UNIT-IV

Protein Databases on the Internet

Protein databases have become a crucial part of modern biology. Huge amounts of data for protein structures, functions, and particularly sequences are being generated. These data cannot be handled without using computer databases. Searching databases is often the first step in the study of a new protein. Without the prior knowledge obtained from such searches, known information about the protein could be missed, or an experiment could be repeated unnecessarily. Comparison between proteins and protein classification provide information about the relationship between proteins within a genome or across different species, and hence offer much more information than can be obtained by studying only an isolated protein. In this sense, protein comparison through databases allows one to view life as a forest instead of individual trees. In addition, secondary databases derived from experimental databases are also widely available. These databases reorganize and annotate the data or provide predictions. The use of multiple databases often helps researchers understand evolution, structure, and function of a protein.

Protein databases are especially powered by the Internet. Unlike traditional media, such as the CD-ROM, the Internet allows databases to be easily maintained and frequently updated with minimum cost. Researchers with limited resources can afford to set up their own databases and disseminate their data quickly. Notably, many small databases on specific types of proteins, such as the EF-Hand Calcium-Binding Proteins Data Library (http://structbio.vanderbilt.edu/cabp_database/), are widely available. Users worldwide can easily access the most up-to-date version through a user-friendly interface. Most protein databases have interactive search engines so that users can specify their needs and obtain the related information interactively. Many protein databases also allow submitters to deposit data, and database servers can check the format of the data and provide immediate feedback.

Although some protein databases are widely known, they are far from being fully utilized in the protein science community. This unit provides a starting point for readers to explore the potential of protein databases on the Internet. Databases for different aspects of proteins are discussed with the focus on sequence, structure, and family. The strengths and weaknesses of the databases are addressed. For Web addresses of the databases discussed in this unit, see Internet Resources and Table 19.4.1. From hundreds of on-line protein databases, several major databases are discussed as examples to illustrate their features and how they can be used effectively. Most other protein databases can be explored in a similar way.

Web Addresses and Sizes of Selected Protein Databases

PROTEIN SEQUENCE DATABASES

Thanks to the Human Genome Project and other sequencing efforts, new sequences have been generated at a prodigious rate. These sequences provide a rich information source and are the core of the revolutionary movement toward “large-scale biology.” The protein sequences can be computationally annotated from these genomic sequences. Various databases contain protein sequences with different focuses. Among all protein sequence databases, UniProt (UniProt Consortium, 2011) is the most widely used one. It provides more annotations than any other sequence database with a minimal level of redundancy through human input or integration with other databases. UniProtKB has three components: (1) Protein knowledgebase, including Swiss-Prot (manually annotated and reviewed) and TrEMBL (automatically annotated) (Bairoch and Apweiler, 1999); (2) UniRef (sequence clusters for fast sequence similarity searches); and (3) UniParc (sequence archive for keeping track of sequences and their identifiers). In addition to Swiss-Prot and TrEMBL, UniProtKB includes information from Protein Sequence Database (PSD) in the Protein Identification Resource (PIR; Barker et al., 1999), which builds a complete and non-redundant database from a number of protein and nucleic acid sequence databases together with bibliographic and annotated information. The National Center for Biotechnology Information (NCBI; http://www.ncbi.nlm.nih.gov) also provides rich information and a number of useful tools for protein sequences. For example, the nr protein database is used for BLAST search (Altschul et al., 1997), which is described in UNIT 2.5 of this book. It includes entries from the non-redundant GenBank (Benson et al., 1999) translations, UniProt, PIR, Protein Research Foundation (PRF) in Japan, and the Protein Data Bank (PDB). Only entries with absolutely identical sequences are merged.

Most of the sequence databases have a sequence search tool and cross-references to entries of other protein and gene databases. Many sequence databases, such as UniProt, also provide text searching using, for instance, protein names or key words. To study a new protein, the author recommends first performing a sequence search using BLAST in nr if the protein sequence is available. The search often gives entry names in the protein databases included in nr. Even when the protein is not found in nr, it is likely that a homologous protein will be hit, which can often lead to some useful information, such as the function of the query protein. If the sequence of the query protein is unavailable, doing a text search in UniProt usually identifies the protein. UniProt is probably the place to obtain the most information about a protein if it can be found in UniProt. However, some additional information may be found by checking other sequence databases. For example, the Kyoto Encyclopedia of Genes and Genomes (KEGG; Ogata et al., 1999) annotates some gene entries with information about metabolic and regulatory pathways. One can also study proteins based on gene models (predicted protein sequences) from many species-specific genome resources, such as Mouse Genome Database (MGD, http://www.informatics.jax.org), FlyBase (a resource for Drosophila genes, http://flybase.org), WormBase (a resource for C. elegans, http://www.wormbase.org), Saccharomyces Genome Database (SGD, http://www.yeastgenome.org), Arabidopsis Information Resource (TAIR, http://www.arabidopsis.org), and Soybean Knowledge Base (SoyKB, http://soykb.org). Although predicted sequences generated by computational gene-finding tools in these resources may contain errors, a large number of proteins are covered and are often reliable enough to provide useful information. When the protein of interest is from a species that is not covered by any of these databases, it is likely that some information can be retrieved from its homolog of a model organism in one of the databases.

UniProt, as a curated protein sequence database, offers a portal to a wide range of annotations, covering areas such as function, family, domain parsing, post-translational modifications, and variants. UniProt can be accessed at http://www.uniprot.org.

Human vitronectin is used here as an example for searching protein sequence databases. To locate the UniProt entry for this protein, one can search either the entry name (VTNC_HUMAN) or the accession number (P04004) obtained from a BLAST search. Alternatively, one can use the full-text search at the UniProt Web page to search by protein name (human vitronectin) or key words (e.g., serum spreading, as vitronectin is also called serum spreading factor s-protein). A combination of several entries can be used in a search.

The entry name in UniProt has the general format X_Y, where X is a mnemonic code of up to four characters indicating the protein name (in this case, VTNC), and Y is a mnemonic species identification code of up to five characters for the biological source of the protein. Some codes used for Y are full English names, e.g., HORSE, HUMAN, MAIZE, MOUSE, PIG, RAT, SHEEP, YEAST (baker’s yeast, Saccharomyces cerevisiae), and WHEAT. Some are abbreviations, including BOVIN (bovine), CHICK (chicken), ECOLI (Escherichia coli), PEA (garden pea, Pisum sativum), RABIT (rabbit), SOYBN (soybean, Glycine max), and TOBAC (common tobacco, Nicotina tabacum).

An entry name may have several accession numbers if they have been merged. An accession number is always conserved from release to release and, therefore, allows unambiguous citation.

Each entry contains the following items shown in table format in the NiceProt View layout: (1) name and origin, (2) protein attributes, (3) general annotation (comments), (4) ontologies (gene functions), (5) binary protein-protein interactions, (6) sequence annotation (features), (7) sequence, (8) references (literature citation), (9) web resources, (10) cross-references (links to other databases), (11) entry information, and (12) relevant documents. The text in the general annotation entry provides a function annotation for the protein (e.g., “Vitronectin is a cell adhesion and spreading factor found in serum and tissues. Vitronectins interact with glycosaminoglycans and proteoglycans…”). The “Cross-references” entry lists the annotations of the protein by other databases, such as GeneCards (Rebhan et al., 1998) and InterPro (Apweiler et al., 2001). GeneCards, a database of human genes, shows chromosomal location and the involvement of the protein in certain diseases (if applicable). InterPro contains predictive protein “signatures”, such as functional domains, repeats and important sites. Clicking the link to InterPro from UniProt leads to a nice graphic view for domain parsing, as shown in Figure 19.4.1 for vitronectin.

Annotation of human vitronectin by InterPro.
Various research results are given under sequence annotation (features). Some of the sample features items for VTNC_HUMAN are as follows:

Feature key	Position (s)	Length	Description
Signal peptide	1–19	19	Ref.8 Ref.9
Chain	20–398	379	Vitronentin V65 subunit
Peptide	20–63	44	Somatomedin-B (Ref. 8)
Domain	161–204	44	Hemopexin-like 1
Motif	64–66	3	Cell attachment site
Site	398–399	2	Cleavage.
Modified residue	75	1	Sulfotyrosine (Ref. 22)
Glycosylation	86	1	N-linked (GlcNAc…)
Disulfide bond	24 ← → 40		Alternative (by similarity)
Natural variant	122	1	A→S.[dbSNP:rs2227741]
Sequence conflict	50	50	C → N AA sequence

Here, “peptide” represents an active peptide in the mature protein, “modified residue” indicates a post-translationally modified residue, and “sequence conflict” shows that different papers report differing sequences.

Go to:

PROTEIN STRUCTURAL DATABASES

Searching structure databases is becoming more and more popular in molecular biology. The three-dimensional structures of proteins not only define their biological functions, but also hold a key in rational drug design. Traditionally, protein structures were solved at a low-throughput mode. However, advances in new technologies, such as synchrotron radiation sources and high-resolution nuclear magnetic resonance (NMR), accelerate the rate of protein structure determination substantially. The only international repository for the processing and distribution of protein structures is the PDB (Bernstein et al., 1977). The structures in the PDB were determined experimentally by X-ray crystallography, NMR, electron microscopy, etc. Theoretical models have been removed from PDB, effective July 2, 2002, based on the new PDB policy. The PDB also contains some structures of chemical ligands and nucleotides. Each PDB entry is represented by a four-character identifier (PDB ID), where the first character is always a number from 0 to 9 (e.g., 1cau, 256b). The PDB can be accessed at http://www.rcsb.org/pdb/or http://www.pdb.org.

The PDB offers a broad range of search methods, from PDB ID and keywords to structural features and binding ligands. The PDB stores structural information in two formats: the PDB file format (Bernstein et al., 1977) and the macromolecular crystallographic information file (mmCIF) format (Bourne et al., 1997). The PDB file format is still the dominant format used in the protein community. It contains three parts: annotations, coordinates, and connectivities. The connectivity part, which shows chemical connectivities between atoms, is optional. It is listed at the end of the PDB file, beginning the line with the key word CONECT. The coordinate part uses each line for a three-dimensional coordinate of an atom, starting from ATOM (for standard amino acids) or HETATM (for nonstandard groups). The following shows an example of the PDB file format:

```
HEADER    OXIDOREDUCTASE      (OXYGEN(A))     14-JUN-89
          1GOX 1GOX 3
COMPND    GLYCOLATEOXIDASE    (E.C.1.1.3.1)  1GOX 4
…
ATOM232   N       ALA   29      54.035 4.332   19.352 1.00    23.93   1GOX 374
ATOM233   CA      ALA   29      52.992 65.356 19.569 1.00    24.74   1GOX 375
ATOM234   C       ALA   29      53.519 66.762 19.309 1.00    25.43   1GOX 376
ATOM235   O       ALA   29      54.648 67.179 19.655 1.00    25.66   1GOX 377
ATOM236   C       BALA 29       52.433 65.340 20.993 1.00    24.54   1GOX 378
…
HETATM    3165    O       HOH   658     62.480 62.480 0.000   0.50    65.79N1GOX
          3170
```

```
CONECT      2837   2838   2854                                         1GOX
        3171
```

Each line shows the atom serial number, atom type, residue type, chain identifier (in case of multi-chain structure), residue serial number, orthogonal coordinates (three values), occupancy, temperature factor, and segment identifier.

The annotation part of the PDB file format contains dozens of possible record types, including: HEADER (name of protein and release date), COMPND (molecular contents of the entry), SOURCE (biological source), AUTHOR (list of contributors), SSBOND (disulfide bonds), SLTBRG (salt bridges), SITE (groups comprising important sites), HET (nonstandard groups or residues [heterogens]), MODRES (modifications to standard residues), SEQRES (primary sequence of backbone residues), HELIX (helical substructures), SHEET (sheet substructures), and REMARK (other information and comments).

