules is examined using random searching techniques such as molecular dynamics, Monte Carlo and distance geometry.

If the size of a molecule limits exhaustive searching approaches, one can redefine the problem by modifying the algorithm, examining a subset of the structure or introducing screens to limit the number of conformations produced by the search. Marshall and co-workers utilize the latter approach to reduce the combinatorial problems of searching large sets of structures. Their method examines each conformation to assure that its steric energy is within a user specified range, that the conformer does not have any two atoms closer together than the sum of their van der

Waals radii and that it obeys user defined distance ranges for specified atoms. In addition, the method uses logical operators on the maps of distances and angles, which were calculated from searches on previous molecules in the series, to constrain searches on new members of the series. When this method is applied to compounds which are active in a given biological assay, the final map of distances defines the 3D orientations of atoms in the molecule which are responsible for activity. This technique has been widely used in medicinal research. A recent report detailed the successful development of a receptor model based on the analysis of 28 angiotensin converting enzyme (ACE) inhibitors.

One of the problems which is common to many of the methods described is the need to define the spatial regions where molecules can orient properly in a given active site. The algorithms used in energy minimization, conformational searching and pharmacophore identification all attempt to find optimal solutions to problems which have several potential solutions. Examples of this issue include the multiple-minima problem in molecular mechanics optimizations and combinatorial problems seen in conformational search. Computational chemistry software developers have recently started to investigate the use of genetic algorithms as an approach to circumvent these problems.

Genetic algorithms attempt to use the rules of natural selection to subset computationally demanding tasks. Rather than run the problem as a linear progression or allocate potential solutions into tree structures, the algorithm randomly builds sets of solutions which are governed by an ongoing process of natural selection. For chemistry applications, the algorithm operates on strings of binary numbers which represent molecular composition, position and/or conformation (defined as the individuals). When individuals are acted upon by genetic operators such as crossover, mutation and selection factors, families of individuals evolve over a period of time. These families then can be prioritized by constraints which define optimal solutions to the problem.

Genetic algorithms have been applied to a variety of chemical problems including NMR studies of proteins and peptides (optimization of distance constraints), conformational search problems (systematic searching of conformational space for large structures) and optimization of the overlap of common pharmacophores in structurally diverse molecules. In each case, the algorithms significantly expanded either the scope of the problem under investigation or increased the speed and efficiency of the process relative to traditional algorithms.

Payne and Glen recently published work exploring the application of genetic algorithms to problems in molecular similarity, conformational search and pharmacophore development, searching receptor antagonists to the putative pharmacophore using distance constraints, fitting peptides to distance constraints derived from NMR and fitting two molecules with radically different structure, but similar activity.

The computational methods are used to optimize molecular geometry and calculate physical and electronic properties. An equally important aspect of CAMD/CADD is the ability to display these properties in a manner which increases the chemist's ability to interpret experimental findings and correlate these finding with structural features. Molecular surfaces play an important role in these studies.

Drug discovery is a more complex problem than it was in the past due, in part, to the fact that the etiology of the diseases which we seek to control are more complex. The amount of data generated in a typical study can easily overwhelm the scientists who are responsible for guiding the study. Computer systems which store, manipulate and display chemical structures and their associated data have therefore become an important and growing tool in the research process. The continual improvement in algorithm design and incorporation of new mathematical models for chemical simulation promises further advances in this field.

3.5.1.4 Summary

Connections have been growing recently between the molecular modeling community and the computational geometry. Many questions in molecular modeling can be understood geometrically in terms of arrangements of spheres in three dimensions. Problems include computing properties of such arrangements such as their volume and topology, testing intersections and collisions between molecules, finding offset surfaces (related to questions of accessibility of molecule subregions to solvents such as water), data structures for computing interatomic forces and performing molecular dynamics simulations, and computer graphics algorithms for rendering molecular models accurately and efficiently (taking advantage of their special geometric structure). Classical molecular modeling has dealt with biological molecules which generally have a tree-like structure, but applications to nanotechnology require dealing with more complicated diamond-like structures; it is unclear to what extent this affects the relevant algorithms.

3.5.1.5 Self Assessment Questions

1. Give the basic concepts of molecular modeling and simulation studies.

3.5.1.6 Reference book:

Bioinformatics Computing – Bryan Bergeron

AUTHOR:

B.M.REDDY M.Tech. (HBTI, Kanpur)

Lecturer, Centre for Biotechnology
Acharya Nagarjuna University.

Lesson 3.5.2

Phylogenetic Analysis

Objective:

- To know about the tree of life, which is graphical representation of the evolutionary relationship (phylogenetic relationships) between all forms of life that we know to exist on earth.
- To know about evolutionary trees, methods and models of building trees.

3.5.2.1 Introduction

The dominant view of the evolution of life is that all existing organisms are derived from some common ancestor and that a new species arises by a splitting of one population into two or more populations that do not crossbreed, rather than by mixing of two populations into one. Therefore, the high level history of life is ideally organized and displayed as a rooted, *directed tree*. Study of the life with an evolutionary perspective is of utmost importance. Phylogenetics is the study of evolutionary relationships. Phylogenetic analysis is the means of inferring or estimating these relationships. The evolutionary history inferred from phylogenetic analysis is usually depicted as branching (treelike) diagrams, which represent a sort of pedigree of the inherited relationships among molecules ('gene trees'), organisms, or both. Phylogenetic is sometimes called cladistics, because the word "clade", a set of descendants from a single ancestor, is derived from the Greek word for branch. Macromolecules, especially sequences have surpassed morphological and other organismal characters as the most popular form of data for phylogenetic analysis.

3.5.2.2 What is the Tree of Life?

The tree of life is a graphical representation of the evolutionary relationship (phylogenetic relationships) between all forms of life that we know to exist on earth. Knowing an organism's phylogeny helps us understand how related one organism is to another. The Tree of Life gives us a visual image of organism relatedness, by depicting a variety of organisms whose position in the tree corresponds to their relatedness to other organisms within the tree. As on a road map, the fewer turns you make, and the less distance you need to travel, the closer you are to your destination. This is the case for organisms on the Tree of Life, as well. For instance, *Giardia* is located near the center of the tree (just below the label "0.1 changes per site"). The archaeon *Methanopyrus* is also located very close to the axis of the tree. By their relative positions, we can deduce that *Methanopyrus* and *Giardia* are closer relatives than the *Flavobacterium* (located near the very end of the Bacterial "road") and *Cryptomonas* (located near the end of the Eucarya "road"). The Tree of Life Web Project is an interesting site that discusses of the nuances of determining phylogeny.

The Tree of Life

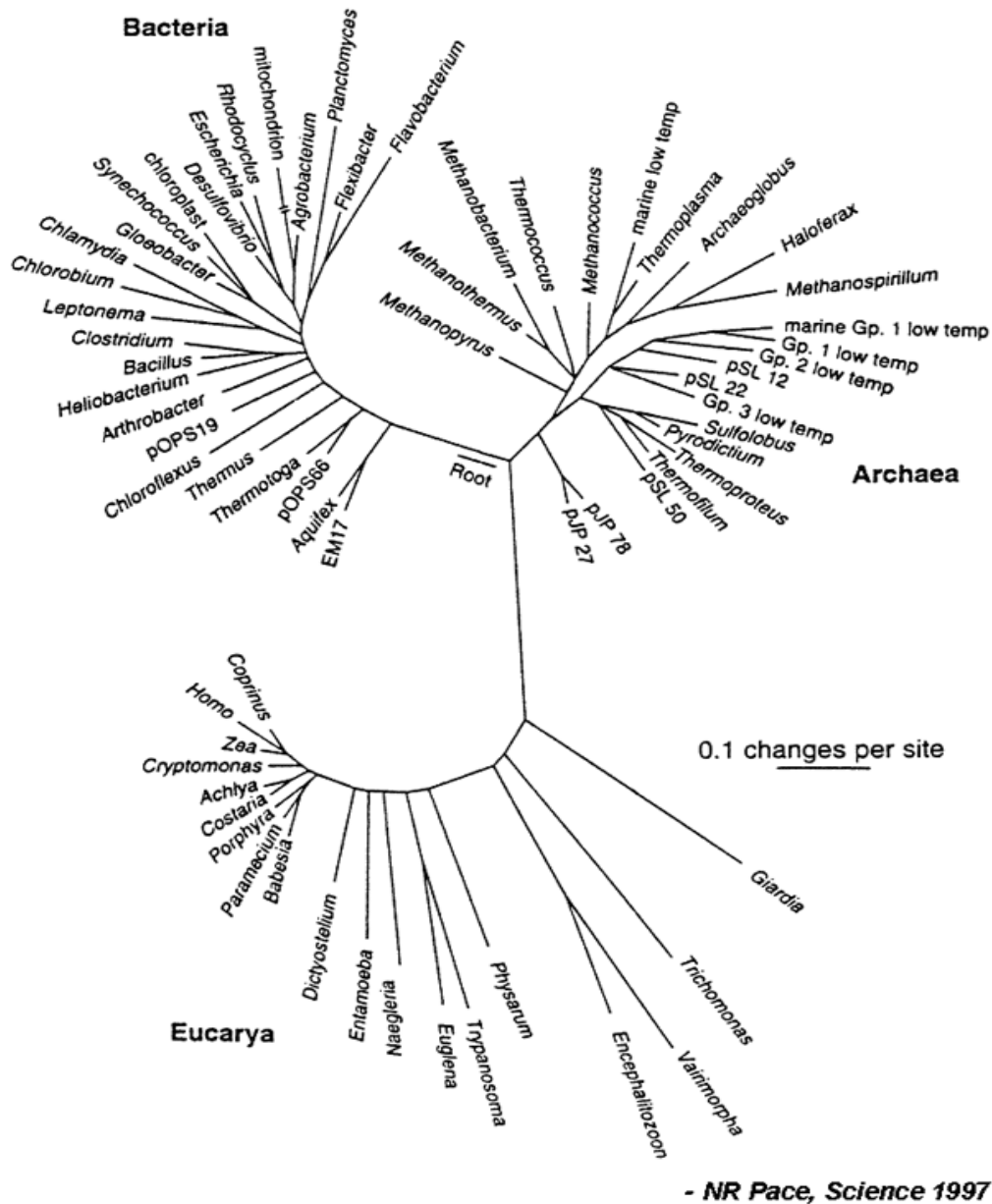


Fig: Rooted tree of life showing principal relationships among prokaryotic domains Bacteria and Archaea.

3.5.2.3 Evolutionary Trees

Phylogenetic analysis gives insight into how a family of related sequences has been derived during evolution. The evolutionary relationships among the sequences are shown as branches of a tree. The length and nesting of these branches reflects the

degree of similarity between any two given sequences. The objective of phylogenetic analysis is to determine the length of the branches and to figure out how the tree should be drawn. Sequences that are the most closely related are drawn as neighboring branches on a tree.

Phylogenetic analysis is dependent upon good multiple sequence alignment programs. Given a multiple sequence alignment, phylogenetic analysis tries to group sequences with similar patterns of substitutions in order to reconstruct a phylogenetic tree. For instance, consider that we have two sequences that are related. Given these two sequences, an ancestral sequence can be (partially) derived. With more similar sequences, more information can be gathered to add to a correct derivation and evolutionary history.

An evolutionary tree is a two dimensional graph showing the evolutionary relationship among a set of items being compared. This set can be organisms, genes, or dna sequences. Consider for the moment that each of the units in the set are referred to as a taxon. Each taxon will be defined by a distinct unit on the tree.

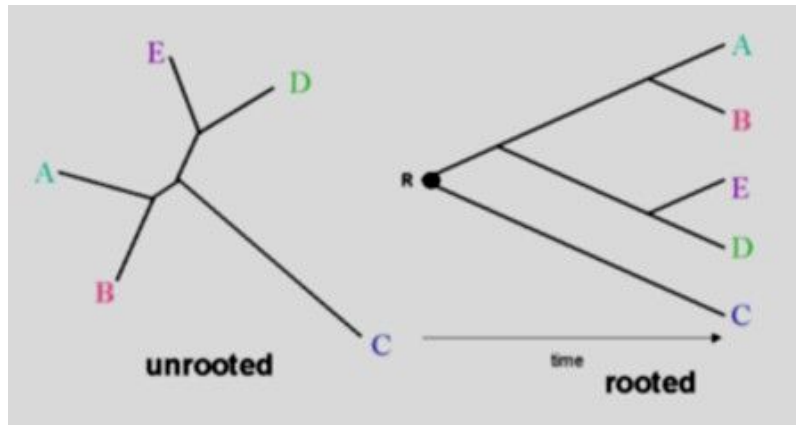
An evolutionary tree is composed of outer branches or leaves that represent the taxa and nodes and branches representing the relationships among the taxa. Two taxa that are derived from the same common ancestor will share a node in the graph. In general, approaches to designing evolutionary trees attempt to define the length of each branch to the next node according to the number of sequence level changes that occurred. One thing to be careful of in phylogenetic analysis is that this distance may not be in direct relation to evolutionary time. Analyses that prescribe to the theory of a uniform rate of mutation are known as the molecular clock hypothesis.

Rooted and Unrooted Trees

In a rooted tree topology, one sequence (the root) is defined to be the common ancestor of all of the other sequences. A unique path leads from the root node to any other node, and the direction of the path indicates evolutionary time. The root is chosen by including a sequence from an organism that is thought to have branched off earlier than the other sequences. If the molecular clock hypothesis holds, it is also possible to predict a root. As the number of sequences increase, the number of possible rooted trees increases very rapidly. In some cases, a bifurcating binary tree is the best model to simulate evolutionary events in which case one species branches off into two separate species.

An unrooted tree (sometimes referred to as a star topology) shows the evolutionary relationship among sequences, without revealing the location of the oldest ancestry. There are fewer choices for an unrooted tree than a rooted tree.

The figure below shows that the two different representations of a phylogenetic tree among five taxa. In a rooted tree, the root represents the common ancestor of the entire taxonomic units. The direction and the length of each branch correspond to evolution time. An unrooted tree is a tree that only specifies relationships among taxonomic unit but not the evolution path. It does not identify common ancestors.



A rooted tree is more commonly constructed to study evolutionary relationships. Rooting a tree can be accomplished by adding an outgroup, or species that is known to be more distantly related to each of the remaining species than they are to each other. The point in the tree where the edge to the outgroup join is therefore the best candidate for the root position. In the example shown below, taxon A is treated as the outgroup. Then the three tree rooting steps are applied. First of all, imagine the tree is made of a string. Then assign an outgroup. In the end, pull the outgroup branch at the root until all the taxa fall opposite of the root to produce a rooted tree. The same tree rooting procedure could be applied to each internal branch. The 4-taxon unrooted tree therefore yields 5 different rooted trees (if all taxa have the same probability to be the outgroup).

3.5.2.4 Models of Phylogenetic Analysis

Phylogenetic analysis is the study of these evolutionary relationships. The view of the history of life as a tree must be frequently modified when considering the evolution. The observed data is molecular sequence data based on which the trees are constructed. The methods to create evolutionary trees are increasingly, encoded into computer programs.

To construct these trees sampling of individuals from a "group" is enough to determine the relationships. Stochastic processes may model evolutionary changes where each "position" evolves independently. In this lecture we discuss the following models for phylogeny.

- Ultrametric Trees
- Additive-distance trees
- Maximum-Parsimony trees

Most tree building algorithms can be classified into two broad categories: distance-based methods and maximum-parsimony methods. In the distance-based methods the input to the problem consists of evolutionary distance (such as edit distances from sequences, strength of antibody cross reactions etc.) and the goal is to reconstruct a weighted tree whose pair wise distances "agree" with a given evolutionary distances. On the other hand the maximum-parsimony methods take a different approach. They do not reduce biological data to distances; instead they are character-based methods that work directly on character data, very often using aligned sequence data. The goal is to build a tree with the input taxa at the leaves and inferred taxa in the internal nodes, that minimizes the cost of mutations implied by the evolutionary history.

Ultrametric Trees

Ultrametric trees can be used in evolutionary reconstruction when the data "perfectly fit" certain strong assumptions such as knowledge about rates of accepted mutations in amino acid sequences of certain proteins. Such a data is obtained from either estimation of mutation distance by lab methods or from PAM distance matrices, for protein sequences. The laboratory methods such as denaturing DNA from two different taxa and observing differences in reannealing temperatures. Ultrametric trees or approximation of them can be used to deduce both the branching patterns of evolutionary history and some measure of the time that has passed along each branch. In these trees the nodes are labeled.

$D[i,j]$ = label of least common ancestor (i,j)

Definition of an Ultrametric tree:

Let D be a symmetric $n \times n$ matrix of real numbers. An Ultrametric tree for D is a rooted tree T with the following properties:

1. T contains n leaves, each labeled by a unique row of D .
2. Each internal node of T is labeled by one entry from D and has at least two children.
3. Along any path from the root to a leaf, the numbers labeling internal nodes strictly decrease.
4. For any two leaves i, j of T , $D(i, j)$ is the label of the least common ancestor of i and j in T .

Thus an Ultrametric tree is a compact representation of the matrix D .

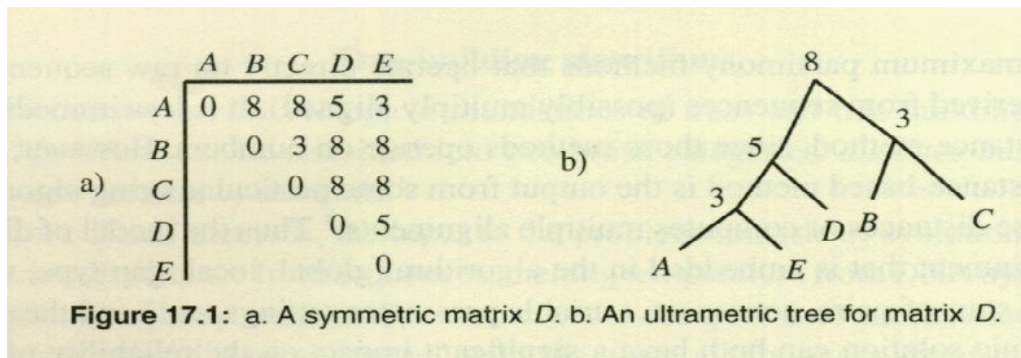


Figure 17.1: a. A symmetric matrix D . b. An ultrametric tree for matrix D .

The main property for the Ultrametric trees is as follows:

A matrix D is an Ultrametric distance matrix and can be represented by a Ultrametric tree T if and only if every three indices i, j, k there is a tie for the maximum of $D[i, j]$, $D[i, k]$, $D[j, k]$

In the above example if we consider row 'A' to be 'i'; 'C' to be 'j' and 'D' to be 'k'

Then:

$$D[i, j] = 8 ; D[j, k] = 5; D[i, k] = 8$$

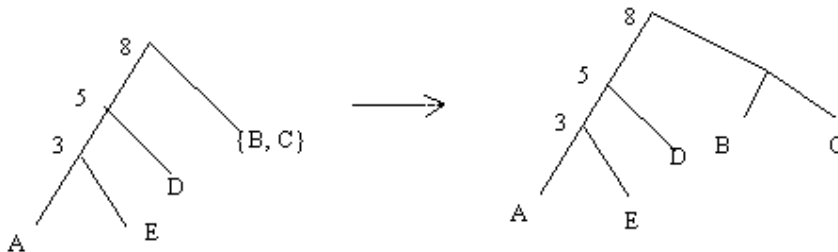
Thus there is a tie for the maximum satisfying the Ultrametric property.

Note: This tree is not necessarily a binary tree. It could be of any of the following form.



The tree is formed from the matrix in the following manner:

In the example matrix above, consider the row 'A'. It has distances 0 8 8 5 3 to the nodes A,B,C,D,E. Clearly 8 is the largest and least common ancestor of 'A' and there are nodes 5 and 3 on the same path as follows



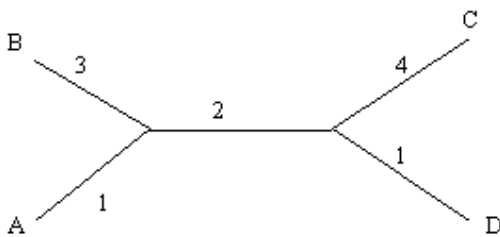
Running time: If D is an Ultrametric matrix, then an Ultrametric tree for D can be constructed in $O(n^2)$ time.

Additive-distance Trees

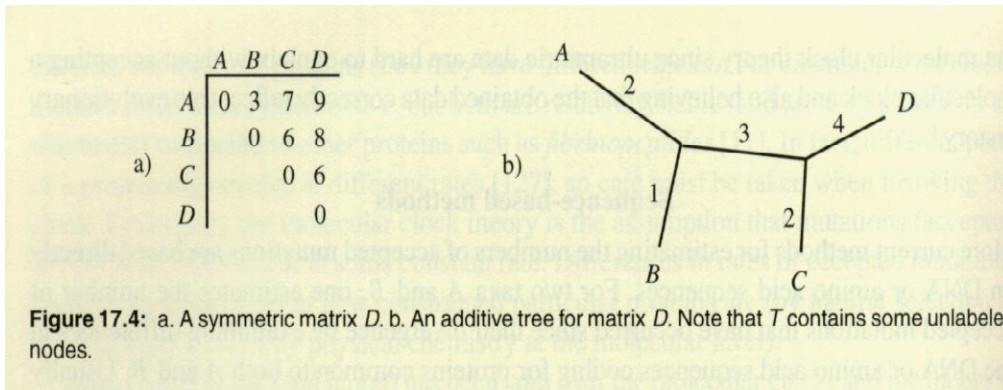
Ultrametric data is very useful for phylogenetic reconstruction. The real data are rarely Ultrametric. A weaker requirement on the evolutionary-distance data is that they be additive.

Definition of an Additive-distance tree:

Let D be a symmetric $n \times n$ matrix where the numbers on the diagonal are all zero and the off-diagonal numbers are all strictly positive. Let T be an edge weighted tree with at least n nodes, where n distinct nodes of T are labeled with the rows of D. Tree T is called an additive tree for matrix D if, for every pair of labeled nodes (i, j) the path from node i to node j has total weight (or distance) exactly D(i, j)



Example distances: $D[A, C] = 7$; $D[A, B] = 4$

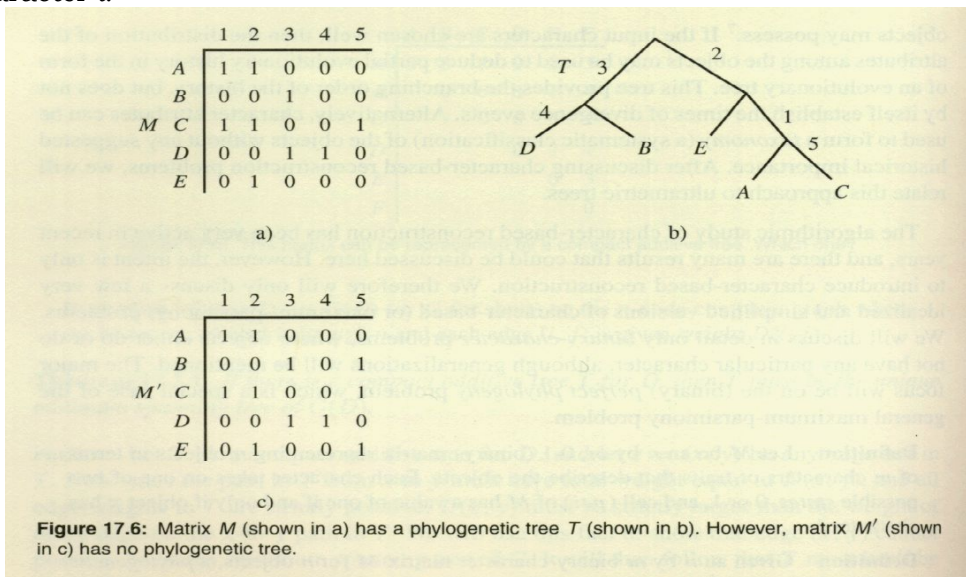


Parsimony Tree

The other major approach for constructing evolutionary tree is the character-based approach. In this approach, the input is a set of attributes called characters that the objects may possess. The characters referred to here are not a member of an alphabet but to an attribute or trait of an object. This tree provides the branching order of the history, but does not by itself establish the times of divergence events. Alternatively, character attributes can be used to form a taxonomy of objects without any historical importance.