The PDB allows a user to view a molecule structure interactively through Jmol (Hanson, 2010), PDB SimpleViewer, PDB ProteinWorkshop, and RCSB-Kiosk, when the browser is configured to support these free rendering tools. The PDB provides related information about the protein, such as secondary structure assignment and geometry. Each PDB entry also links to a wide range of annotations from secondary databases, including (1) summary and display databases such as Structural Biology Knowledgebase (SBKB, http://sbkb.org), PISA (Protein Interfaces, Surfaces and Assemblies; Krissinel and Henrick, 2007), Molecular Modelling Database (MMDB; Marchler-Bauer et al., 1999) in Entrez, PDBsum (Laskowski et al., 1997), Jena Library of Biological Macromolecules (JenaLib, http://www.fli-leibniz.de/IMAGE.html), PDBWiki (a community annotated knowledge base of biological molecular structures, http://pdbwiki.org), and Proteopedia (a collaborative 3D-encyclopedia of proteins and other molecules; Prilusky et al., 2011); (2) domain annotation from SCOP (Murzin et al., 1995), CATH (Orengo et al., 1997), and Pfam (Finn et al., 2010); (3) structure comparison to other proteins using various methods; (4) the MEDLINE bibliography; (5) protein movements recorded in Database of Macromolecular Movement (MolMovDB; Gerstein and Krebs, 1998); and (6) geometry analyses of the protein, such as CSU Contacts of Structural Units (Sobolev et al., 1999) and castP Identification of Protein Pockets & Cavities (Liang et al., 1998).

In addition to PDB and its linking databases, other structure-related databases can also provide useful information. For example, pdbLight (http://mufold.org/pdblight.php) integrates protein sequence and structure data from multiple sources for protein structure prediction and analysis, together with predicted SCOP classification for the weekly updated PDB structures. BioMagResBank (BMRB; University of Wisconsin, 1999) is a repository for NMR spectroscopy data on proteins, peptides, and nucleic acids. Particularly, it provides partial NMR data (e.g., chemical shifts) before the full structure is solved. Protein Model

Portal (PMP; Arnold et al., 2009) provides predicted structural models and their quality assessments for a large number of proteins.

Go to:

PROTEIN FAMILY DATABASES

Introduction

Proteins can be classified according to their sequence, evolutionary, structural, or functional relationships. A protein in the context of its family is much more informative than the single protein itself. For example, residues conserved across the family often indicate special functional roles. Two proteins classified in the same functional family may suggest that they share similar structures, even when their sequences do not have significant similarity.

There is no unique way to classify proteins into families. Boundaries between different families may be subjective. The choice of classification system depends in part on the problem; in general, the author suggests looking into classification systems from different databases and comparing them. Three types of classification methods are widely adopted based upon the similarity of sequence, structure, or function. Sequence-based methods are applicable to any proteins whose sequences are known, while structure-based methods are limited to the proteins of known structures, and function-based methods depend on the functions of proteins being annotated. Sequence- and structure-based classifications can be automated and are scalable to high-throughput data, whereas function-based classification is typically carried out manually. Structure- and function-based methods are more reliable, while sequence-based methods may result in a false positive result when sequence similarity is weak (i.e., two proteins are classified into one family by chance rather than by any biological significance). In addition, since protein structure and function are better conserved than sequence, two proteins having similar structures or similar functions may not be identified through sequence-based methods.

Databases for Sequence-Based Protein Families

Sequence-based protein families are classified according to a profile derived from a multiple-sequence alignment. The profile can be shown across a long domain (tens of residues or more) or can be revealed in short sequence motifs. Classification methods based on profiles across long domains tend to be more reliable but less sensitive than those based on short sequence motifs.

Several sequence-based methods focus more on profiles across long domains, including Pfam (Finn et al., 2010), ProDom (Corpet et al., 1999), and Clusters of Orthologous Group (COG; Tatusov et al., 1997). These methods differ in the techniques used to construct families. Pfam builds multiple-sequence alignments of many common protein domains using hidden Markov models. The ProDom protein domain database consists of homologous domains based on recursive PSI-BLAST searches (UNIT 2.5). COG aims toward finding ancient conserved domains by delineating families of orthologs across a wide phylogenetic range. SMART (Simple Modular Architecture Research Tool; Letunic et al., 2009) collects domain families, which are annotated with respect to phyletic distributions, functional class, three-dimensional structures and functionally important residues. It can be used for identification and annotation of genetically mobile domains and analysis of domain architectures. The iProClass database (Wu et al., 2004) combines multiple sources of information for protein classification. One can use all these databases for a comprehensive analysis or choose one of them based on the purpose of the study. Various sequence-based protein families have different focuses. For example, Pfam focuses on function, ProDom on sequence domain, and COG on evolution.

The following shows an example of Pfam for the GRIP domain (accession number PF01465). Pfam lists some useful functional information for the entry as follows:

"The GRIP (golgin-97, RanBp2alpha, Imh1p and p230/golgin-245) domain is found in many large coiled-coil proteins. It has been shown to be sufficient for targeting to the Golgi. The GRIP domain contains a completely conserved tyrosine residue. At least some of these domains have been shown to bind to GTPase Arl1, see structures in [4,5]."

In addition, Pfam gives the alignment among the family members.

One can identify some features of the family through this pattern (i.e., from particularly conserved residues at specific alignment positions).

Some methods are based on "fingerprints" of small conserved motifs in sequences, as with PROSITE (Hofmann et al., 1999), PRINTS (Attwood et al., 1999), and BLOCKS (Heniko et al., 1999). In protein sequence families, some regions have been better conserved than others during evolution. These regions are generally important for the function of a protein or for the maintenance of its three-dimensional structure or function. The fingerprints may be used to assign a newly sequenced protein to a specific family. Fingerprints are derived from gapped alignments in PROSITE and PRINTS, but are derived from ungapped alignments (corresponding to the highly conserved regions in proteins) in BLOCKS. A fingerprint in PRINTS may contain several motifs from PROSITE, and thus may be more flexible and powerful than a single PROSITE motif. Therefore, PRINTS can provide a useful adjunct to PROSITE. It should be noted that some functionally unrelated proteins may be classified together due to chance matches in short motifs.

Databases for Structure-Based Protein Families

The hierarchical relationship among proteins can be clearly revealed in structures through structure-structure comparison. Structure families often provide more information on the relationship between proteins than what sequence families can offer, particularly when two proteins share a similar structure but no significant sequence identity. Figure 19.4.2 shows an example of a structure-structure alignment between two proteins. Sometimes, sequence similarity between two proteins exists but is not strong enough to produce an unambiguous alignment. In this case, the alignment between two structures can generate better alignment in terms of biological significance, and thus may pinpoint the evolutionary relationship and active sites more accurately.

Figure 19.4.2

Figure 19.4.2

Structure superposition between glycolate oxidase(1gox, in black) and inosine monophosphate dehydrogenase (1ak5, in gray). This figure was made using MOLSCRIPT (Kraulis, 1991).

Different structure-structure comparison methods yield different structure families. CATH (Class, Architecture, Topology and Homologous superfamily; Orengo et al., 1997) is a hierarchical classification of protein domain structures. CE (Combinatorial Extension of the optimal path; Shindyalov and Bourne, 1998) provides structural neighbors of the PDB entries with structure-structure alignments and three-dimensional superposition. FSSP (Fold classification based on Structure-Structure alignment of Proteins; Holm and Sander, 1996) features a protein family tree and a domain dictionary, in addition to whole-chain-based classification, sequence neighbors, and multiple structure alignments. SCOP (Structural Classification of Proteins; Murzin et al., 1995) uses augmented manual classification, class, fold, superfamily, and family classification. VAST (Vector Alignment Search Tool; Gibrat et al., 1996) contains representative structure alignments and three-dimensional superposition. Among these five databases, SCOP provides more function-related information. However, due to the manual work involved, SCOP is not updated as frequently as the others (as of September 2011, it was last updated for the PDB release on June, 2009), whereas FSSP and CATH follow the PDB updates closely.

SCOP is used here as an example to show the features of structure-based families. SCOP can be accessed through its home server in the UK (http://scop.mrc-lmb.cam.ac.uk/scop/). SCOP describes the hierarchical relationship among proteins through the major levels of (homologous) family, superfamily, and fold. Proteins are clustered together into a (homologous) family if they have significant sequence similarity. Different families that have low sequence similarity but whose structural and functional features suggest a common evolutionary origin are placed together in a superfamily. Different superfamilies are categorized into a fold if they have the same major secondary structures in the same arrangement and with the same topological connections (the peripheral elements of secondary structure and turn regions may differ in size and conformation). Two superfamilies in the same fold may not have a common evolutionary origin. Their structural similarities may arise from the physics and chemistry of proteins favoring certain packing arrangements and chain topologies (Murzin et al., 1995). Figure 19.4.3 shows the SCOP interface using an example of protein 1gox in the PDB.

Figure 19.4.3
Figure 19.4.3
An example of the SCOP interface when searching the structure of 1gox in the PDB.

Databases for Function-Based Protein Families

There are various protein functional families classified from different perspectives. The ENZYME data bank (Bairoch, 1993) contains the following data for each enzyme: EC number, recommended name, alternative names, catalytic activity, cofactors, pointers to the UniProt entry, and pointers to any disease associated with a deficiency of the enzyme. BRENDA (Scheer et al., 2011) collects extensive enzyme functional data. Catalytic Site Atlas (Porter et al., 2004) is a database of three-dimensional enzyme active sites derived from PDB structures. Various gene ontologies, such as Gene Ontology (GO; The Gene Ontology Consortium, 2000) and KEGG, also organize proteins into functional categories. Annotation and analysis by these ontologies for a given list of genes can be carried out using tools and databases such as DAVID (Database for Annotation, Visualization and Integrated Discovery; Huang et al., 2009). In addition, there are a growing number of databases dedicated to special types of proteins, such as G-protein-coupled receptors, transporters, and protein kinases, as shown in Table 19.4.1.

Go to:

OTHER DATABASES

Protein Modification Databases

There are a number of databases for protein post-translational modifications. O-GlycBase (Gupta et al., 1999) collected, experimentally verified O- or C-glycosylation sites. Plant Protein Phosphorylation Database (P3DB; Gao et al., 2009) condenses phosphoproteomics information (including experimental phosphorylation sites) from various plants. Compendium of protein lysine acetylation (CPLA; Liu et al., 2010) includes manually curated lysine acetylated substrates with their sites.

Protein Localization Databases

A number of databases are available to describe protein subcellular localization and targeting. These databases are for various species, such as eSLDB (eukaryotic Subcellular Localization database) for general eukaryotes (Pierleoni et al., 2007), LOCATE for human and minor (Sprenger et al., 2008), SUBA for Arabidopsis (Heazlewood et al., 2007), and PSORTdb for bacteria and archaea (Yu et al., 2011). Some databases focus on special organelles, such as Organelle DB (Wiwatwattana and Kumar, 2005) and Centrosome:db (Nogales-Cadenas et al., 2009).

Protein Binding Databases

Protein binding includes protein-substrate docking and protein-protein association. ReLiBase (Hendlich, 1998) is a database system for analyzing receptor-ligand complexes in the PDB. BindingDB (Liu et al., 2007) describe many interactions between drug-target proteins and small, drug-like molecules. As protein-protein interactions are measured in large scales, there are many protein interaction databases. An early one is Database of Interacting Proteins (DIP; Xenarios et al., 2000). Some later databases are more comprehensive. For example, Biological General Repository for Interaction Datasets (BioGRID; Stark et al., 2011) includes protein–protein and genetic interactions for all major model organism species; STRING (Search Tool for the Retrieval of Interacting Genes/Proteins; Jensen et al., 2009) covers known and predicted protein interactions for many species, as well as direct (physical) and indirect (functional) associations. Furthermore, some protein interaction databases are based on protein structures, such as 3D Complex (Levy et al., 2006), DOMMINO (http://dommino.org), etc.

Protein Energetics Databases

There are few databases for protein energetics, due to the low-throughput nature of the data source. One useful energetics database can be found in ProTherm (Thermodynamic Database for Proteins and Mutants; Gromiha et al., 1999). It contains thermodynamic data on mutations, including Gibbs free energy, enthalpy, heat capacity, and transition temperature. Another database is 3D-footprint (Contreras-Moreira, 2010), which provides estimates of binding specificity for protein-DNA complexes in PDB.

Bibliographic Databases

Searching for protein information through traditional bibliographic databases, such as MEDLINE or Grateful Med, can be rewarding. In addition, some bibliographic reference databases dedicated to proteins may provide certain information more directly. For example, iProLINK (integrated Protein Literature, INformation and Knowledge; Hu et al., 2004) provides literature information on proteins and their features or properties.

Combined Databases

By integrating different types of protein databases together, a database of databases (or a data warehouse) can be built. Such combined databases not only serve as a “one-stop shop,” but also provide cross-references between entries in different databases. One example of such databases is SRS (Sequence Retrieval System; Etzold et al., 1996), which is a comprehensive database for molecular biology. The home server at http://srs.ebi.ac.uk supports many biological databases, including almost all the major protein/genetic databases. As an indexing system, it provides fast access to different databases through searches by sequence or by key words from various data fields. SRS also builds indices using cross-references between databases. An entry from one database can be linked to other databases that contain the entry. However, it should be noted that the contents of SRS might lag behind the other databases in updating (i.e., some new entries in the original databases may not be included in SRS).