Definition of Parsimony tree:

Let M be an $n \times m$, 0-1 (binary) matrix representing n objects in terms of m characters or traits that describe the objects. Each character takes on one of two possible states, 0 or 1, and cell (p, i) of M has a value of one if and only if object p has character i .



In the figure above M' does not have perfect phylogeny because it violates the following property of parsimony tree:

M is said to have perfect phylogeny if and only if O_i and O_j are either disjoint or one is contained in the other

O_i = Set of "taxa" with character i

O_j = Set of "taxa" with character j

The origin of characters is traditionally morphological features or traits of the objects. These characters may be gross features such as "possessing a backbone" or may be very fine features only understood by specialists studying those organisms.

Running time: The algorithm could take $O(nm^2)$ time but it has been shown this could be improved to $O(nm)$ time.

3.5.2.5 Relationship Of Phylogenetic Analysis To Sequence Alignment

When the sequences of two nucleic acid or protein molecules found in two different organisms are similar, they are likely to have been derived from a common ancestor sequence. A sequence alignment reveals which positions in the sequences were conserved and which diverged from a common ancestor sequence, as illustrated in Figure 1. When one is quite certain that two sequences share an evolutionary relationship, the sequences are referred to as being homologous. The commonest method of multiple sequence alignment first aligns the most closely related pair of sequences and then sequentially adds more distantly related sequences or sets of sequences to this initial alignment. The alignment so obtained is influenced by the most alike sequences in the group and thus may not represent a reliable history of the evolutionary changes that have occurred. Other methods of multiple sequence alignment attempt to circumvent the influence of alike sequences. Once a multiple sequence alignment has been obtained, each column is assumed to correspond to an individual site that has been evolving according to the observed sequence variation in the column. Most methods of phylogenetic analysis assume that each position in the protein or nucleic acid sequence changes independently of the others. As indicated above, the analysis of sequences that are strongly similar along their entire lengths is quite straightforward. However, to align most sequences requires the positioning of gaps in the alignment. Gaps represent an insertion or deletion of one or more sequence characters during evolution. Proteins that align well are likely to have the same three-dimensional structure. In general, sequences that lie in the core structure of such proteins are not subject to insertions or deletions because any amino acid substitutions must fit into the packed hydrophobic environment of the core. Gaps should therefore be rare in regions of multiple sequence alignments that represent these core sequences. In contrast, more variation, including insertions and deletions, may be found in the loop regions on the outside of the three-dimensional structure because these regions do not influence the core structure as much. Loop regions interact with the environment of small molecules, membranes, and other proteins.

Gaps in alignments can be thought of as representing mutational changes in sequences, including insertions, deletions, or rearrangements of genetic material. The expectation that a gap of virtually any length can occur as a single event introduces the problem of judging how many individual changes have occurred and in what order. Gaps are treated in various ways by phylogenetic programs, but no clear-cut model as to how they should be treated has been devised. Many methods ignore gaps or focus on regions in an alignment that do not have any gaps. Nevertheless, gaps can be useful as phylogenetic markers in some situations.

Another approach for handling gaps is to avoid analysis of individual sites in the sequence alignment and instead to use sequence similarity scores as a basis for phylogenetic analysis. Rather than trying to decide what has happened at each sequence position in an alignment, a similarity score based on a scoring matrix with penalties for gaps is often used. As discussed below, these scores may be converted to distance scores that are suitable for phylogenetic analysis by distance methods.

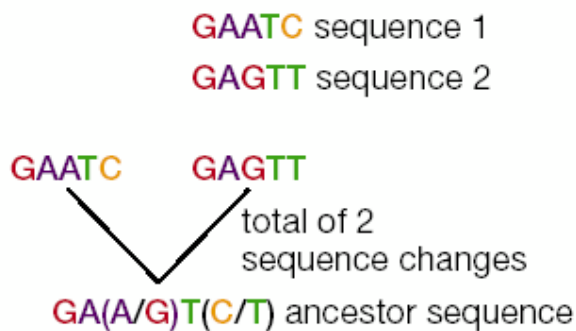


Figure 1. Origin of similar sequences. Sequences 1 and 2 are each assumed to be derived from a common ancestor sequence. Some of the ancestor sequence can be inferred from conserved positions in the two sequences. For positions that vary, there are two possible choices at these sites in the ancestor.

3.5.2.6 Methods for Determining Evolutionary Trees

Distance-Based Methods:

The most popular distance-based methods are the unweighted pair group method with arithmetic mean (UPGMA), neighbor joining (NJ) and those that optimize the additivity of a distance tree (FM and ME).

UPGMA Method

This method follows a clustering procedure:

- (1) Assume that initially each species is a cluster on its own.
- (2) Join closest 2 clusters and recalculate distance of the joint pair by taking the average.
- (3) Repeat this process until all species are connected in a single cluster.

Strictly speaking, this algorithm is phenetic, which does not aim to reflect evolutionary descent. It assigns equal weight on the distance and assumes a randomized molecular clock.

WPGMA is a similar algorithm but assigns different weight on the distances. UPGMS method is simple, fast and has been extensively used in literature. However, it behaves poorly at most cases where the above presumptions are not met.

Neighbor Joining Method (NJ)

This algorithm does not make the assumption of molecular clock and adjust for the rate variation among branches. It begins with an unresolved star-like tree (fig 4(a)). Each pair is evaluated for being joined and the sum of all branches length is calculated of the resultant tree. The pair that yields the smallest sum is considered the closest neighbors and is thus joined. A new branch is inserted between them and the rest of the tree and the branch length is recalculated. This process is repeated until only one

terminal is present. NJ method is comparatively rapid and generally gives better results than UPGMA method. But it produces only one tree and neglects other possible trees, which might be as good as NJ trees, if not significantly better. Moreover, since errors in distance estimates are exponentially larger for longer distances, under some condition, this method will yield a biased tree.

Weighted Neighbor-Joining (Weighbor)

This is a new method proposed recently

. The Weighbor criterion consists of two terms; an additivity term (of external branches) and a positivity term (of internal branches), that quantifies the implications of joining the pair. Weighbor gives less weight to the longer distances in the distance matrix and the resulting trees are less sensitive to specific biases than NJ and relatively immune to the "long branches attraction/distracton" drawbacks observed with other methods.

Fitch-Margoliash (FM) and Minimum Evolution (ME) Methods

Fitch and Margoliash proposed in 1967 a criteria (FM Method) for fitting trees to distance matrices. This method seeks the least squared fit of all observed pair-wise distances to the expected distance of a tree. The ME method also seeks the tree with the minimum sum of branch lengths. But instead of using all the pair-wise distances as FM, it fixed the internal nodes by using the distance to external nodes and then optimizes the internal branch lengths. FM and ME methods perform best in the group of distance-based methods, but they work much more slowly than NJ, which generally yield a very close tree to these methods.

Character-Based Methods

Distance-based methods are more rapid and less computationally intensive than characterbased methods, but the actual characters are discarded once the distance matrix is derived. On the other hand, character-based methods make use of all known evolutionary information, i.e. the individual substitutions among the sequences, to determine the most likely ancestral relationships.

Maximum parsimony (MP)

The criterion of MP method is that the simplest explanation of the data is preferred, because it requires the fewest conjectures. By this criterion, the MP tree is the one with fewest substitutions/evolutionary changes for all sequences to derive from a common ancestor. For each site in the alignment, all possible trees are evaluated and are given a score based on the number of evolutionary changes needed to produce the observed sequence changes. The best tree is thus the one that minimized the overall number of mutation at *all* site.

MP works faster than ML and the weighted parsimony schemes can deal with most of the different models used by ML. However, this method yields little information about the branch lengths and suffers badly from long-branch attraction, that is the long branches have become artificially connected because of accumulation of inhomogous similarities, even if they are not at all phylogenetically related. MP yields more than one tree with the same score.

Maximum Likelihood (ML)

Like MP methods, ML method also uses each position in an alignment and evaluates all possible trees. It calculates the likelihood for each tree and seeks the one with the maximum likelihood. For a given tree, at each site, the likelihood is determined by evaluating the probability that a certain evolutionary model has generated the observed data. The likelihood's for each site are then multiplied to provide likelihood for each tree.

ML method is the slowest and most computationally intensive method, though it seems to

give the best result and the most informative tree.

3.5.2.7 Difficulties with phylogenetic analysis

Phylogenetic analysis would be easier if evolution occurred in a vertical fashion. However, horizontal or lateral transfer of genetic material (for instance through viruses) occurs, which makes it difficult to determine the phylogenetic origin of some evolutionary events.

If a gene is under selective pressure in different organisms, it can be rapidly evolving. Such an evolution can mask earlier changes that had occurred phylogenetically. In addition, different regions of a genome are under different pressures, and therefore different sites within two comparative sequences may be evolving at different rates.

Rearrangements of genetic material can also lead to false conclusions with phylogenetic analysis, especially if two sequences of different evolutionary origins are placed next to each other.

Gene duplication events also cause problems with phylogenetic analysis, since the duplicated genes can evolve along separate pathways, leading to different functions.

Summary

Phylogenetics is the study of evolutionary relationships. Phylogenetic analysis is the means of inferring or estimating these relationships. The tree of life is a graphical representation of the evolutionary relationship (phylogenetic relationships) between all forms of life that we know to exist on earth. The evolutionary relationships among the sequences are shown as branches of a tree. The length and nesting of these branches reflects the degree of similarity between any two given sequences. Phylogenetic analysis would be easier if evolution occurred in a vertical fashion.

Model Questions:

1. What is phylogenetic analysis and evolutionary trees? Discuss different models of phylogenetic analysis?
2. Write a short note on the difference between phylogenetic analysis and sequence alignment? Discuss various methods of tree building?

References:

1. Bioinformatics - Sequence and Genome Analysis- David W. Mount.
2. Bioinformatics – A practical guide to the Analysis of Genes and Proteins – Andreas D. Baxevanis and B F Francis Quelette.
3. Bioinformatics, Concepts, Skills, and Applications by S.C.Rastogi & NAmita MEndiratta.

B.M.REDDY

Lesson 3.5.3**Structural Prediction of Biopolymers****Structure****3.5.3.1 Introduction****3.5.3.2 What can structure prediction do for us?****3.5.3.3 Protein Secondary Structure Prediction****3.5.3.4 RNA Secondary Structure Prediction****3.5.3.5 Methods for protein structure prediction****3.5.3.6 Tertiary structure prediction****3.5.3.7 Methods for predicting the tertiary structure****Summary****Model Questions****References****Objective:**

The objective of this lesson is to know the importance of structure prediction and methods of prediction.

3.5.3.1 Introduction

A **biopolymer** is a polymer found in nature. Starch, proteins and peptides, DNA, and RNA are all examples of biopolymers, in which the monomer units, respectively, are sugars, amino acids, and nucleic acids. The exact chemical composition and the sequence in which these units are arranged is called the polymer's primary structure. Many biopolymers spontaneously "fold" into characteristic shapes, which determine their biological functions and depend in a complicated way on their primary structures. Structural biology is the study of the shapes of biopolymers.

Genome sequencing projects are producing linear amino acid sequences, but full understanding of the biological role of these proteins will require knowledge of their structure and function. Although experimental structure determination methods are providing high-resolution structure information about a subset of the proteins, computational structure prediction methods will provide valuable information for the large fraction of sequences whose structures will not be determined experimentally. The first class of protein structure prediction methods, including threading and comparative modeling, rely on detectable similarity spanning most of the modeled sequence and at least one known structure. The second class of methods, de novo or ab initio methods, predict the structure from sequence alone, without relying on similarity at the fold level between the modeled sequence and any of the known structures.

Primary Structure:

The convention for a protein is to list its constituent amino acid residues as they occur from the amino terminus to the carboxylic acid terminus. The convention for a nucleic acid sequence is to list the nucleotides as they occur from the 5' end to the 3' end of the polymer chain, where 5' and 3' refer to the numbering of carbons around the

ribose ring which participate in forming the phosphate diester linkages of the chain. Such a sequence is called the primary structure of the biopolymer.