Go to:
SUMMARY
This unit reviews some of major protein databases on the Internet and shows what kind of information users can expect from protein databases. Although all technical procedures cannot be described here, most of the protein databases are easy to use and provide detailed on-line manuals so that even users with little computer skill can learn them quickly. Readers are encouraged to study additional protein databases that are not covered in this unit. For example, the portals listed in “INTERNET RESOURCES” give links to many other protein databases. Furthermore, the journal “Nucleic Acids Research” has a Database issue every year, which describes many high-quality, well-maintained protein databases.

Protein databases may not always be easily accessible or usable through the Internet. Sometimes a database server may be down or the Internet connection may be interrupted. For a frequent user, it may be worthwhile to install the database on a local machine. On the other hand, it must be kept in mind that a mirror site or a local copy may contain an older version of the database than the one on the home server.

It is important to assess the quality of the data. There are three types of data in protein databases. (1) Experimental data are generally very reliable. However, some entries may contain errors (e.g., some protein sequences) or may be based on low-resolution data (e.g., some protein structures determined by NMR). (2) Annotation data uses computational techniques on experimental data, for example, secondary structure assignment and domain partition in structure. These data depend on the quality of the experimental data and the computational methods used. Different methods may yield different results. (3) Prediction data includes, for example, sequence domain parsing and three-dimensional structure prediction. No matter how good the method, the results are still predictions and should be subjected to experimental verification. In addition, different methods typically give different predictions.

While protein databases on the Internet become indispensable resources for studying proteins, caution is needed when using the data from databases to draw a conclusion. The qualities of databases vary significantly. Some databases are not well maintained and contain obsolete information. It is not rare to see some protein databases disappear after a few years. In addition, the data in some databases are not carefully validated and may not be reliable. It is worthwhile to check the same type of data from different databases and compare them. It is sometimes necessary to use additional computational tools (e.g., tools to assess the quality of a structure) for further analysis.

Go to:
INTERNET RESOURCES
The Web addresses of the databases mentioned in this unit are listed in Table 19.4.1. Readers can find more protein databases and related bioinformatics tools in the following Web pages, which collect a large number of useful links:

http://bioinformatics.ca/links_directory/ (Bioinformatics Links Directory)
http://www.biophys.uni-duesseldorf.de/BioNet/Pedro/research_tools.html (Pedro’s biomolecular research tools)
http://www.expasy.org (SIB Bioinformatics Resource Portal)
http://www.123genomics.com (Genomics, Proteomics and Bioinformatics Knowledge Base)
http://bioinformatics.ws/index.php/Bioinformatics_tools_and_algorithms (Bioinformatics tools and algorithms)

PREDICTION OF THREE DIMENSIONAL STRUCTURES FROM PRIMARY STRUCTURES

Proteins are one of the major biological macromolecules performing a variety functions such as enzymatic catalysis, transport, regulation of metabolism, nerve conduction, immune response etc. The three-dimensional structure of a protein is responsible for its function. In this an overview of the need for protein structure prediction, the different approaches available as of now and their applications and limitations will be discussed.

Sequence-Structure Gap and the Need for Structure Prediction

With the advent of recombinant DNA technology it has become possible to determine the amino acid sequences of proteins quite rapidly. However, determining the three dimensional structure of proteins is a time consuming task and hence there exists a vast gap between the number of proteins of known amino acid sequence and that of known structures. This is called as the sequence-structure gap. As the knowledge of the 3-D structure of a protein is very essential to understand its function, it is imperative to develop techniques to predict the structure of a protein from its amino acid sequence.

Basis for Structure Prediction:

The classic experiments carried out by C.B. Afinson in the 60's on the enzyme ribonuclease led to the conclusion that the information to specify the 3-D structure of a protein resides in its amino acid sequence. Within the cell a newly synthesized protein chain spontaneously folds into the compact globular structure to perform its function. Thus nature has an algorithm to fold proteins to their native structures. Efforts have been directed for the past four decades to discover nature's algorithm and computational methods have been developed to predict the structure of proteins from their sequences.

Approaches to Structure Prediction

Prediction of protein structures can be classified into two major categories viz.

1. Prediction of secondary structure and

2. Prediction of tertiary (3-D) structure.

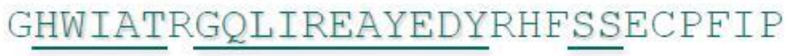
Prediction of secondary structure of proteins attempts to locate segments of the polypeptide chain adopting the α-helical or β-strand structure. Regions that are devoid of these regular secondary structural elements are considered to adopt coil conformation.

In tertiary structure prediction, one attempts to predict the three-dimensional structure of a protein or the native structure. While so far this has remained an elusive goal, different methods have been developed to press forward to the attainment of this goal.

Secondary structure prediction

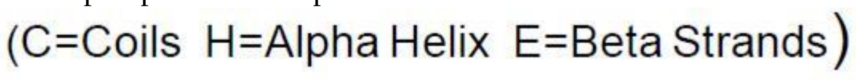
What?

1. Given a protein sequence (primary structure)

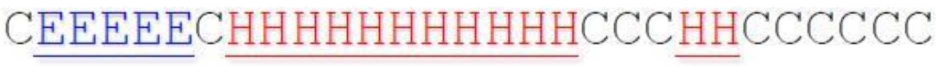


GHWIATRGQLIREAYEDYRHFSSECPFIP

2. 1 st step in prediction of protein structure.



(C=Coils H=Alpha Helix E=Beta Strands)



CEEEEECHHHHHHHHHHHCCCHHCCCCCC

8. Technique concerned with determination of secondary structure of given polypeptide by locating the Coils Alpha Helix Beta Strands in polypeptide

Why?

1. secondary structure —tertiary structure prediction
2. Protein function prediction
3. Protein classification
4. Predicting structural change
5. detection and alignment of remote homology between proteins
6. on detecting transmembrane regions, solvent-accessible residues, and other important features of molecules
7. Detection of hydrophobic region and hydrophilic region

Prediction methods

Chou-Fasman method

• Based on the propensities of different amino acids to adopt different secondary structures

9. Predictions are made using a rules-based approach to identify groups of amino acids with shared secondary structure propensities

Garnier, Osguthorpe, Robson (GOR) method

• Statistical method of secondary structure prediction based on information theory & Bayesian probability

Multiple Sequence Alignment (MSA) methods

1. Performs secondary structure prediction on a multiple sequence alignment as opposed to a single protein sequence

Neural network-based methods

3. Example: **P**rofile network from **H**ei**d**elberg (PHD)

Chou-Fasman method,

1. Alpha Helix Prediction:

A. Nucleate a helix by scanning for groups of 6 residues with at least 4 helix formers (Hα and hα) and no more than 1 helix breaker (Bα and bα).

• Two Iα residues count as one helix former for nucleating a helix

B. Propagate predicted helix in both directions until reach a four residue window with average propensity (Pα) < 1.0

C. The average propensity (Pα) for a predicted helix must be Pα > 1.03 and Pα > Pβ

2. Beta Strand Prediction:

A. Nucleate a β-strand by scanning for groups of 5 residues with at least 3 strand formers (Hβ and hβ) and no more than 1 strand breaker (B$ and b$).

B. Propagate predicted β-strand in both directions until reach a four residue window with average propensity (Pβ) < 1.0

C. The average propensity (Pβ) for a predicted β-strand must be Pβ > 1.05 and Pβ > Pα

3. Resolving conflicting predictions:

(regions with both α-helix and β-strand assignment)

• If average Pα > average Pβ, then the region is alpha helix

• If average Pβ > average Pα, then the region is beta strand

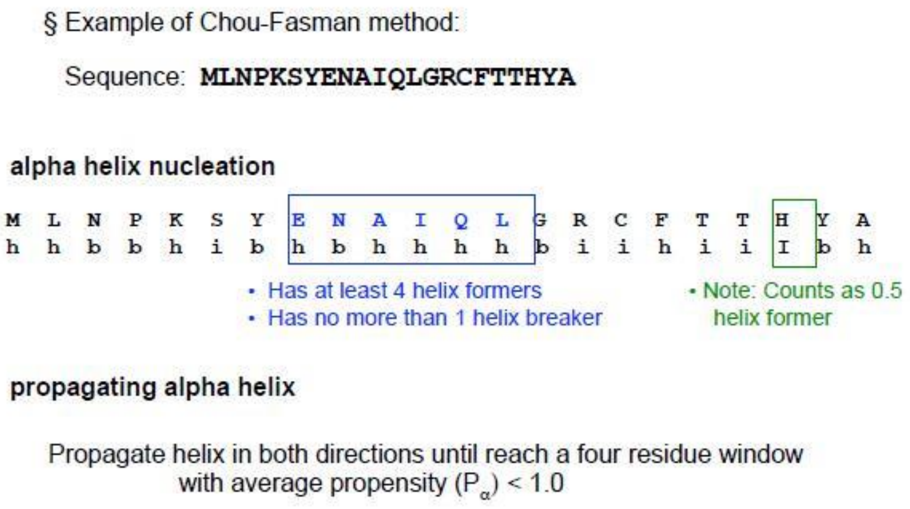
§ Notes about Chou-Fasman algorithm:

• Later versions of the algorithm included predictions for turns

• The original algorithm contained additional rules about the location of certain residues (e.g., proline) in α-helices and β-strands

• More recent versions of the algorithm have used sequential tetrapeptide average propensities to predict secondary structure

• The propensity values have also been variously recalculated with larger protein data sets (original data sets based on 15 and 29 proteins)



§ Example of Chou-Fasman method:

Sequence: **MLNPKSYENAIQLGRCFTTHYA**

alpha helix nucleation

M	L	N	P	K	S	Y	E	N	A	I	Q	L	G	R	C	F	T	T	H	Y	A
h	h	b	b	h	i	b	h	b	h	h	h	h	b	i	i	h	i	i	I	b	h

• Has at least 4 helix formers
• Has no more than 1 helix breaker

• Note: Counts as 0.5 helix former

propagating alpha helix

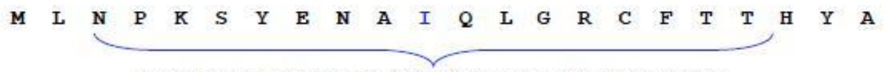
Propagate helix in both directions until reach a four residue window with average propensity (P_α) < 1.0

GOR (Garnier,Osguthorpe,Robson) Method

Key difference: Chou-Fasman uses individual amino acid propensities, while GOR incorporates information about neighboring amino acids to make prediction

A 20 x 17 matrix of directional information values for each secondary structure class was calculated from a database of known structures

These matrices are used to predict the secondary structure of the central (9th) residue in a 17 residue window:

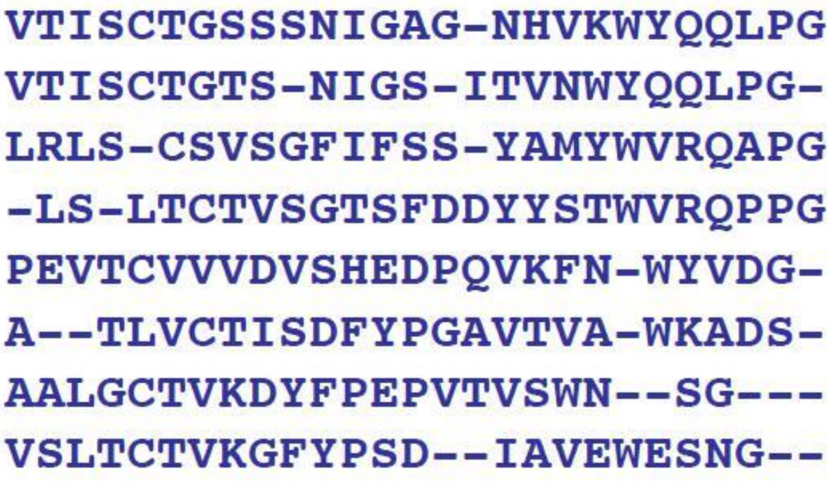


2° structure prediction for residue I is based on 17 residue window

The secondary structure class with highest information score over 17 residue window is selected as the prediction for the central residue of the window (e.g., I is predicted to be α-helix)

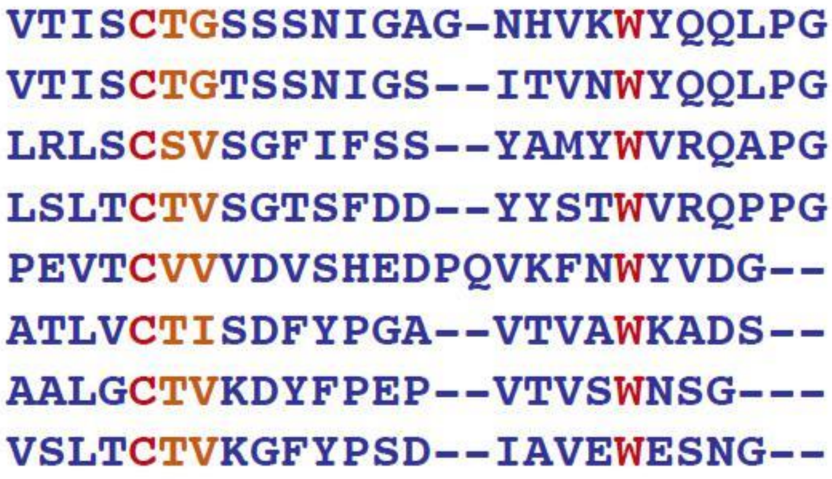
Multiple sequence alignment method

A multiple sequence alignment arranges protein sequences into a rectangular array with the goal that residues in a given column are homologous (derived from a common ancestor), superposable (in a 3D structural alignment - α helix / β sheet) or play a common functional role (catalytic sites, nuclear localisation signal, protein-protein interaction sites,...). Uses BLAST to identify homologous protein sequence fragments in a protein structure database (PDB)

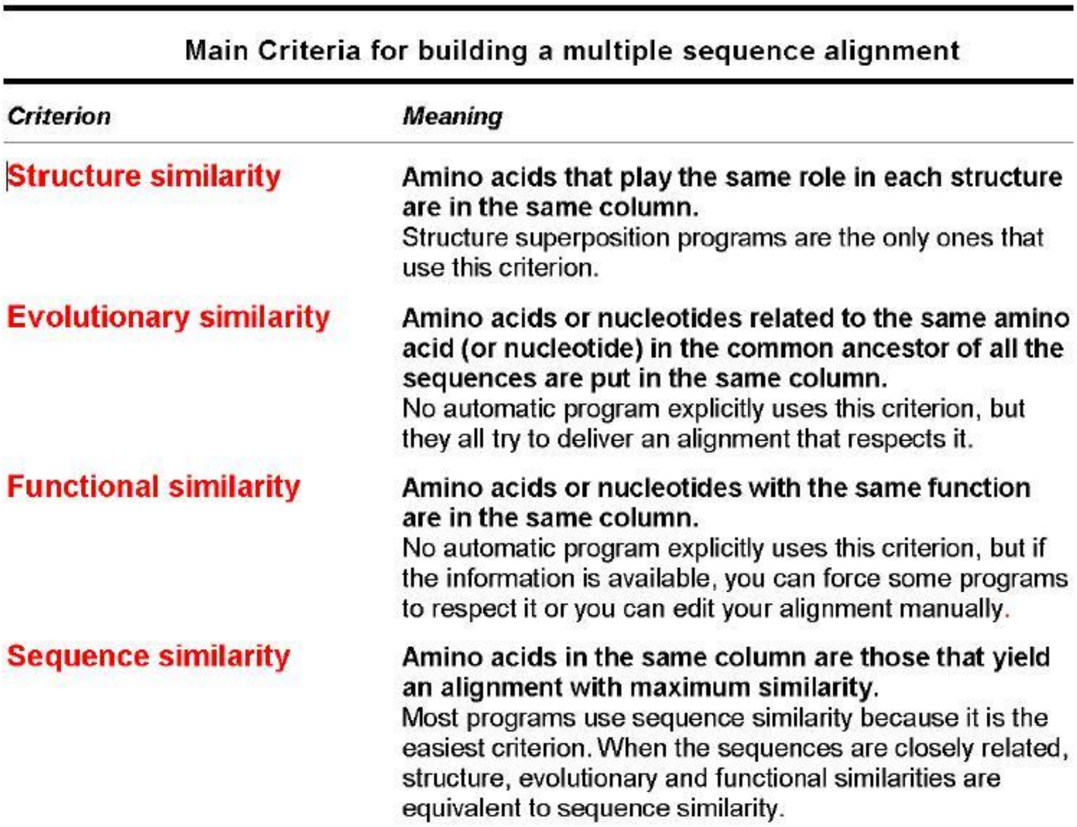


```
VTISCTGSSSNIGAG-NHVKWYQQLPG
VTISCTGTS-NIGS-ITVNWYQQLPG-
LRLS-CSVSGFIFSS-YAMYWVRQAPG
-LS-LTCTVSGTSFDDYYSTWVRQPPG
PEVTCVVVDVSHEDPQVKFN-WYVDG-
A--TLVCTISDFYPGAVTVA-WKADS-
AALGCTVKDYFPEPVTVSWN--SG---
VSLTCTVKGFYPSD--IAVEWESNG--
```

Goal: try to have a maximum of identical/similar residues in a given column of the alignment



```
VTISCTGSSSNIGAG-NHVKWYQQLPG
VTISCTGTSSNIGS--ITVNWYQQLPG
LRLSCSVSGFIFSS--YAMYWVRQAPG
LSLTCTVSGTSFDD--YYSTWVRQPPG
PEVTCVVVDVSHEDPQVKFNWYVDG--
ATLVCTISDFYPGA--VTVAWKADS--
AALGCTVKDYFPEP--VTVSWNSG---
VSLTCTVKGFYPSD--IAVEWESNG--
```

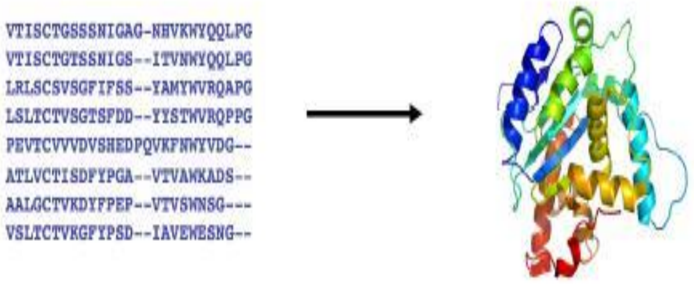


Main Criteria for building a multiple sequence alignment

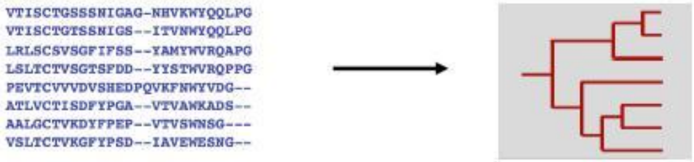
Criterion	*Meaning*
Structure similarity	**Amino acids that play the same role in each structure are in the same column.** Structure superposition programs are the only ones that use this criterion.
Evolutionary similarity	**Amino acids or nucleotides related to the same amino acid (or nucleotide) in the common ancestor of all the sequences are put in the same column.** No automatic program explicitly uses this criterion, but they all try to deliver an alignment that respects it.
Functional similarity	**Amino acids or nucleotides with the same function are in the same column.** No automatic program explicitly uses this criterion, but if the information is available, you can force some programs to respect it or you can edit your alignment manually.
Sequence similarity	**Amino acids in the same column are those that yield an alignment with maximum similarity.** Most programs use sequence similarity because it is the easiest criterion. When the sequences are closely related, structure, evolutionary and functional similarities are equivalent to sequence similarity.

What are the applications of multiple sequence alignment

§ Protein structure and function prediction


VTISCTGSSSNIGAG-NHVKWYQQLPG
VTISCTGTSSNIGS--ITVNWYQQLPG
LRLSCSVSGFIFSS--YAMYWVRQAPG
LSLTCTVSGTSFDD--YYSTWVRQPPG
PEVTCVVVDVSHEDPQVKFNWYVDG--
ATLVCTISDFYPGA--VTVAWKADS--
AALGCTVKDYFPEP--VTVSWNSG---
VSLTCTVKGFYPSD--IAVEWESNG--

§ Phylogenetic inference


VTISCTGSSSNIGAG-NHVKWYQQLPG
VTISCTGTSSNIGS--ITVNWYQQLPG
LRLSCSVSGFIFSS--YAMYWVRQAPG
LSLTCTVSGTSFDD--YYSTWVRQPPG
PEVTCVVVDVSHEDPQVKFNWYVDG--
ATLVCTISDFYPGA--VTVAWKADS--
AALGCTVKDYFPEP--VTVSWNSG---
VSLTCTVKGFYPSD--IAVEWESNG--

§ Detecting similarities between sequences (closely or distantly related) and conserved regions / motifs in sequences.

§ Detection of structural patterns (hydrophobicity/hydrophilicity, gaps etc), thus assisting improved prediction of secondary and tertiary structures and loops and variable regions.

§ Predict features of aligned sequences like conserved positions which may have structural or functional importance.

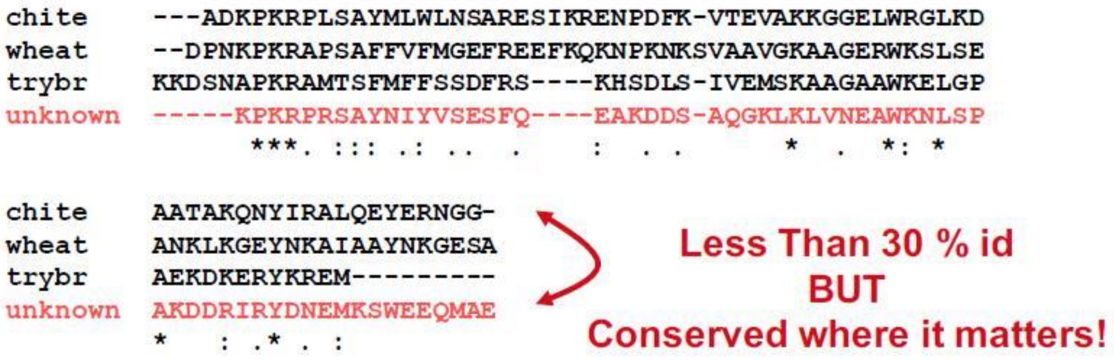
§ Computing consensus sequence.

§ Making patterns or profiles that can be further used to predict new sequences falling in a given family.

§ Deriving profiles or Hidden Markov Models that can be used to remove distant sequences (outliers) from protein families.

§ Inferring evolutionary trees / linkage.

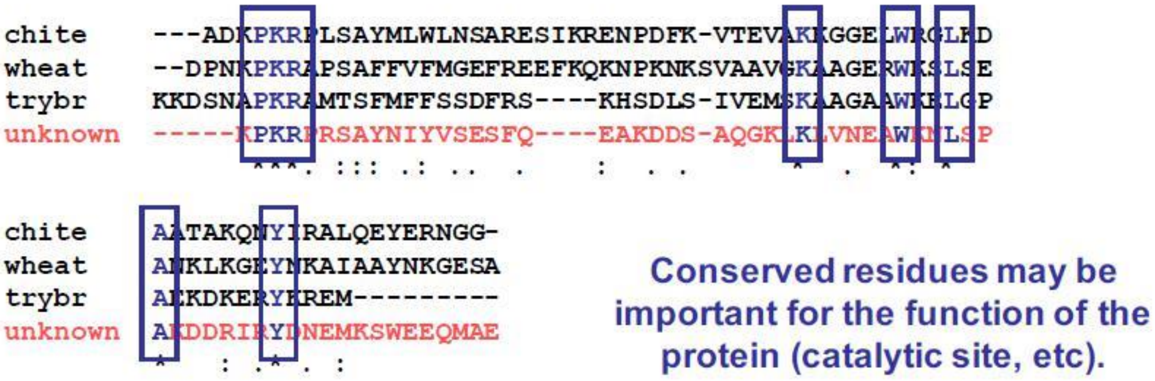
How is a multiple sequence alignment used?



```
chite     ---ADKPKRPLSAYMLWLNSARESIKRENPDFK-VTEVAKKGGELWRGLKD
wheat     --DPNKPKRAPSAFFVFMGEFREEFKQKNPKNKSVAAVGKAAGERWKSLSE
trybr     KKDSNAPKRAMTSFMFFSSDFRS----KHSDLS-IVEMSKAAGAAWKELGP
unknown   -----KPKRPRSAYNIYVSESFQ----EAKDDS-AQGKLKLVNEAWKNLSP
                ***. ::: .: ..  .    :  . .      *  .  *: *

chite     AATAKQNYIRALQEYERNGG-
wheat     ANKLKGEYNKAIAAYNKGESA
trybr     AEKDKERYKREM---------
unknown   AKDDRIRYDNEMKSWEEQMAE
          *   : .* . :
```

Less Than 30 % id
BUT
Conserved where it matters!