There are a number of biophysical techniques for determining sequence information. Protein sequence can be determined by Edman degradation, in which the N-terminal residues are hydrolyzed from the chain one at a time, derivatized, and then identified. Mass spectrometer techniques can also be used. Nucleic acid sequence can be determined using gel electrophoresis and capillary electrophoresis. Lastly, mechanical properties of these biopolymers can often be measured using optical tweezers or atomic force microscopy.

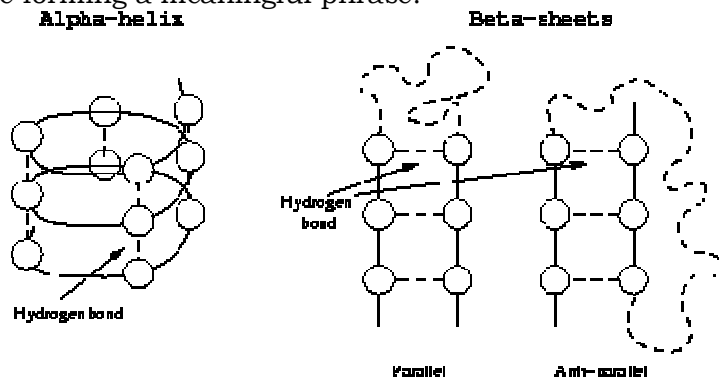
3.5.3.2 What can structure prediction do for us?

Given the large volume of genes being sequenced, the rate of new protein sequences is growing exponentially relative to the rate of protein structures being solved by experimental methods. In many situations, even a crude or approximate model can help an experimentalist significantly in guiding his/her experiments. Thus even though the current methods are still in their infancy, prediction of structures for all protein sequences of complete genomes in conjunction with experimental work is a realistic goal. Structural analyses on demand of proteins for further mutagenesis, substrate and inhibitor design, and enhanced function and stability is also possible, as is analysis of basic functional behaviour on demand using time-tested methods such as molecular dynamics simulations. These methods method can use structural data and methods for structure prediction to probe protein and organismal function and evolution.

SECONDARY STRUCTURE PREDICTION

3.5.3.3 Protein Secondary Structure Prediction

Proteins present local regularities in their 3D structure, formed and maintained by hydrogen bonds between atoms. These regular structures are referred to as the protein's *secondary structure*. The most common configurations observed in proteins are called *alpha helices* and *beta strands*, while all the other conformations are referred to as *coils*. A group of adjacent amino acids sharing the same conformation are members of a *segment* of secondary structure. Segments of secondary structure are well defined and stable aggregations of amino acids which strongly influence the chain's folding and which usually carry out specific functions inside the protein, like a list of words in a particular language forming a meaningful phrase.



Schematic representations of secondary structures segments.

Reliable predictors of the secondary structure of a protein are fundamental to study its folding and functions. Threading algorithms which attempts to study the fold of unknown proteins, use predicted secondary structure sequences to search in databases of known folds. Moreover, the predicted secondary structure content of a protein can be used to identify its folding family (CATH, SCOP) and thus estimate its functions.

Proteins are the biological molecules that are the building blocks of cells and organs, and the biochemical processes required to keep living organisms alive are catalyzed and regulated by a particular category of proteins called enzymes. Proteins are linear polymers of amino acids that fold into complex conformations dictated by the physical and chemical properties of the amino acid chain. The biological function of a protein is dependent on the protein folding into the correct, or "native", state. Protein structure is described by biologists in terms of primary structure, which is the amino acid sequence, secondary structure, wherein the polypeptide backbone assembles into local regions of alpha-helices, beta-sheets, coils and turns, tertiary structure, which refers to the entire 3-dimensional structure of the protein, and quaternary structure, which describes interactions between separate polypeptide chains, called subunits, that exist in some large protein complexes. Computational methods have been developed that can predict protein secondary structure with a reasonable degree of accuracy. Prediction methods exist for predicting tertiary structure, but the accuracy of such methods is highly dependent on whether or not the protein in question is related in sequence to any members of the existing library of known protein structures. The development of *ab initio* tools to predict the complete structural fold of a protein from its amino acid sequence is a burgeoning field in computational biology, but true attainment of this goal is still pretty distant.

Determining the process by which proteins fold into particular shapes, characteristic of their amino acid sequence, is commonly called "the protein folding problem". One approach to studying the protein folding process is the application of statistical mechanics techniques and simulations to the *study of protein folding*. These methods allow the investigation of larger systems than methods that try to represent atomic detail in their simulations of biological molecules, and have had success correlating the computational folding model with folding intermediates and transition states that have been experimentally measured for a limited test set of relatively large proteins.

Although attempts at predicting tertiary and quaternary structure from the amino acid sequence of proteins are relatively new, methods for predicting protein secondary structure have been in existence for some time. Depending on the method, secondary structure predictions can be performed with approximately 60 - 70% accuracy. Originally, empirical prediction methods were based on tables which listed each amino acid and the frequency with which that amino acid was found in alpha-helices, beta-sheets, turns and random coil. Currently, prediction methods usually employ machine learning in the form of neural networks that are trained with test sets consisting of sequences with known structure. In these cases, the selection of the test set is critically related to the accuracy of the method. However, given the ever increasing number of known structural folds, selecting a representative test set that includes many proteins of diverse structure has become easier.

3.5.3.4 RNA Secondary Structure Prediction:

The study of RNA structure calls for a distinct set of computational tools designed expressly for RNA applications. Recall there are three major categories of RNA, messenger RNA (mRNA), transfer RNA (tRNA) and ribosomal RNA (rRNA). Ribosomal RNA and ribozymes have catalytic functions, like proteins, while messenger RNA has an information storage function, like DNA

RNA is usually thought of as a single stranded linear molecule, however, in a biological system this is not the case. Frequently, different regions of the same RNA strand will fold together via base pair interactions to make intricate secondary and tertiary structures that are essential for correct biological function. Common secondary structure motifs include hairpin loops, stems, and bulges.

Though RNA is usually single stranded, in some RNA virus genomes it will form a double stranded helix. However, unlike DNA, RNA forms an A-form double helix. The RNA double helix differs from that of the DNA double helix because of the presence of ribose, rather than deoxyribose, in the sugar phosphate backbone of the molecule. The addition of a hydroxyl group at the C2 position in the ribose sugar is responsible for the A-form geometry in double stranded RNA. The A-form makes a right-handed helix, like the B-form double helix, but is a shorter, wider helix than the B-form, and the major groove is deep, but narrow, making it virtually inaccessible to proteins. It is in the major groove that the chemical groups are sequence-specific and dependent on base identity, and therefore, this is where proteins tend to bind a DNA double helix. Because the RNA A-form double helix contains a major groove that is too narrow and deep for proteins to access, the minor groove becomes more important for protein interactions with RNA helices. Also, proteins that interact with specific RNA sequences commonly bind single-stranded RNA segments. For an example of an RNA/protein complex, view the *NDB entry* for the Protein/Hepatitis Delta Virus Ribozyme Complex. When a strand of DNA forms a double helix with a strand of RNA, this will also result in an A-form helix.

3.5.3.5 Methods for protein structure prediction

One of the most important open problems in molecular biology is the prediction of the spatial conformation of a protein from its primary structure, ie from its sequence of amino acids. The classical methods for structure analysis of proteins are X-ray crystallography and nuclear magnetic resonance (NMR). Unfortunately, these techniques are expensive and can take a long time (sometimes more than a year). On the other hand, the sequencing of proteins is relatively fast, simple, and inexpensive. As a result, there is a large gap between the number of known protein sequences and the number of known three-dimensional protein structures. This gap has grown over the past decade (and is expected to keep growing) as a result of the various genome projects worldwide. Thus, computational methods which may give some indication of structure and/or function of proteins are becoming increasingly important. Unfortunately, since it was discovered that proteins are capable of folding into their unique native state without any additional genetic mechanisms, over 25 years of effort has been expended on the determination of the three-dimensional structure from the sequence alone, without further experimental data. Despite the amount of effort, the protein folding problem remains largely unsolved and is therefore one of the most fundamental unsolved problems in computational molecular biology today.

How can the native state of a protein be predicted (either the exact or the approximate overall fold)? There are three major approaches to this problem: 'comparative modelling', 'threading', and 'ab initio prediction'. Comparative modelling exploits the fact that evolutionarily related proteins with similar sequences, as measured by the percentage of identical residues at each position based on an optimal structural superposition, often have similar structures. For example, two sequences that have just 25% sequence identity usually have the same overall fold. Threading methods compare a target sequence against a library of structural templates, producing a list of scores. The scores are then ranked and the fold with the best score is assumed to be the one adopted by the sequence. Finally, the ab initio prediction methods consist in modelling all the energetics involved in the process of folding, and then in finding the structure with lowest free energy. This approach is based on the 'thermodynamic hypothesis', which states that the native structure of a protein is the one for which the free energy achieves the global minimum. While ab initio prediction is clearly the most difficult, it is arguably the most useful approach.

There are three major theoretical methods for predicting the structure of proteins: comparative (Homology) modelling, fold recognition, and *ab initio* prediction.

Comparative modelling

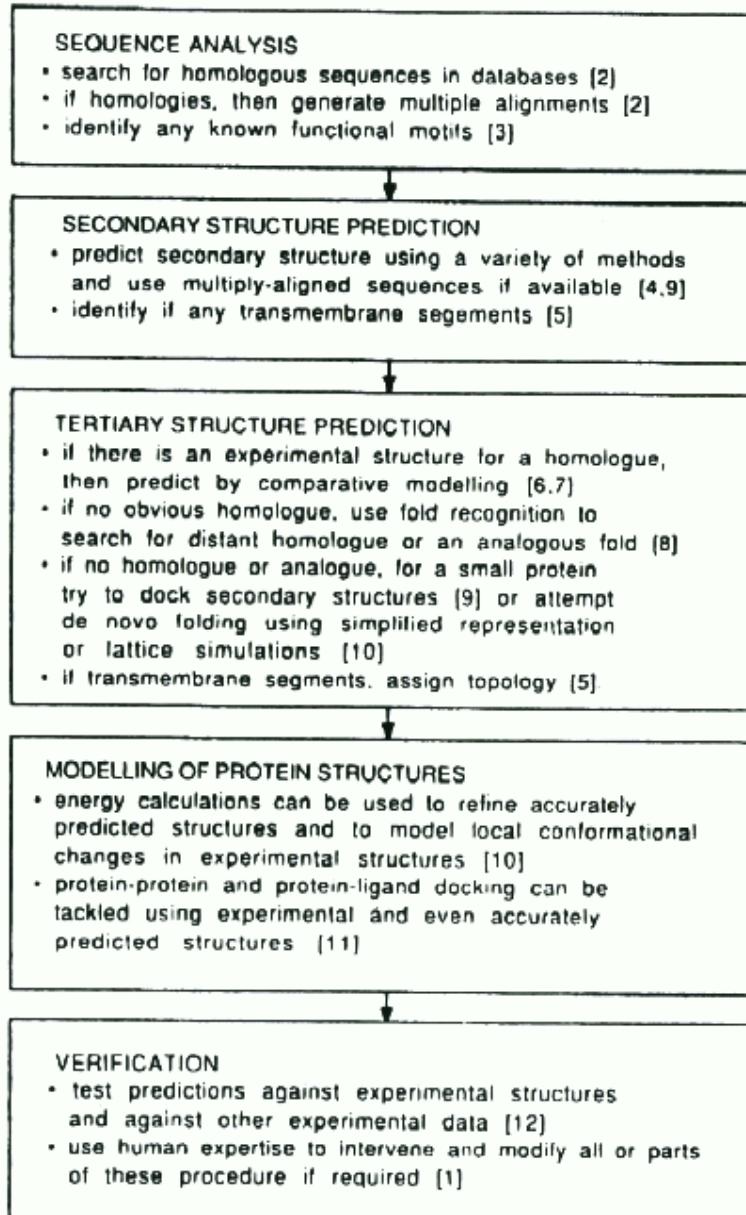
Comparative modelling exploits the fact that evolutionarily related proteins with similar sequences, as measured by the percentage of identical residues at each position based on an optimal structural superposition, have similar structures. The similarity of structures is very high in the so-called "core regions", which typically are comprised of a framework of secondary structure elements such as alpha-helices and beta-sheets. Loop regions connect these secondary structures and generally vary even in pairs of homologous structures with a high degree of sequence similarity.

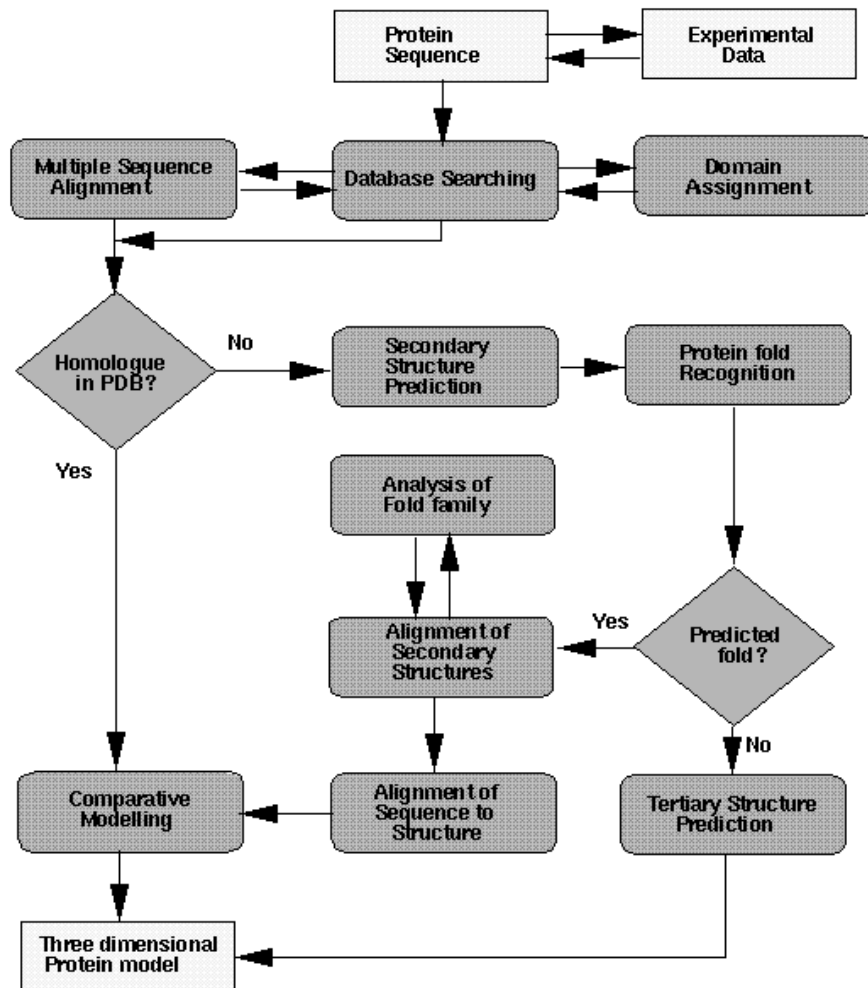
The process of building a comparative model is conceptually straightforward. First, an alignment is performed between the sequence for which the structure has been determined by experimental methods (the parent) with the sequence to be modelled (the target). This sequence alignment is used to construct an initial model (sometimes referred to as a framework or template) by copying over some main chain and side chain coordinates from the parent structure based on the equivalent residue in the sequence alignment. Side chains must be built for residues in the target that does not correspond to an identity in the alignment, and for residues where the side chain conformation is thought to vary in the target relative to the parent structure. Main chains must be built in the case of insertions, regions surrounding a deletion, and in other regions of suspected main chain variation.

Homology utilizes structure and sequence similarities for predicting unknown protein structures. This is useful since protein and nucleic acid sequences are continually published, and structure determination from X-ray diffraction and NMR spectroscopy remains a time-consuming process. As a result, many proteins exist for which sequences are known, but molecular structures are unknown. Such proteins are often members of families containing related proteins with known structures. The Homology module in Insight II builds a three-dimensional model of a protein using both its amino acid sequence and the structures of known, related proteins.

Overview

The Homology program provides simultaneous optimization of both structure and sequence homologies for multiple proteins in a three-dimensional graphics environment, based on a method developed by Greer1.





The protein models built with Homology can be used in drug design or as starting points for X-ray or NMR structure refinement. Typical examples include modeling rennin or HIV protease to known structures of aspartyl proteases³, modeling TPA to known structures of serine proteases, modeling new or modified immunoglobulins to known structures within the family, generating starting models for NMR structure determination⁴, and generating starting models for molecular replacement and model fitting in X-ray crystallography of homologous or mutant proteins.

Integrated Protein Modeling Environment

Homology is fully integrated into Insight II®, Accelrys' 3D modeling environment. What is created within Homology can be used later by other Accelrys programs for structure refinement and analysis of large and small molecules. Such programs include Biopolymer for constructing models of peptides, proteins, carbohydrates, and nucleic acids, CHARMM® and Discover® for molecular mechanics and dynamics calculation, DeCIPHER for analysis of molecular dynamics calculations, DelPhi for calculating electrostatic potentials, MODELER for automatic homology model

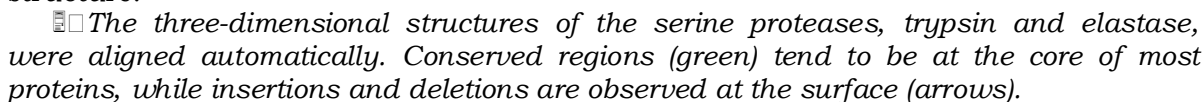
building, SeqFold for protein fold prediction, NMRchitect for analyzing NMR data, Affinity for protein-ligand docking, and Ludi for *de novo* ligand design.

Homology combines fully automatic model building procedures and user-controlled operations, allowing scientific judgement at any stage of the process:

- Search protein structure databases for proteins similar to model protein being built
- Display and align amino acid sequences, up to 10 sequences simultaneously
- Find structurally conserved regions (SCRs)
- Propose structures for loops using structural database searching and *de novo* methods
- Copy coordinates from reference proteins to model
- Refine new structure using molecular mechanics and dynamics

Structure Validation

You can use the ProStat tool to compare either existing X-ray and NMR structures or newly created model structures to validate structural parameters. The interactive display enables you to quickly identify potential problem areas in your protein structure.

 *The three-dimensional structures of the serine proteases, trypsin and elastase, were aligned automatically. Conserved regions (green) tend to be at the core of most proteins, while insertions and deletions are observed at the surface (arrows).*

Advantages of Homology

- Structural similarities can be determined easily, either manually or automatically
- Interactive system that permits the inclusion of scientific judgement and previous experience

- Simultaneous optimization of structures and sequences
- All input monitored for self-consistency

Structure Building Tools

- Allows automatic loop building either by database searching or through random conformational searching

- Enables secondary structure prediction
- Provides hydropathy plotting and graphing
- Generates consensus SCRs from alignments of multiple proteins
- Coordinates transfer from any reference protein to the model in each SCR
- Automatically replaces side chains, preserving the conformations of the reference protein

- Completes C α traces
- Manual or automatic optimization of side chain conformations

using a rotamer library

- Automatically assigns coordinates at the N- and C-termini
- Identifies proteins of similar secondary structural motifs by database searching

Fold recognition or "threading"

Threading uses a database of known three-dimensional structures to match sequences without known structure with protein folds. This is accomplished by the aid of a scoring function that assesses the fit of a sequence to a given fold. These functions are usually derived from a database of known structures and generally include a pairwise atom contact and solvation terms. Threading methods compare a target sequence against a library of structural templates, producing a list of scores. The scores

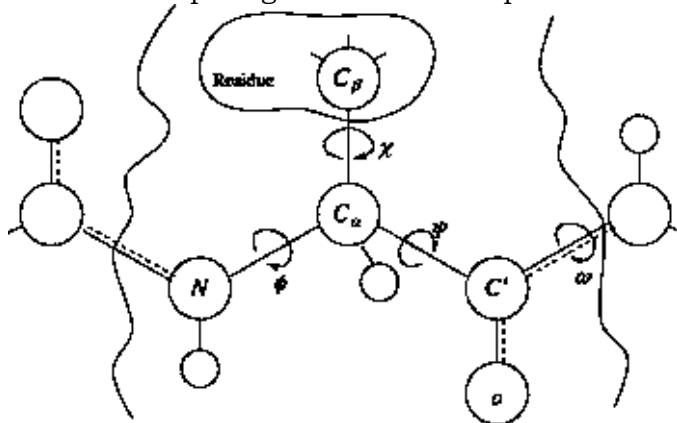
are then ranked and the fold with the best score is assumed to be the one adopted by the sequence. The methods to fit a sequence against a library of folds can be extremely elaborate computationally, such as those involving double dynamic programming, dynamic programming with frozen approximation, Gibbs Sampling using a database of "threading" cores, and branch and bound heuristics, or as "simple" as using sophisticated sequence alignment methods such as Hidden Markov Models.

Ab initio prediction

The *ab initio* approach is a mixture of science and engineering. The science is in understanding how the three-dimensional structure of proteins is attained. The engineering portion is in deducing the three-dimensional structure given the sequence. The biggest challenge with regards to the folding problem is with regards to *ab initio* prediction, which can be broken down into two components: devising a scoring function that can distinguish between correct (native or native-like) structures from incorrect (non-native) ones, and a search method to explore the conformational space. In many *ab initio* methods, the two components are coupled together such that a search function drives, and is driven by, the scoring function to find native-like structures.

Currently there does not exist a reliable and general scoring function that can always drive a search to a native fold, and there is no reliable and general search method that can sample the conformational space adequately to guarantee a significant fraction of near-natives (< 3.0 angstroms RMSD from the experimental structure).

Some methods for *ab initio* prediction include Molecular Dynamics (MD) simulations of proteins and protein-substrate complexes provide a detailed and dynamic picture of the nature of inter-atomic interactions with regards to protein structure and function; Monte Carlo (MC) simulations that do not use forces but rather compare energies, via the use of Boltzmann probabilities; Genetic Algorithms which tries to improve on the sampling and the convergence of MC approaches, and exhaustive and semi-exhaustive lattice-based studies which are based on using a crude/approximate fold representation (such as two residues per lattice point) and then exploring all or large amounts of conformational space given the crude representation.



Backbone torsion angles of a protein

Unfortunately, this direct approach is not really useful in practice, both due to the difficulty of formulating an adequate scoring function and to the formidable computational effort required to solve it. To see why this is so, note that any fully-

descriptive energy function must consider interactions between all pairs of atoms in the polypeptide chain, and the number of such pairs grows exponentially with the number of amino acids in the protein. To make matters worse, a full model would also have to contend with vitally important interactions between the protein's atoms and the environment, the so-called 'hydrophobic effect'. Thus, in order to make the computation practical, simplifying assumptions must necessarily be made.

Different computational approaches to the problem differ as to which assumptions are made. A possible approach, based on the discretization of the conformational space, is that of deriving a protein-centric lattice, by allowing the backbone torsion angles, phi, psi, and omega, to take only a discrete set of values for each different residue type. Under biological conditions, the bond lengths and bond angles are fairly rigid. Therefore, the internal torsion angles along the protein backbone determine the main features of the final geometric shape of the folded protein. Furthermore, one can assume that each of the torsion angles is restricted to a small, finite set of values for each different residue type. As a matter of fact, not all torsion angles are created equally. While they may feasibly take any value from -180 to 180 degrees, in nature all these values do not occur with uniform probability. This is due to the geometric constraints from neighboring atoms, which dramatically restrict the commonly occurring legal values for the torsion angles. In particular, the peptide bond is rigid (and shorter than expected) because of the partial double-bond character of the CO-NH bond. Hence the torsion angle omega around this bond generally occurs in only two conformations: 'cis' (omega about 0 degrees) and 'trans' (omega about 180 degrees), with the trans conformation being by far the more common. Moreover the other two torsion angles, phi and psi, are highly constrained, as noted by G. N. Ramachandran (1968).

3.5.3.6 Tertiary structure prediction

To predict the three-dimensional structure of one, few or even a whole family of proteins using different bioinformatics approaches. In general, its not handle with the linear sequence of amino acids, α -helices and β -strands of the proteins (primary and 2nd structure levels). Instead, it focuses on knowing the local or overall folding conformations of the proteins of interest (personal understanding).