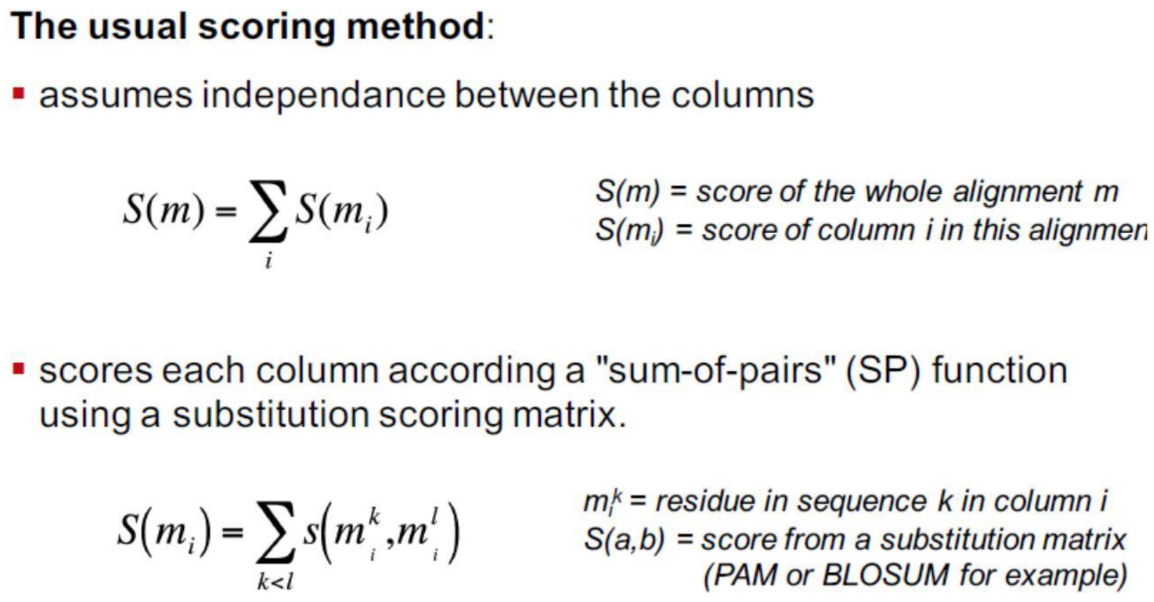


```
chite     ---ADKPKRPLSAYMLWLNSARESIKRENPDFK-VTEVAKKGGELWRGLKD
wheat     --DPNKPKRAPSAFFVFMGEFREEFKQKNPKNKSVAAVGKAAGERWKSLSE
trybr     KKDSNAPKRAMTSFMFFSSDFRS----KHSDLS-IVEMSKAAGAAWKELGP
unknown   -----KPKRPRSAYNIYVSESFQ----EAKDDS-AQGKLKLVNEAWKNLSP
                ***. ::: .: ..  .    :  . .      *  .  *: *

chite     AATAKQNYIRALQEYERNGG-
wheat     ANKLKGEYNKAIAAYNKGESA
trybr     AEKDKERYKREM---------
unknown   AKDDRIRYDNEMKSWEEQMAE
          *   : .* . :
```

Conserved residues may be important for the function of the protein (catalytic site, etc).

How to score a multiple sequence alignment?



The usual scoring method:

- assumes independance between the columns

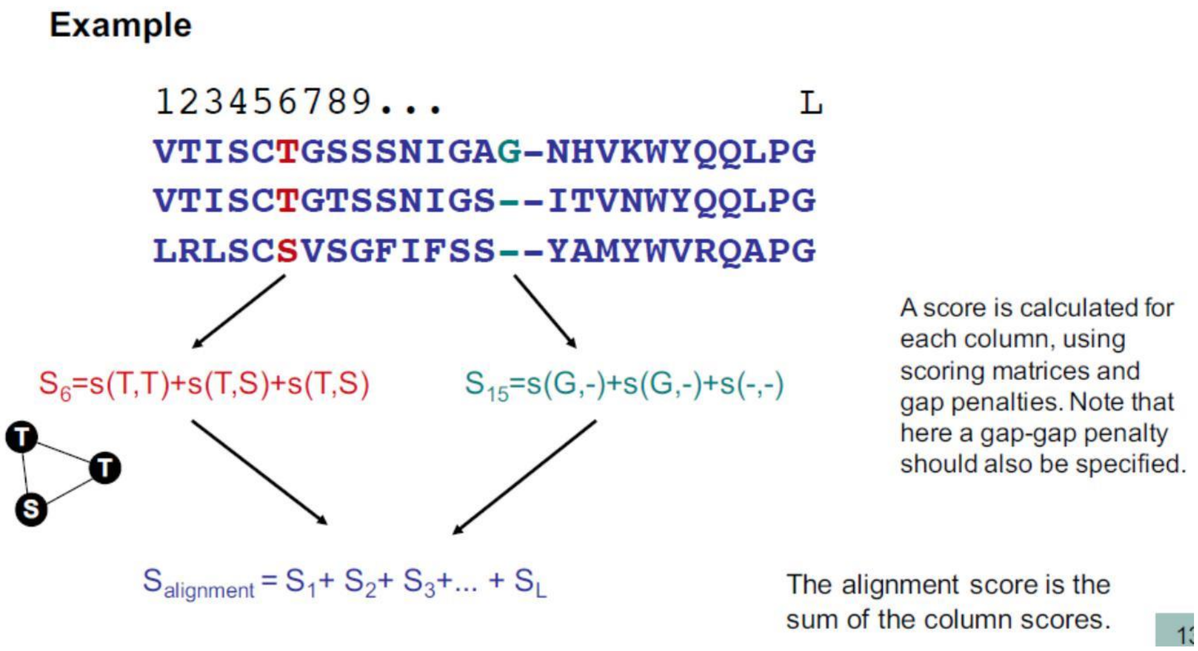
$$S(m) = \sum_i S(m_i)$$

S(m) = score of the whole alignment m
S(m$_i$) = score of column i in this alignmen

- scores each column according a "sum-of-pairs" (SP) function using a substitution scoring matrix.

$$S(m_i) = \sum_{k<l} s\left(m_i^k, m_i^l\right)$$

m$_i^k$ = residue in sequence k in column i
S(a,b) = score from a substitution matrix
(PAM or BLOSUM for example)



Example

```
123456789...                L
VTISCTGSSSNIGAG-NHVKWYQQLPG
VTISCTGTSSNIGS--ITVNWYQQLPG
LRLSCSVSGFIFSS--YAMYWVRQAPG
```

S_6=s(T,T)+s(T,S)+s(T,S)

S_{15}=s(G,-)+s(G,-)+s(-,-)

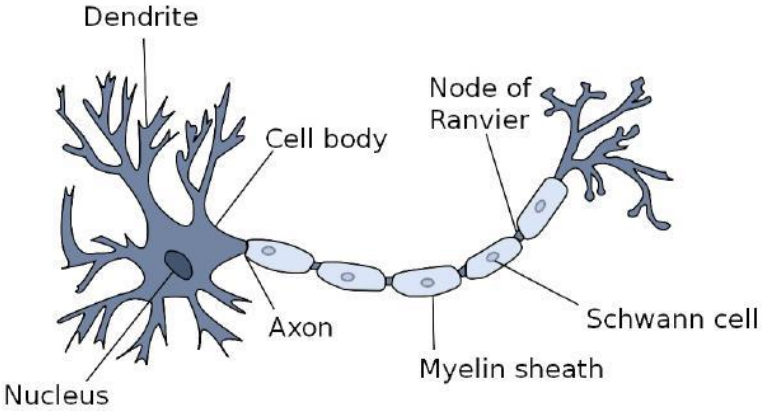
$S_{alignment} = S_1 + S_2 + S_3 + ... + S_L$

A score is calculated for each column, using scoring matrices and gap penalties. Note that here a gap-gap penalty should also be specified.

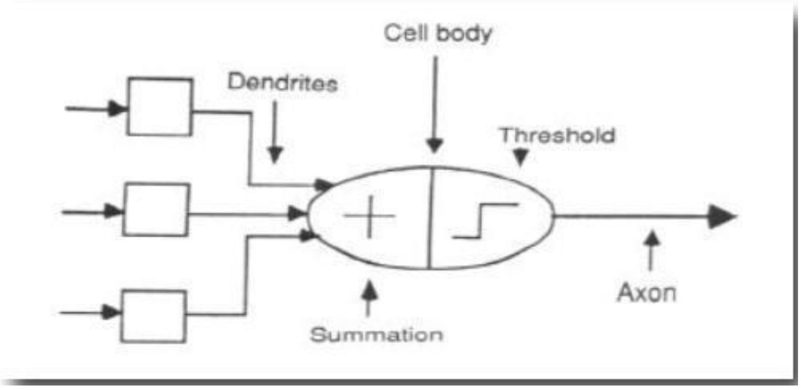
The alignment score is the sum of the column scores.

Neural network secondary structure prediction methods

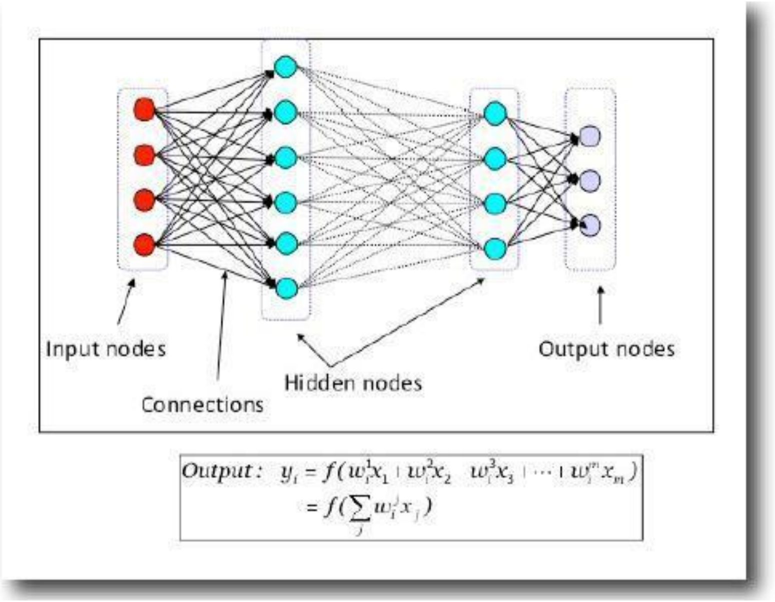
Artificial neural networks (ANN), with both statistical (linear regression and discriminant analysis) and artificial intelligence roots, are information processing units that that are modeled after the brain and its 100 billion neurons. In a neuron, the distal and proximal dendrites receive signals and communicate to the cell body, which in turn communicates with other neurons via its axon and its terminals.



Similarly, an ANN receives inputs (dendrites) that are processed with influence by weights to become outputs (axon).



The neurons or nodes interconnect with informational flows (unidirectional or bidirectional) at various weights or strengths. The simplest architecture is the perceptron, which consists of 2 layers (input and output layers) that are separated by a linear discrimination function (10). In a multi-layer perceptron (MLP) model, there are three layers: the input nodes, the hidden nodes layer, and the output nodes.

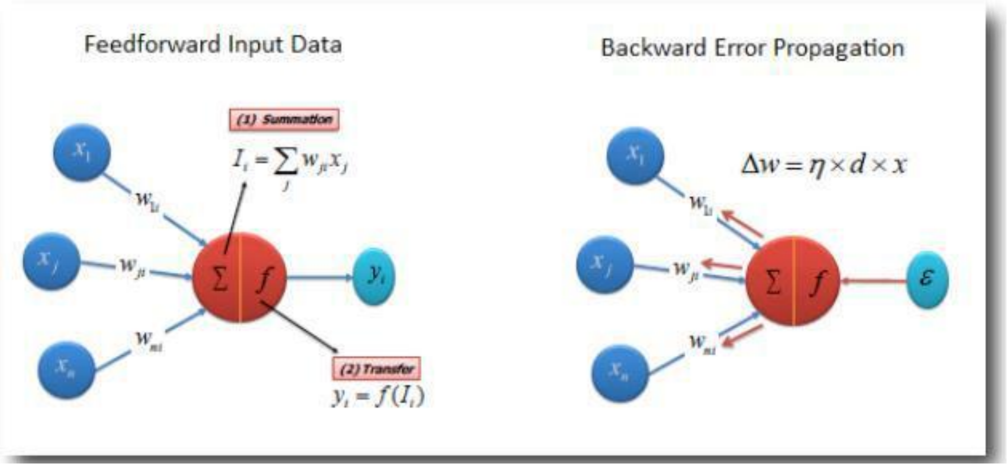

Input nodes
Connections
Hidden nodes
Output nodes
$Output: \; y_i = f(w_i^1x_1 + w_i^2x_2 \;\; w_i^3x_3 + \cdots + w_i^mx_m)$
$= f(\sum_j w_i^jx_j)$

Learning/ Training

In a feed-forward neural network architecture, a unit will receive input from several nodes or neurons belonging to another layer. These highly interconnected neurons therefore form an infrastructure (similar to the biological central nervous system) that is capable of learning by successfully perform pattern recognition and classification tasks. Training of the ANN is a process in which learning occurs from representative data and the knowledge is applied to the new situation.

This training or learning process occurs by arranging the algorithms so that the weights of the ANN are adjusted to lead to the final desired output. The learning in neural networks can be supervised (such as the multilayer perceptron that trained with sets of input data) or unsupervised (such as the Kohonen self-organizing maps which learn by finding patterns). Neural networks can also perform both regression and classification.

The ANN learning process consists of both a forward and a backward propagation process. The forward propagation process involves presenting data into the ANN whereas the important backward propagation algorithm determines the values of the weights for the nodes during a training phase. This latter process is accomplished by directing the errors for input values backwards so that corrections for the weights can be made to minimize the error of actual and desired output data. A recurrent neural network is a series of feed-forward neural networks sharing the same weights and is good for time series data. ANN can therefore extract patterns or detect trends from complicated and imprecise data sets.

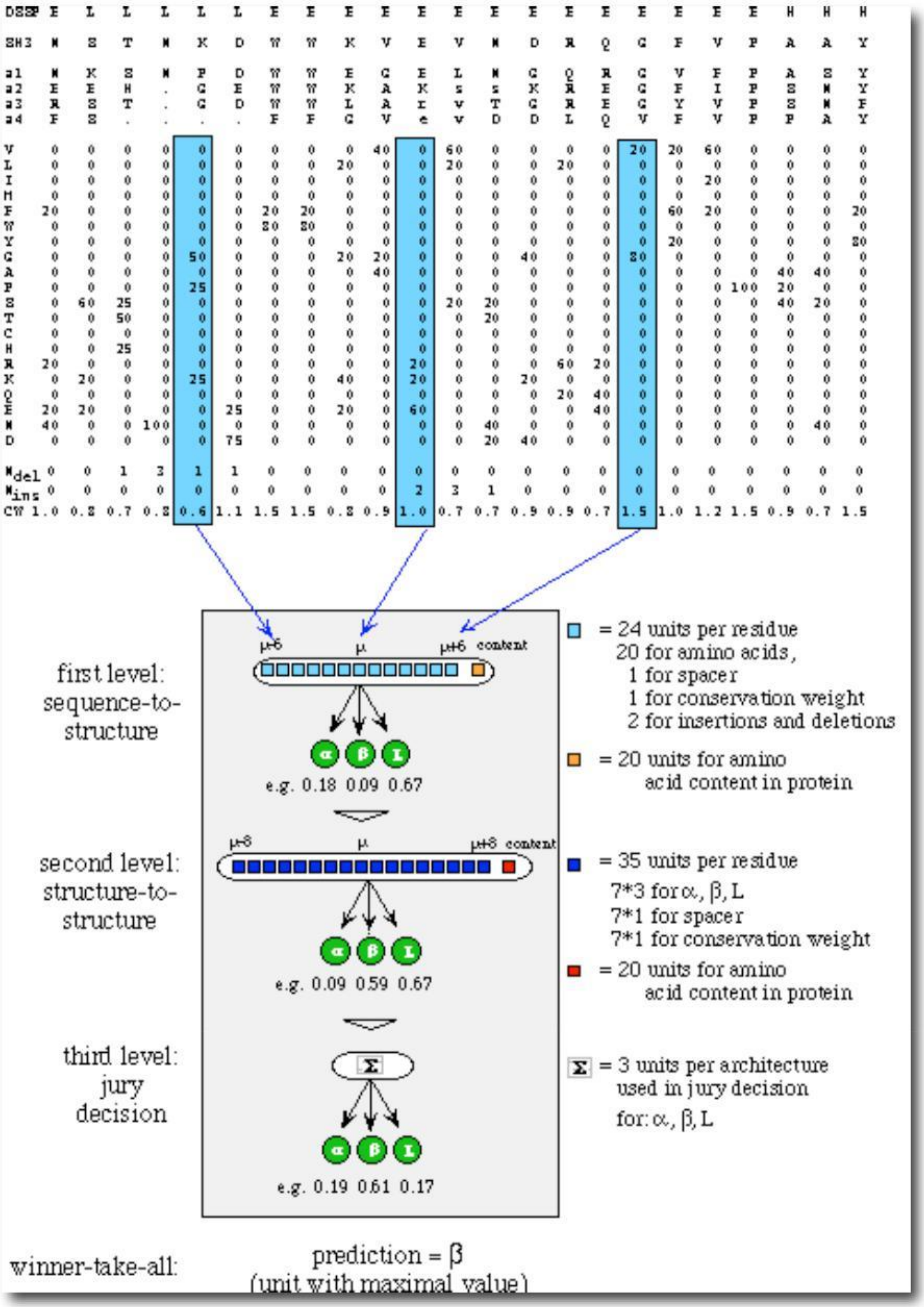


Application of ANN to bioinformatics needs the following strategy:

Extraction of features from molecular sequences to serve as training/prediction data; preprocessing that consists of feature selection and encoding into vectors of real numbers; neural network for training or prediction; postprocessing that consists of output encoding from the neural network; and finally the myriad of applications (such as sequence analysis, gene expression data analysis, or protein structure prediction).

In secondary structure prediction, neural network methods are trained using sequences with known secondary structure, and then asked to predict the secondary structure of proteins of unknown structure

§ Example: **P**rofile network from **He**i**d**elberg (PHD) uses multiple sequence alignment with neural network methods to predict secondary structure



DSSP	E	L	L	L	L	L	E	E	E	E	E	E	E	E	E	E	E	E	E	E	H	H	H
SH3	N	S	T	N	K	D	W	W	K	V	E	V	N	D	R	Q	G	F	V	P	A	A	Y
a1	N	K	S	N	P	D	W	W	E	G	E	L	N	G	Q	R	G	V	F	P	A	S	Y
a2	E	E	H	.	G	E	W	W	K	A	K	s	s	K	R	E	G	F	I	P	S	N	Y
a3	R	S	T	.	G	D	W	W	L	A	r	v	T	G	R	E	G	Y	V	P	S	N	F
a4	F	S	.	.	.	.	F	F	G	V	e	v	D	D	L	Q	V	F	V	P	P	A	Y
V	0	0	0	0	0	0	0	0	0	40	0	60	0	0	0	0	20	20	60	0	0	0	0
L	0	0	0	0	0	0	0	0	20	0	0	20	0	0	20	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	20	0	0	0	0	0	20	20	0	0	0	0	0	0	0	0	0	60	20	0	0	0	20
W	0	0	0	0	0	0	80	80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	80
G	0	0	0	0	50	0	0	0	20	20	0	0	0	40	0	0	80	0	0	0	0	0	0
A	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	40	40	0
P	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	20	0	0
S	0	60	25	0	0	0	0	0	0	0	0	20	20	0	0	0	0	0	0	0	40	20	0
T	0	0	50	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	20	0	0	0	0	0	0	0	0	0	20	0	0	0	60	20	0	0	0	0	0	0	0
K	0	20	0	0	25	0	0	0	40	0	20	0	0	20	0	0	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	40	0	0	0	0	0	0	0
E	20	20	0	0	0	25	0	0	20	0	60	0	0	0	0	40	0	0	0	0	0	0	0
N	40	0	0	100	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	40	0
D	0	0	0	0	0	75	0	0	0	0	0	0	20	40	0	0	0	0	0	0	0	0	0
N_{del}	0	0	1	3	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N_{ins}	0	0	0	0	0	0	0	0	0	0	2	3	1	0	0	0	0	0	0	0	0	0	0
CW	1.0	0.8	0.7	0.8	0.6	1.1	1.5	1.5	0.8	0.9	1.0	0.7	0.7	0.9	0.9	0.7	1.5	1.0	1.2	1.5	0.9	0.7	1.5

Network architecture (PHD). A profile-based neural network system for protein secondary structure prediction. The multiple alignment is seen at the top with a profile of amino acid occurrences compiled. Then the alignment is fed into the neural network,

which consists of 3 layers: 2 network layers and an additional layer for averaging over the independently trained networks

Protein structure prediction is one of the most important goals pursued by bioinformatics and theoretical chemistry; it is highly important in medicine (for example, in drug design) andbiotechnology (for example, in the design of novel enzymes). Every two years, the performance of current methods is assessed in the CASP experiment (Critical Assessment of Techniques for Protein Structure Prediction). A continuous evaluation of protein structure prediction web servers is performed by the community project CAMEO3D.

Accuracy of Secondary Structure Prediction

§ Prediction accuracy

- Accuracy is usually measured by Q3 (or Qindex) value
- For a single conformation state, i:

$$Q_i = \frac{\text{number of residues correctly predicted in state i}}{\text{number of residues observed in state i}} * 100\%$$

- where i is either helix, strand, or coil. For all three states:

$$Q_3 = \frac{\text{number of residues correctly predicted}}{\text{number of all residues}} * 100\%$$

§ Accuracy of prediction methods

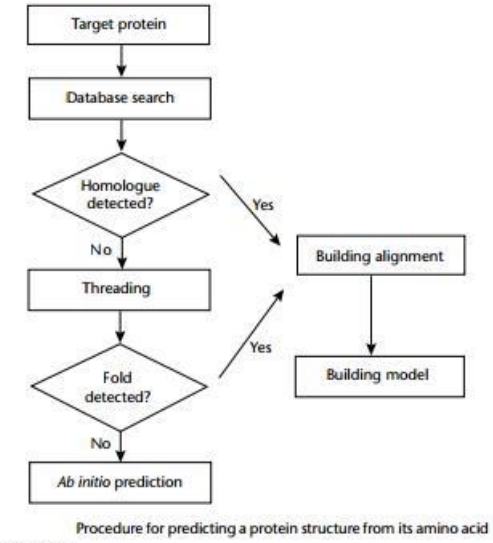
- A random prediction has a Q3 value of ~ 33-38%
- Chou-Fasman method typically has a Q3 ~ 56-60%
- GOR method (depending upon version) has a Q3 ~ 60-65%

- MSA, neural network methods have Q3 ~70%

PROTEIN TERTIARY STRUCTURES: PREDICTION FROM AMINO ACID SEQUENCES

The biological function of a protein is often intimately dependent upon its tertiary structure. X-ray crystallography and nuclear magnetic resonance are the two most mature experimental methods used to provide detailed information about protein structures. However, to date the majority of the proteins still do not have experimentally determined structures available. As at December 2000, there were about 14 000 structures available in the protein data bank (PDB, http://www.pdb.org), and there are about 10 106 000 sequence records sequences in GenBank (http://www.ncbi.nlm.nih.gov/Genbank). Thus theoretical methods are very important tools to help biologists obtain protein structure information. The goal of theoretical research is not only to predict the structures of proteins but also to understand how protein molecules fold into the native structures. The current methods for protein structure prediction can be roughly divided into three major categories: comparative modelling; threading; and ab initio prediction. For a given target protein with unknown structure, the general procedure for predicting its structure is described below

.



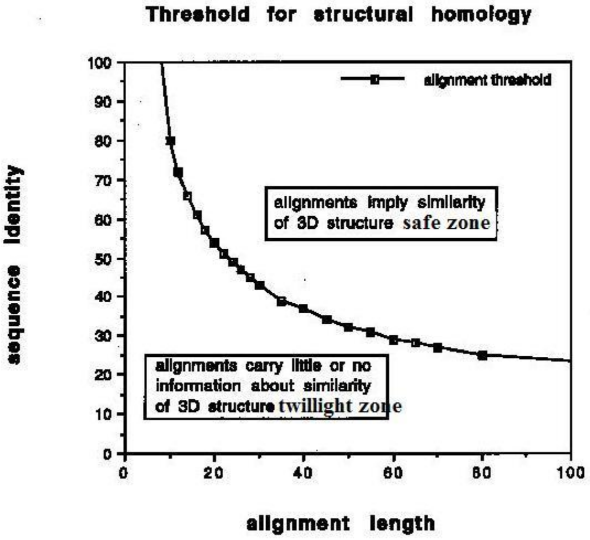
Procedure for predicting a protein structure from its amino acid sequence.