As the number of completely sequenced genomes rapidly increases, including now the complete Human Genome sequence, the post-genomic problems of genome-scale protein structure determination and the issue of gene function identification become ever more pressing. In other words, genome sequencing already provided mass of information about the linear amino acid sequence, but full understanding of the biological role of these proteins is far beyond the capability of the conventional biology experimental methods, such as NMR and X-ray crystallography. It has been proven that knowing the 3-D structure of a protein is helpful to predict its function.

3.5.3.7 Methods for predicting the tertiary structure

The main stream approaches:

•Threading and comparative modeling — 1st class method

Relying on detectable similarity spanning most of the modeled sequence and at least one known structure.

The threading method is also called fold assignment method, which assigns a fold to the target sequence by aligning it with the most compatible

known protein structure from a set of alternatives. As such, the fold assignment methods are best seen as the first, and in many cases the most important, step in comparative protein structure modeling.

•**De novo / ab initio modeling — 2nd class method**

Predicting the structure from sequence alone, without relying on similarity at the fold level between the modeled sequence and any of the known structures.

There are also other approaches, which employ a combination of these two methods or with other biological methods.

Threading or comparative model

A typical threading or comparative modeling method consists of 4 steps:

1. Finding a similar structure called template.
2. Aligning the sequence with the template.
3. Building a model.
4. Assessing the model.

1. Finding a similar structure called template

The first step is to attempt to find related known protein structures in the Protein Data Bank for as many domains in the modeled sequence as possible (fold recognition or fold assignment). The folds of domains in the target sequence can be assigned by pairwise and multiple sequence similarity searches as well as by threading methods that rely explicitly on the known structures of the candidate template proteins.

2. Aligning the sequence with the template

While fold assignment predicts a structural relationship between two proteins, it does not produce an explicit 3-D model of the target sequence. Thus, fold assignment is generally followed by alignment of the target sequence with one or more template structures to establish the best possible correspondence between the residues in the target and template sequences. In the more difficult cases, semi-manual alignment of the whole family is necessary for the best results.

3. Building a model

After the alignment, the next step is comparative model building that relies on the alignment and the template structures to produce explicit 3-D models of the aligned domains of the target protein. These models usually consist of all non-hydrogen atoms for both the main chain and side chains, including the insertions and deletions relative to the template structures.

4. Assessing the model

Finally, the models need to be evaluated by considering structural and energetic criteria, not sequence similarity alone. Model evaluation helps to assess what information can be extracted from the model. If the model is unsatisfactory, it is possible to iterate through the cycle of fold assignment, alignment, modeling, and model evaluation in the search for a satisfactory model. In fact, a useful approach to fold assignment and alignment is to accept uncertain fold assignments and alignments, build a full atom comparative model of the target sequence, and make the final decision about whether or not the match and the alignment are accurate by evaluating the resulting comparative model.

The de novo/ab initio approach

Because in many cases, there are no suitable fold assignments, alignments and models obtained using the threading and comparative modeling approaches, the only recourse is the ab initio/de novo protein structure prediction methods that depend solely on the sequence of the protein to be modeled.

- **De novo methods are based on the following assumption:**

The native state of a protein is at the global free energy minimum and carry out a large-scale search of conformational space for protein tertiary structures that are particularly low in free energy for the given amino acid sequence.

- **Two key components of such methods:**

- a) The procedure for efficiently carrying out the conformational search.
- b) The free energy function used for evaluating possible conformations.

To allow rapid and efficient searching of conformational space, often only a subset of the atoms in the protein chain is represented, as opposed to the comparative methods in which virtually all atoms are included.

ROSETTA - the most successful de novo method:

Based on a picture of protein folding in which short segments of the protein chain flicker between different local structures consistent with their local sequence, and folding to the native state occurs when these local segments are oriented such that low free energy interactions are made throughout the protein.

In simulating this process, each short segment is allowed to sample the local structures adopted by the sequence segment in known protein structures, and a search is carried out through the combinations of these local structures for compact tertiary structures that bury the hydrophobic residues and pair the α -strands.

This strategy resolves some of the problems with both the conformational search and the free energy function: The search is greatly accelerated because switching between different possible local structures can occur in a single step, and fewer demands are placed on the free energy function because the use of fragments of known structures ensures that the local interactions are close to optimal.

Summary:

Many biopolymers spontaneously "fold" into characteristic shapes, which determine their biological functions and depend in a complicated way on their primary structures. Genome sequencing projects are producing linear amino acid sequences, but full understanding of the biological role of these proteins will require knowledge of their structure and function. The classical methods for structure analysis of proteins are X-ray crystallography and nuclear magnetic resonance (NMR). Unfortunately, these techniques are expensive and can take a long time (sometimes more than a year). Thus, computational methods which may give some indication of structure and/or function of proteins are becoming increasingly important. The three major computational approaches to this problem are 'comparative modelling', 'threading', and 'ab initio prediction'.

Model Questions:

1. Briefly explain the methods of secondary structure prediction of biopolymers along with flowsheets?

2. Outline the procedure and steps to predict the structure of an unknown protein sequence?

References:

1. Bioinformatics - Sequence and Genome Analysis- David W. Mount.
2. Developing Bioinformatics Computer Skills by Cynthia Gibas, Per Jambeck
3. Introduction to Bioinformatics by Arthur M. Lesk.
4. Ab Initio Methods for Protein Structure Prediction: A New Technique based on Ramachandran Plots by Anna Bernasconi and Alberto M. Segre
5. Protein Folding and Protein Structure Prediction by Ram Samudrala
6. Protein tertiary structure prediction: Basic concepts and applications by Tsaipei Wang & Jianzhong Shen

AUTHOR:

B.M.REDDY M.Tech. (HBTI, Kanpur)

Lecturer, Centre for Biotechnology

Acharya Nagarjuna University.

Lesson 3.5.4

Structural Optimization of Biopolymers

Structure

3.5.4.1 Introduction

3.5.4.2 Probability density functions

3.5.4.3 Protein structural alignment

3.5.4.4 Visualization of Structural Alignment

3.5.4.5 Comparison and Optimization

3.5.4.6 A Combinatorial and Global Optimization Approach

Summary

Model Questions

References

Objective:

This lesson explains the structure optimization techniques, which is an important aspect in structure prediction.

3.5.4.1 Introduction

Protein structure optimization is the process of bringing a structure into agreement with some "ideal" set of geometric parameters. When we discussed structure quality checking, protein structural models sometimes violate the laws of chemistry. Placing atoms too close together causes unfavorable intramolecular contacts, or vander Waals bumps. Bond lengths, bond angles, and dihedral angles between atoms in the protein can also be "wrong"; that is, they can fall outside some normal range of values expected for that type of bond or angle.

Structure optimization is an important issue not just to developers of theoretical models, but to researchers who experimentally determine protein structures. All protein atomic coordinates are, in an important sense, structural models. Structure optimization tools have long been part of the x-ray crystallographer's toolkit. The process of optimization can be computationally intensive. Because all atoms in a protein structure are connected by bonds with rigidly fixed lengths, moving an atom in one part of the protein structure has wide-ranging effects on its neighbors. Often moving one part of the protein into a better configuration means moving another part of the protein, into an unfavorable configuration. Optimization is, essentially, an iterative series of small changes designed to converge to the best overall result. There are many methods of optimization, which is its own subdiscipline within theoretical computer science.

You won't always need to know the particulars of optimization methods, but if you begin using structure optimization and molecular simulation methods frequently, you should be aware that your choice of optimization algorithms may be an issue. It's not always certain that optimization will provide you with a better structural model; if the method is based on incorrect structural rules, or if the rules are prioritized incorrectly, optimization can actually give you a worse model than you started with.

Informatics Plays a Role in Optimization

What are the "ideal" parameters or constraints used in optimization? In some cases, they are based entirely on chemical principles: bond lengths and angles determined by steric restrictions and nonbonded interactions described as Lennard-Jones potentials. In other cases, structural constraints are based on information derived from the database of known protein structures. If a particular amino acid in a particular sequence context always has the same conformation, a higher probability can be assigned to it assuming that conformation again, rather than a different conformation. Secondary structure prediction methods use an information based approach to predicting likely conformations for the protein backbone. Optimization methods use information to refine atomic structures at the level of individual side chain atoms once the backbone trace has been worked out.

Rotamer Libraries

Rotamer libraries are parameter sets specifically for the optimization of sidechain positions in molecular model building. They are called *rota mer* libraries because they contain information about allowed rotations of the remote amino acid side chain atoms around the C_α-C_β bond, expressed as the allowed values of side chain dihedral angles.

Because of steric constraints on bond rotation, amino acid side chains in proteins can assume only a few conformations without unfavorable energetic consequences. Rotamer libraries can be derived using chemical bond and angle constraints, but, they are more likely to be developed by analysis of the conformations assumed by amino acid side chains in known protein structures. Rotamer libraries can be either backbone-dependent or -independent. Backbone-independent rotamer libraries classify all instances of a particular amino acid as part of the same set, even if one occurrence is within a beta sheet and the other is within an alpha helix. Backbone-dependent rotamer libraries, on the other hand, further classify amino acids according to their occurrence in specific secondary structures:

SCRWL, available from the Fred Cohen research group at UCSF, is a program that allows you to model sidechain conformations using a backbone-dependent rotamer library.

3.5.4.2 Probability density functions

The derivation of probability density functions (PDFs) is similar in concept to the development of rotamer libraries, although more mathematically rigorous. The essence of a PDF is that a mathematical function is developed to represent a distribution of discrete values. The discrete values that make up the distribution are harvested from occurrences of a situation in a representative database of samples. That mathematical function can then be used to evaluate and optimize (and in some cases even predict) the properties of future occurrences of the same situation.

In protein modeling, PDFs have been used to describe intra- and inter-residue interatomic distances, as well as bond angles, dihedral angles, and other more spatially extensive regions of protein structure. Modeller, which *Predicting Protein Structure and Function from Sequence*, uses a combination of bond angle and dihedral angle PDFs to optimize the protein structure models it builds. Modeller's internal OPTIMIZE routine can be used for PDF-based structure optimization.

The data from which PDFs are generated can be broken down into specific occurrences; for example, all contacts between C_α in residue *i* and C_α in residue *i+4* when both residues are leucine but again, trade-offs between classification detail and class

population occur. Distance PDFs for proteins have been used by several groups to evaluate and optimize protein structures. Most such work is still in its early stages, and software isn't yet available for public use.

Figure 1 shows a plot of a distance probability density function for tertiary interactions between sulfur atoms in cysteine residues generated from known protein structures. The function's peak near 2 angstroms corresponds to the high propensity with which the sulfur atoms form disulfide bridges between cysteine residues. These data are taken from the Biology Workbench at the San Diego Supercomputer Center (<http://workbench.sdsc.edu/>) and plotted using *xmgr*. Note that the Workbench PDFs make a distinction between cysteine residues participating in disulfide bridges (pictured here and referred to as CSS residues at the Workbench site) and those cysteines that don't participate in disulfide bonds (which the Workbench site calls CYS).