Comparative modelling

It is based on two major observations:

1. The structure of a protein is uniquely determined by its amino acid sequence. Knowing the sequence should, at least in theory, suffice to obtain the structure.

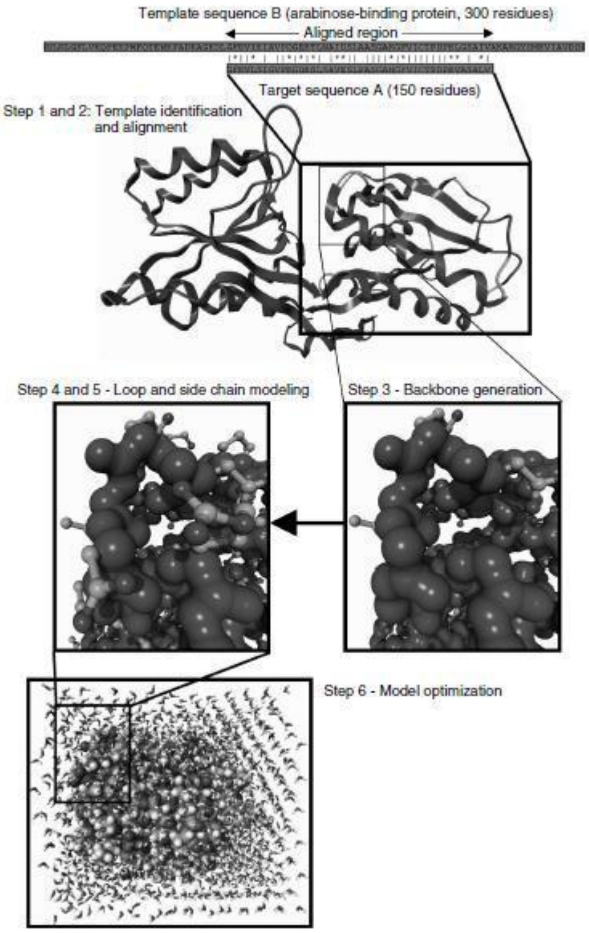
2. During evolution, the structure is more stable and changes much slower than the associated sequence, so that similar sequences adopt practically identical structures, and distantly related sequences still fold into similar structures. This relationship was first identified by Chothia and Lesk (1986) and later quantified by Sander and Schneider (1991). Thanks to the exponential growth of the Protein Data Bank (PDB), Rost (1999) could recently derive a precise limit for this rule, shown in Figure below. As long as the length of two sequences and the percentage of identical residues fall in the region marked as "safe," the two sequences are practically guaranteed to adopt a similar structure



For a sequence of 100 residues, for example, a sequence identity of 40% is sufficient for structure prediction. When the sequence identity falls in the safe homology modeling zone, we can assume that the 3D-structure of both sequences is the same.

The known structure is called the template, the unknown structure is called the target.

Homology modeling of the target structure can be done in 7 steps:

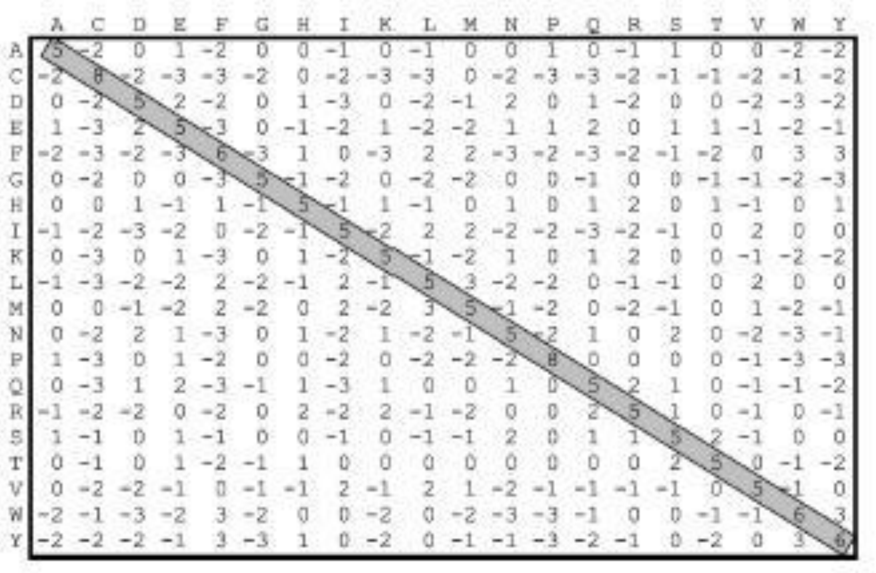

Template sequence B (arabinose-binding protein, 300 residues)
Aligned region
Target sequence A (150 residues)
Step 1 and 2: Template identification and alignment
Step 4 and 5 - Loop and side chain modeling
Step 3 - Backbone generation
Step 6 - Model optimization

1: Template recognition and initial alignment

In the safe homology modeling zone, the percentage identity between the sequence of interest and a possible template is high enough to be detected with simple sequence alignment programs such as BLAST or FASTA. To identify these hits, the program compares the query sequence to all the sequences of known structures in the PDB using mainly two matrices:

1. A residue exchange matrix (A). The elements of this 20 ∗ 20 matrix define the likelihood that any two of the 20 amino acids ought to be aligned. It is clearly seen that the values along the diagonal (representing conserved residues) are highest, but one can also observe that

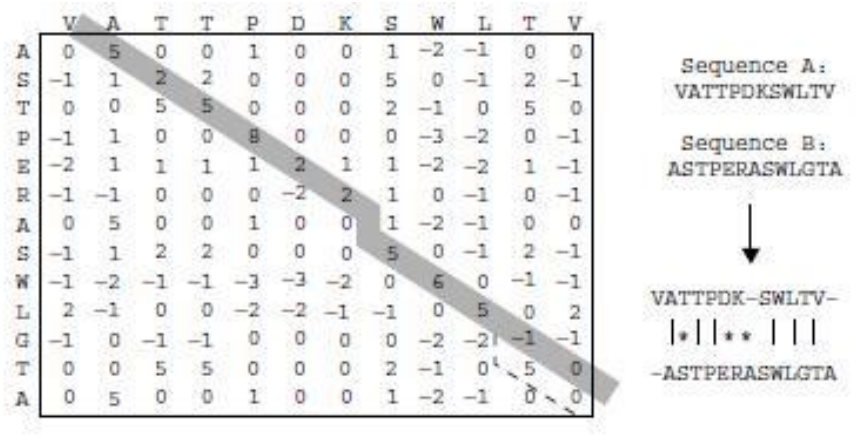
exchanges between residue types with similar physicochemical properties (for example F → Y) get a better score than exchanges between residue types that widely differ in their properties.



	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A	5	-2	0	1	-2	0	0	-1	0	-1	0	0	1	0	-1	1	0	0	-2	-2
C	-2	8	-2	-3	-3	-2	0	-2	-3	-3	0	-2	-3	-3	-2	-1	-1	-2	-1	-2
D	0	-2	5	2	-2	0	1	-3	0	-2	-1	2	0	1	-2	0	0	-2	-3	-2
E	1	-3	2	5	-3	0	-1	-2	1	-2	-2	1	1	2	0	1	1	-1	-2	-1
F	-2	-3	-2	-3	6	-3	1	0	-3	2	2	-3	-2	-3	-2	-1	-2	0	3	3
G	0	-2	0	0	-3	5	-1	-2	0	-2	-2	0	0	-1	0	0	-1	-1	-2	-3
H	0	0	1	-1	1	-1	5	-1	1	-1	0	1	0	1	2	0	1	-1	0	1
I	-1	-2	-3	-2	0	-2	-1	5	-2	2	2	-2	-2	-3	-2	-1	0	2	0	0
K	0	-3	0	1	-3	0	1	-2	5	-1	-2	1	0	1	2	0	0	-1	-2	-2
L	-1	-3	-2	-2	2	-2	-1	2	-1	5	3	-2	-2	0	-1	-1	0	2	0	0
M	0	0	-1	-2	2	-2	0	2	-2	3	5	-1	-2	0	-2	-1	0	1	-2	-1
N	0	-2	2	1	-3	0	1	-2	1	-2	-1	5	-2	1	0	2	0	-2	-3	-1
P	1	-3	0	1	-2	0	0	-2	0	-2	-2	-2	8	0	0	0	0	-1	-3	-3
Q	0	-3	1	2	-3	-1	1	-3	1	0	0	1	0	5	2	1	0	-1	-1	-2
R	-1	-2	-2	0	-2	0	2	-2	2	-1	-2	0	0	2	5	1	0	-1	0	-1
S	1	-1	0	1	-1	0	0	-1	0	-1	-1	2	0	1	1	5	2	-1	0	0
T	0	-1	0	1	-2	-1	1	0	0	0	0	0	0	0	0	2	5	0	-1	-2
V	0	-2	-2	-1	0	-1	-1	2	-1	2	1	-2	-1	-1	-1	-1	0	5	-1	0
W	-2	-1	-3	-2	3	-2	0	0	-2	0	-2	-3	-3	-1	0	0	-1	-1	6	3
Y	-2	-2	-2	-1	3	-3	1	0	-2	0	-1	-1	-3	-2	-1	0	-2	0	3	6

A* A typical residue exchange or scoring matrix used by alignment algorithms. Because the score for aligning residues A and B is normally the same as for B and A, this matrix is symmetric.

2. An alignment matrix (B). The axes of this matrix correspond to the two sequences to align, and the matrix elements are simply the values from the



	V	A	T	T	P	D	K	S	W	L	T	V
A	0	5	0	0	1	0	0	1	−2	−1	0	0
S	−1	1	2	2	0	0	0	5	0	−1	2	−1
T	0	0	5	5	0	0	0	2	−1	0	5	0
P	−1	1	0	0	8	0	0	0	−3	−2	0	−1
E	−2	1	1	1	1	2	1	1	−2	−2	1	−1
R	−1	−1	0	0	0	−2	2	1	0	−1	0	−1
A	0	5	0	0	1	0	0	1	−2	−1	0	0
S	−1	1	2	2	0	0	0	5	0	−1	2	−1
W	−1	−2	−1	−1	−3	−3	−2	0	6	0	−1	−1
L	2	−1	0	0	−2	−2	−1	−1	0	5	0	2
G	−1	0	−1	−1	0	0	0	0	−2	−2	−1	−1
T	0	0	5	5	0	0	0	2	−1	0	5	0
A	0	5	0	0	1	0	0	1	−2	−1	0	0

Sequence A:
VATTPDKSWLTV

Sequence B:
ASTPERASWLGTA

```
VATTPDK-SWLTV-
|*||** |||
-ASTPERASWLGTA
```

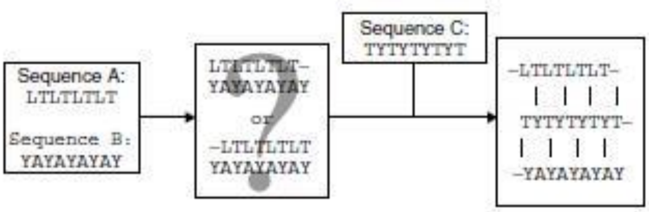
B: The alignment matrix for the sequences VATTPDKSWLTV and ASTPERASWLGTA, using the scores from Figure A. The optimum path corresponding to the alignment on the right side is shown in gray. Residues with similar properties are marked with a star (*). The dashed line marks an alternative alignment that scores more points but requires opening a second gap

residue exchange matrix (Fig. A) for a given pair of residues. During the alignment process, one tries to find the best path through this matrix, starting from a point near the top left, and going down to the bottom right. To make sure that no residue is used twice, one must always take at least one step to the right and one step down. A typical alignment path is shown in Figure B. At first sight, the dashed path in the bottom right corner would have led to a higher score. However, it requires the opening of an additional gap in sequence A (Gly of sequence B is skipped). By comparing thousands of sequences and sequence families, it became clear that the opening of gaps is about as unlikely as at least a couple of nonidentical residues in a row. The jump roughly in the middle of the matrix, however, is justified, because after the jump we earn lots of points (5,6,5), which would have been (1,0,0) without the jump. The alignment algorithm therefore subtracts an "opening penalty" for every new gap and a much smaller "gap extension penalty" for every residue that is skipped in the alignment. The gap extension penalty is smaller simply because one gap of three residues is much more likely than three gaps of one residue each. In practice, one just feeds the query sequence to one of the countless BLAST servers on the web, selects a search of the PDB, and obtains a list of hits—the modeling templates and corresponding alignments.