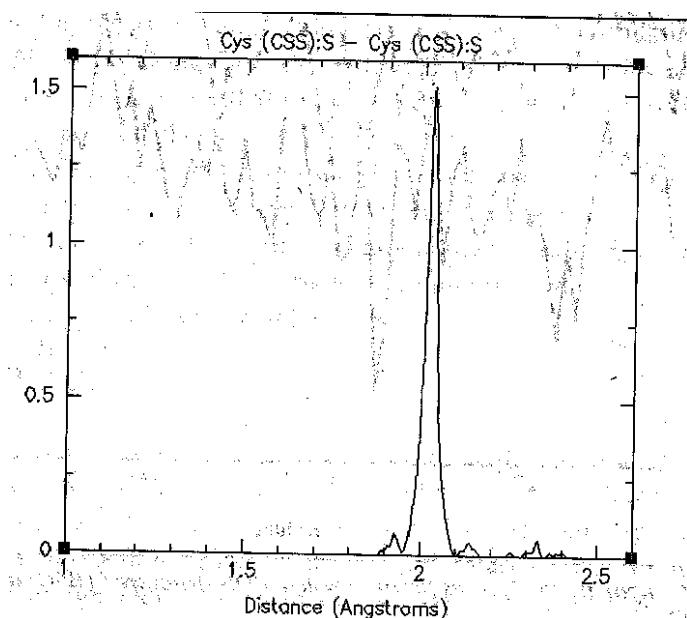


Figure 1 Interatomic distance probability density function

Structure evaluation based on PDFs is implemented in the Structure Tools section of the SDSC Biology Workbench (Figure 2). You can upload a PDB structure or a theoretical model and score the structure either on a residue-by-residue or an atom-by-atom basis. Scores can be displayed on a plot, where the Y-axis represents the relative probability of the region of structure that's being evaluated. This can be thought of in terms of the probability that a particular residue or atom is in the "correct" position, given what is known about other occurrences of that residue or atom in similar sequence environments. Regions with low probability are likely to be misfolded or poorly modeled. PDF probability scores can also be written out in a special PDB file, in place of the temperature factor values found in the original PDB file. These special PDB files can then be displayed using a visualization program such as RasMol or Chime. Coloring the molecule by temperature factor maps the PDF probability scores onto the molecular structure, highlighting regions of the structure that score poorly.

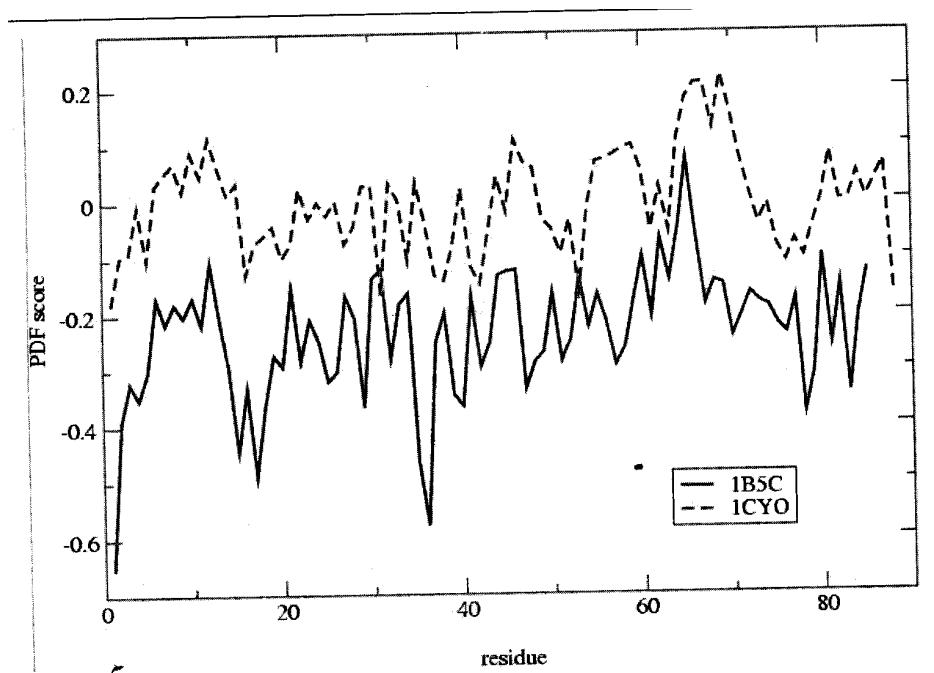
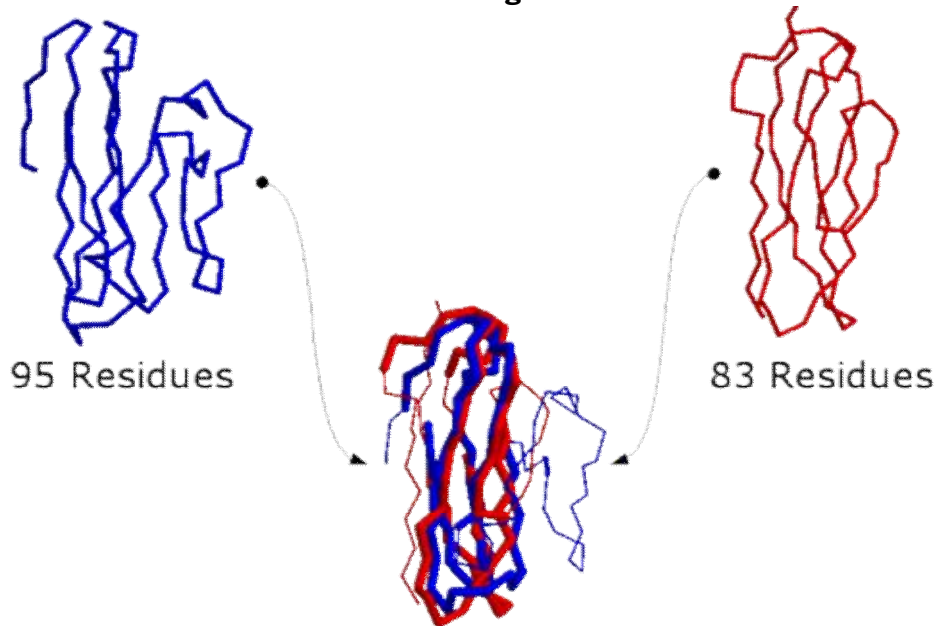


Figure 2. Comparing PDF scores/or an obsolete PDB structure (1B5C) and (1CYO) that superseded it

3.5.4.3 Protein structural alignment

Protein structural alignment is a form of alignment which tries to establish equivalences between two or more protein structures based on their fold. In contrast to simple structural superposition, where at least some equivalent residues of the two structures are known, structural alignment requires no a priori knowledge of equivalent positions. Structural alignment is a valuable tool for the comparison of proteins in the so called "twilight zone" and "midnight zone" of homology, where relationships between proteins can't be detected by sequence alignment methods. The method can therefore be used to establish evolutionary relationships between proteins that share no or nearly no common primary structure. This is especially important in the light of structural genomics and proteomics projects. The result of a structural alignment of two proteins is a superposition of their atomic coordinate sets with a minimal root mean square deviation (RMSD) between the two structures.

3.5.4.4 Visualization of Structural Alignment



Structural Alignment/Superimposition Schematic by **MAMMOTH** (Ortiz *et al.*) of two Immunoglobulin fold structures.

How similar are two Immunoglobulin structures? Use any one of the available structural alignment algorithms (*see Packages*) to superimpose two protein structures.

From structural alignment, you can extract percent of structural identity (PSI), structurally implied sequence alignment, root mean square deviation (RMSD), and a score of the alignment.

The PSI can be easily calculated by normalizing the number of aligned residues by the length of the shortest structure ($N / norm$) where N is the number of the corresponded residues that are within a Cartesian distance of 4 Å; and "norm" is the normalization factor. In the Immunoglobulin example the number of aligned residues within 4 Å is 57 and the norm of the set is 83, the PSI is therefore 68.67%.

Structurally implied sequence alignment is a one dimensional representation of the structural alignment.

RMSD is then calculated by using the distances between the corresponding residues in the alignment.

Algorithms

Up to now there is no definitive algorithmic solution to protein structural alignment. It could be shown that the alignment problem is NP-hard. All current algorithms employ heuristic methods. Therefore different algorithms may not produce exactly the same results for the same alignment problem.

Representation of structures

Protein structures have to be represented in some coordinate independent space to make them comparable. One possible representation is the so-called distance matrix, which is a two-dimensional matrix containing all pairwise distance between all C_{α}

atoms of the protein backbone. This can also be represented as a set of overlapping sub-matrices spanning only fragments of the protein. Another possible representation is the reduction of the protein structure to the level of secondary structure elements (SSEs), which can be represented as vectors, and can carry additional information about relationships to other SSEs, as well as about certain biophysical properties.

3.5.4.5 Comparison and Optimization

In the case of distance matrix representation, the comparison algorithm breaks down the distance matrices into regions of overlap, which are then again combined if there is overlap between adjacent fragments, thereby extending the alignment. If the SSE representation is chosen, there are several possibilities. One can search for the maximum ensemble of equivalent SSE pairs using algorithms to solve the maximum clique problem from graph theory. Other approaches employ dynamic programming or combinatorial simulated annealing.

Key map:

- **Ca** -- Backbone Atom (Ca) Alignment;
- **SSE** -- Secondary Structure Elements Alignment;
- **Pair** -- Pairwise Alignment (2 structures *only*);
- **Multi** -- Multiple Structure Alignment (MStA);
- **C-Map** -- Contact Map

3.5.4.6 A Combinatorial and Global Optimization Approach

The use of computational techniques to create peptide- and protein-based therapeutics is an important challenge in medicine. The ultimate goal, defined about two decades ago, is to use computer algorithms to identify amino acid sequences that not only adopt particular three-dimensional structures but also perform specific functions. To those familiar with the field of structural biology, it is certainly not surprising that this problem has been described as "inverse protein folding". That is, while the grand challenge of protein folding is to understand how a particular protein, defined by its amino acid sequence, finds its unique three-dimensional structure, protein design involves the discovery of sets of amino acid sequences that form functional proteins and fold into specific target structures.

Experimental, computational, and hybrid approaches have all contributed to advances in protein design. Applying mutagenesis and rational design techniques, for example, experimentalists have created enzymes with altered functionalities and increased stability. The coverage of sequence space is highly restricted for these techniques, however. An approach that samples more diverse sequences, called directed protein evolution, iteratively applies the techniques of genetic recombination and in vitro functional assays. These methods, although they do a better job of sampling sequence space and generating functionally diverse proteins, are still restricted to the screening of 10^3 - 10^6 sequences.

Challenges of Generic Computational Protein Design

The limitations of experimental techniques serve to highlight the importance of computational protein design. Practically speaking, in silico methods can sample astronomically large numbers of sequences; the resulting diversity in the selected sequences leads to a much broader spectrum of functional proteins. Computational

methods have already been used successfully to alter existing proteins so that they have better stability and functionality, and to combine or modify proteins for aggregate functionality. The ultimate goal is the *de novo* computational design of proteins—that is, a systematic way to create proteins that have both new structural templates and better properties.

The success of an approach to computational protein design depends on two main ingredients: (i) the method used to search sequence space, and (ii) the principles on which the modeling is based. To better understand the combinatorial nature of a search through sequence space, consider a relatively small protein, 50 residues long. Allowing for any one of the 20 possible amino acids at each residue position of this protein results in 20^{50} , or more than 10^{65} possible amino acid sequences. Clearly, clever optimization techniques are needed to deal with this level of combinatorial complexity. Although stochastic methods have been used, the first successful computational design of a full protein was achieved with a deterministic branch-and-bound technique based on the dead-end elimination theorem. Even in the case of dead-end elimination, however, heuristics must be incorporated to make convergence reasonably fast for large proteins.

Given a method that can effectively search through such large numbers of sequences, the question becomes one of distinguishing between protein sequences. In other words, what is the target function that we would like to optimize? Of course, this will depend on the representation of the protein. Initial efforts focused only on the replacement of core residues, a condition under which the steric van der Waals and hydrophobic forces are expected to dominate. As computational protein design has been extended to full proteins, requiring the addition of hydrogen bonding and solvent and electrostatic effects in various forms and flavors, the models have become more and more complex. Overall, there is no consensus among these models, and it is unclear which methods are more valid and suitable for generic computational protein design. More fundamental concerns still to be addressed include the realization that imposing a rigid template is a severe constraint.

Our recent efforts in the area of computational protein design consist of two separate stages: (i) *in silico* sequence selection, and (ii) validation of fold stability and specificity. In stage (i) we use a novel mixed-integer formulation that incorporates amino acid side-chain specificity to model sequence space. We devised a method for solving this new mixed-integer optimization problem; the solutions provide a set of candidate sequences for input to stage (ii). In stage (ii) we study the reduced set of sequences in more detail, using the principles of AS TROFOLD, a method for *ab initio* prediction of three-dimensional protein structures within a combinatorial and global optimization framework. The approach allows backbone flexibility, and the final results reflect a quantitative ranking of fold stability and specificity for each amino acid sequence. Figure 1 depicts the full computational design approach.

Modeling Sequence Space

The formulation of stage (i) depends on the representation of the protein system. Initially, rather than describe the amino acids by the coordinates of all atoms, we describe the backbone template only by the coordinates of the alpha-carbon atoms. A pairwise distance-dependent interaction potential is used to calculate the energy of an amino acid sequence on this template. The statistically based energy function assigns

energy values according to the alpha-carbon separation distance for each pair of amino acids. Similar structure-based interaction potentials have been used in fold recognition and fold prediction.

The advantages of this representation are the simplicity of the model and the robustness of the system with respect to the rigid backbone approximation. In other words, while the interaction potential implicitly includes amino acid and side-chain specificity, its coarseness allows for an inherent flexibility in the backbone. For the interaction potential used in our study, alpha-carbon distances are discretized into a set of 13 bins, with the 2730 parameters of the model being derived from the solution to a linear optimization formulation that favors native folds over decoy structures.

An explanation of the development of the new mixed-integer formulation begins with a description of the variable set over which the energy function is optimized. First, consider the set $i = 1, \dots, n$, which defines the residue positions along the backbone. At each position i there can be a set of amino acid substitutions represented by $y_{ij} \{i\} = 1, \dots, m_i$ where, for the general case $m_i = 20 \forall i$. The equivalent sets $k = i$ and $l = j$ are defined, and k must be greater than i to ensure that only unique pairwise interactions are represented. Binary variables y_{ij} and y_{kl} are introduced to indicate the possible substitutions at a given position. That is, y_{ij} indicates the amino acid j at a position i in the sequence by taking the value of 1 for one amino acid and 0 for all others. The formulation, in which the goal is to minimize the energy according to the pairwise interaction parameters that multiply the binary variables, can then be expressed as:

$$\begin{aligned} & \min_{y_i^j, y_k^l} \sum_{i=1}^n \sum_{j=1}^{m_i} \sum_{k=i+1}^n \sum_{l=1}^{m_k} E_{ik}^{jl}(x_i, x_k) y_i^j y_k^l \\ & \text{subject to } \sum_{j=1}^{m_i} y_i^j = 1 \quad \forall i \\ & y_i^j, y_k^l = 0 - 1 \quad \forall i, j, k, l. \end{aligned}$$

The parameters $E_{ik}^{jl}(x_i, x_k)$ depend on the distance between the alpha-carbons at the two backbone positions (x_i, x_k) , as well as on the specific amino acids at those positions. The composition constraints require that at most one amino acid appear at each position. Notice that the binary variables appear as bilinear combinations in the objective function. Fortunately, this objective can be reformulated as a strictly linear (integer linear programming) problem :

$$\begin{aligned} & \min_{y_i^j, y_k^l} \sum_{i=1}^n \sum_{j=1}^{m_i} \sum_{k=i+1}^n \sum_{l=1}^{m_k} E_{ik}^{jl}(x_i, x_k) w_{ik}^{jl} \\ & \text{subject to } \sum_{j=1}^{m_i} y_i^j = 1 \quad \forall i \\ & y_i^j, y_k^l - 1 \geq w_{ik}^{jl} \leq y_i^j \quad \forall i, j, k, l \\ & 0 \leq w_{ik}^{jl} \leq y_k^l \quad \forall i, j, k, l \\ & y_i^j, y_k^l = 0 - 1 \quad \forall i, j, k, l. \end{aligned}$$

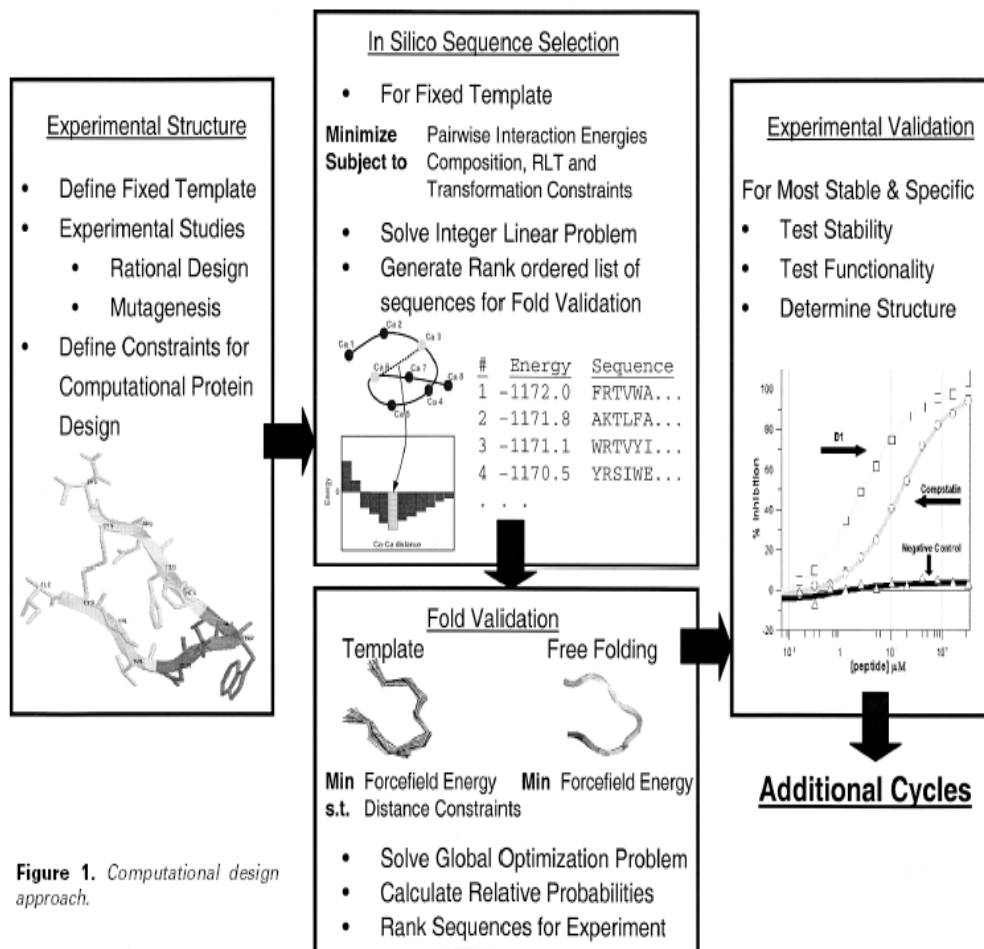


Figure 1. Computational design approach.

This reformulation relies on the transformation of the bilinear combinations into a new set of linear variables w_{ijkl} , while the addition of the four sets of constraints provides the equivalence to the original formulation.

Although the integer linear programming problem can be solved by standard branch-and-bound techniques, convergence is prohibitively slow for large systems. An important finding is that the performance of the branch-and-bound algorithm improves significantly when the principles of reformulation linearization techniques (RLT) are applied. The basic strategy is to multiply appropriate constraints by bounded non-negative factors and then replace the products of the original variables by new variables; in this way, we derive higher-dimensional lower-bounding linear programming relaxations to the original problem. The tighter LP relaxations are included in the course of the overall branch-and-bound search and speed convergence to the global minimum.

Application of the RLT approach to the composition constraint begins with a reformulation of the equations; we form the product of the constraint equations with some binary variables (or their complements). For example, by multiplying the

composition constraint by the set of variables y_{kl} , we produce the following additional set of constraints $\forall j, k, l$:

$$y_k^l \sum_{i=1}^{m_i} y_i^j = y_k^l \quad \forall j, k, l.$$

The variable substitution already introduced to linearize the objective function can now be used to linearize equation (3). The set of RLT constraints becomes:

These additional constraints are added to the formulation given by (2). It is then straightforward to identify a rank-ordered list of the low-energy sequences through the introduction of integer cuts and repetitive solution of the integer linear programming problem.

Predicting 3D Protein Structures

Once a set of sequences has been identified in stage (i), we proceed to stage (ii), using a flexible template to rigorously quantify the stability and specificity of each sequence. The approach is based on the generation of all atomistic structural ensembles for the selected sequences under two sets of conditions. Under the first, the structures are constrained to vary, with some imposed fluctuations, around the template structure; under the second, free folding calculations are performed. The formulations are reminiscent of the structure-prediction problems in protein folding. Specifically, the problems are formulated as constrained global optimizations of a detailed atomistic energy forcefield E_{ff} over a set of internal coordinates ϕ , which describe any conformation of the system. The bounds on these variables are enforced by simple box constraints. Finally, a set of constraints, $\mathcal{E}^r, r = 1, \dots, N_R$, which are nonconvex in the internal coordinate space, can be used to constrain interatomic distances. The formulation is represented by the following set of equations:

$$\begin{aligned} & \min_{\phi} E_{ff} \\ & \text{subject to } E_r^{dis}(\phi) \leq 0 \quad r = 1, \dots, N_R \\ & \phi_s^L \leq \phi_s \leq \phi_s^U \quad s = 1, \dots, N_{\phi}. \end{aligned}$$

Here, $s = 1, \dots, N_{\phi}$ corresponds to the set of internal coordinates ϕ_s , with ϕ_s^L and ϕ_s^U representing lower and upper bounds on these variables. The forms of the distance constraints and the forcefield energy function E_{ff} are completely general. In practice, we use square-well quadratic functions for the distance constraints, and an atomistic-level forcefield that includes van der Waals, hydrogen bonding, and electrostatic and torsional terms for the objective.

These formulations belong to the class of general nonconvex constrained global optimization problems, and are solved via the principles of an α BB deterministic global optimization approach, a branch-and-bound method applicable to the identification of the global minimum of nonconvex optimization problems with twice-differentiable functions. In addition to identifying the global minimum-energy conformation, this global optimization approach has been adapted to locate an ensemble of low-energy

conformations. These ensembles are used to quantify the fold stability and specificity by summing the statistical weights for the conformers from the free-folding simulation that resemble the template structure, and dividing this sum by the total partition function; that is, statistical weights are summed for all conformers from the free-folding simulation. The analysis is an unambiguous method for ranking the fold stability and specificity among a set of different amino acid sequences. The approach described here has been successfully tested on the design of improved analogs for Compstatin, a synthetic therapeutic peptide that prevents the autoimmune-mediated damage of organs during transplantation and in various inflammatory diseases. The new computational design approach yielded a version of Compstatin seven times more efficacious and stable than the original peptide (see Figure 2 for a summary of the results). The result is a significant improvement over analogs identified by either purely rational or experimental combinatorial design techniques.

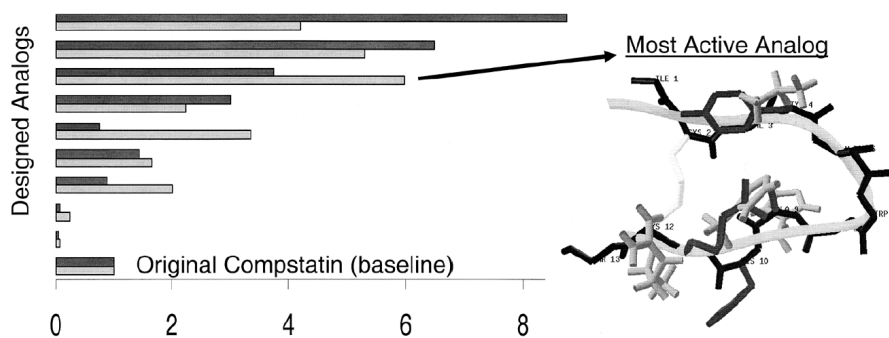


Figure 2. Stability and activity relative to the synthetic therapeutic peptide Compstatin. Experimental results are in light gray; predicted results are in dark gray.

Summary:

Protein structure optimization is the process of bringing a structure into agreement with some "ideal" set of geometric parameters. Structure optimization is an important issue not just to developers of theoretical models, but to researchers who experimentally determine protein structures. Modeller, which *Predicting Protein Structure and Function from Sequence*, uses a combination of bond angle and dihedral angle PDFs to optimize the protein structure models it builds.

Model Questions:

1. Outline the procedure for the structural optimization of biopolymers?
2. Write about the PDFs and Explain how they can help us in structure optimization of proteins?

References:

1. Developing Bioinformatics Computer Skills by Cynthia Gibas, Per Jambeck
2. In Silico Protein Design: A Combinatorial and Global Optimization Approach By John L. Klepeis and Christodoulos A. Floudas