2: Alignment correction

Having identified one or more possible modeling templates using the fast methods described above, it is time to consider more sophisticated methods to arrive at a better alignment. Sometimes it may be difficult to align two sequences in a region where the percentage sequence identity is very low.

One can then use other sequences from homologous proteins to find a solution. A pathological example is shown in C:



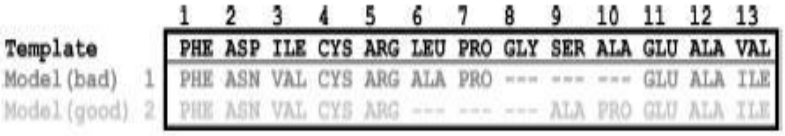
C: A pathological alignment problem. Sequences A and B are impossible to align, unless one considers a third sequence C from a homologous protein.

Suppose you want to align the sequence LTLTLTLT with YAYAYAYAY. There are two equally poor possibilities, and only a third sequence, TYTYTYTYT, that aligns easily to both of them can solve the issue.

The example above introduced a very powerful concept called “multiple sequence alignment.” Many programs are available to align a number of related sequences, for example CLUSTALW, and the resulting alignment contains a lot of additional information.

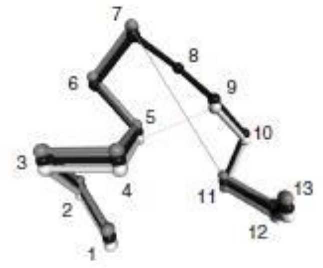
Think about an Ala → Glu mutation. Relying on the matrix in Figure A, this exchange always gets a score of 1. In the 3D structure of the protein, it is however very unlikely to see such an Ala → Glu exchange in the hydrophobic core, but on the surface this mutation is perfectly normal. The multiple sequence alignment implicitly contains information about this structural context. If at a certain position only exchanges between hydrophobic residues are observed, it is highly likely that this residue is buried. To consider this knowledge during the alignment, one uses the multiple sequence alignment to derive positionspecific scoring matrices, also called profiles. When building a homology model, we are in the fortunate situation of having an almost perfect profile—the known structure of the template. We simply know that a certain alanine sits in the protein core and must therefore not be aligned with a glutamate. Multiple sequence alignments are nevertheless useful in homology modeling, for example, to place deletions (missing residues in the model) or insertions (additional residues in the model) only in areas where the sequences are strongly divergent.

A typical example for correcting an alignment with the help of the template is shown in Figures D and E. Although a simple sequence alignment gives the highest score for the wrong answer (alignment 1 in Fig. D), a simple look at the structure of the template reveals that alignment 2 is correct, because it leads to a small gap, compared to a huge hole associated with alignment 1.



		1	2	3	4	5	6	7	8	9	10	11	12	13
Template		PHE	ASP	ILE	CYS	ARG	LEU	PRO	GLY	SER	ALA	GLU	ALA	VAL
Model(bad)	1	PHE	ASN	VAL	CYS	ARG	ALA	PRO	---	---	---	GLU	ALA	ILE
Model(good)	2	PHE	ASN	VAL	CYS	ARG	---	---	---	ALA	PRO	GLU	ALA	ILE

D: Example of a sequence alignment where a three-residue deletion must be modeled. While the first alignment appears better when considering just the sequences (a matching proline at position 7), a look at the structure of the template leads to a different conclusion (Figure E)



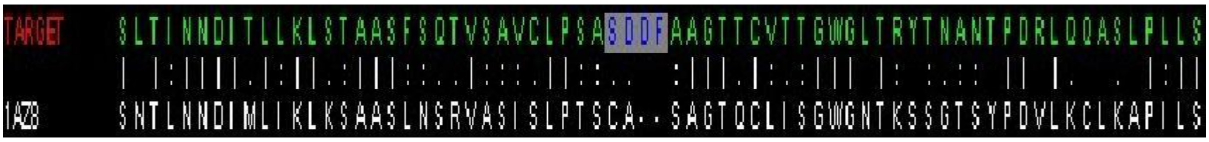
E: Correcting an alignment based on the structure of the modeling template (Cα-trace shown in black). While the alignment with the highest score (dark gray, also in Figure D) leads to a gap of 7.5 A between residues 7 and 11, the second option (white) creates only a tiny hole of ˚ 1.3 A between residues 5 and 9. This can easily be accommodated by small backbone shifts. (The ˚ normal Cα−Cα distance of 3.8 A has been subtracted).

3: Backbone generation

When the alignment is correct, the backbone of the target can be created. The coordinates of the template-backbone are copied to the target. When the residues are identical, the side-chain coordinates are also copied. Because a PDB-file can always contain some errors, it can be useful to make use of multiple templates.

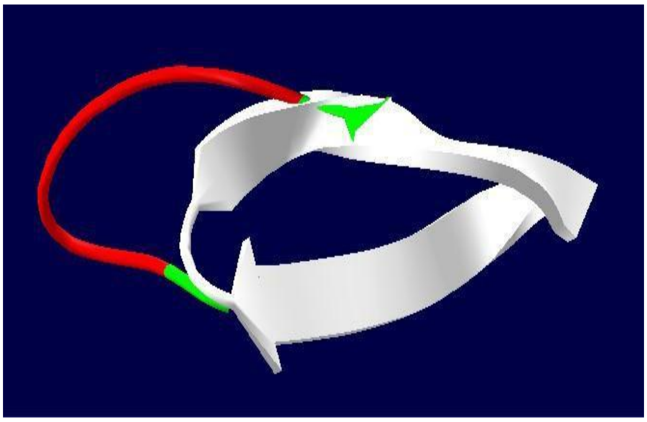
4: Loop modeling

Often the alignment will contain gaps as a result of deletions and insertions. When the target sequence contains a gap, one can simply delete the corresponding residues in the template. This creates a hole in the model, this has already been discussed in step 2. When there is an insertion in the target, shown in Figure B, the template will contain a gap and there are no backbone coordinates known for these residues in the model. The backbone from the template has to be cut to insert these residues. Such large changes cannot be modeled in secondary structure elements and therefore have to be placed in loops and strands. Surface loops are, however, flexible and difficult to predict. One way to handle loops is to take some residues before and after the insertion as "anchor" residues and search the PDB for loops with the same anchor-residues. The best loop is simply copied in the model. This is shown in Figure G. The two residues which are colored green in Figure G are used as anchor, the best loop with the inserted resisdues was found in the database and placed in the model.



```
TARGET  SLTINNDITLLKLSTAASFSQTVSAVCLPSASDDFAAGTTCVTTGWGLTRYTNANTPDRLQQASLPLLS
        | |:||||.|:||.:|||::..|:::.||::..  :|||.|:.:||| |: :.:: || |.  . |:||
1AZB    SNTLNNDIMLIKLKSAASLNSRVASISLPTSCA--SAGTQCLISGWGNTKSSGTSYPDVLKCLKAPILS
```

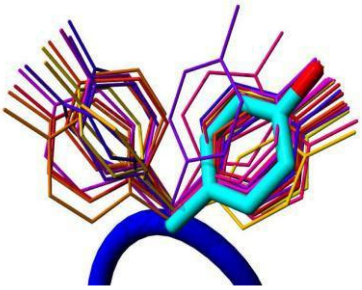
F: Target sequence (green) with insertion (grey box) results in a gap in the template.



F: The red loop is modeled with the green residues as anchor residues. The insertion of 2 residues results in a longer loop.

5: Side-chain modelling

Now it is time to add side-chains to the backbone of the model. Conserved residues were already copied completely. The torsion angle between C-alpha and C-beta of the other residues can also be copied to the model because these rotamers tend to be conserved in similar proteins. It is also possible to predict the rotamer because many backbone configurations strongly prefer a specific rotamer. As shown in Figure G, the backbone of this tyrosine strongly prefers two rotamers and the real side-chain fits in one of them. There are libraries based upon the backbone of the residues flanking the residue of interest. By using these libraries the best rotamer can be predicted. This last method is used by Yasara.



G*: Prefered rotamers of this tyrosin (colored sticks) the real side-chain (cyan) fits in one of them.*

6: Model optimization

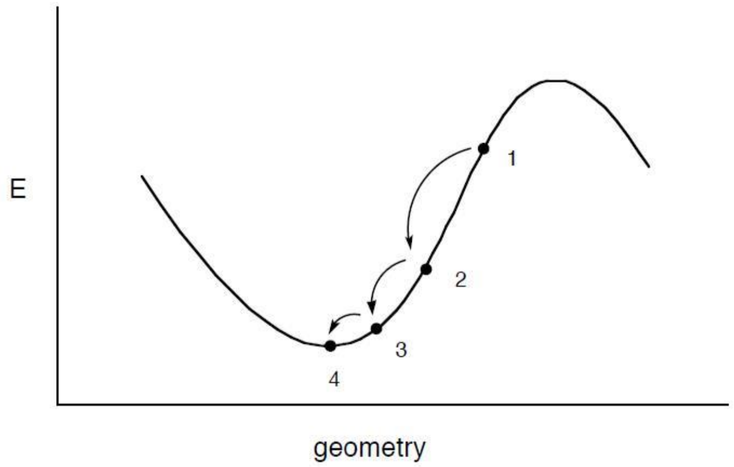
The model has to be optimized because Many structural artifacts can be introduced while the model protein is being built

- ❑ Substitution of large side chains for small ones
- ❑ Strained peptide bonds between segments taken from difference reference proteins
- ❑ Non optimum conformation of loops

Energy Minimisation is used to produce a chemically and conformationally reasonable model protein structure

Two mainly used optimisation algorithms are

- ➢ Steepest Descent
- ➢ Conjugate Gradients



The process of energy minimization changes the geometry of the molecule in a step-wise fashion until a minimum is reached.

Molecular Dynamics is used to explore the conformational space a molecule could visit, Molecular dynamics (MD) is a computer simulation method for studying the physical movements of atoms and molecules

7: Model validation

The models we obtain may contain errors. These errors mainly depend upon two values.

1. The percentage identity between the template and the target.

If the value is > 90% then accuracy can be compared to crystallography, except for a few individual side chains. If its value ranges between 50-90 % r.m.s.d. error can be as large as 1.5 Å, with considerably more errors. If the value is <25% the alignment turns out to be difficult for homology modeling, often leading to quite larger errors.

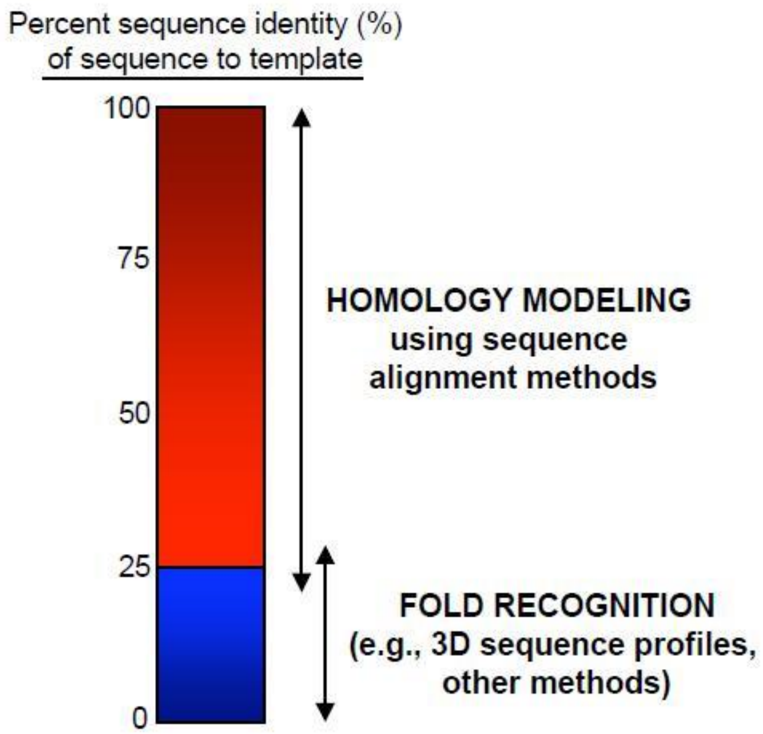
2. The number of errors in the template.

Errors in a model become less of a problem if they can be localized. Therefore, an essential step in the homology modeling process is the verification of the model. The errors can be estimated by calculating the model's energy based on a force field. This method checks to see if the bond lengths and angles are in a normal range. However, this method cannot judge if the model is correctly folded. The 3D distribution functions can also easily identify misfolded proteins and are good indicators of local model building problems.

Modeller

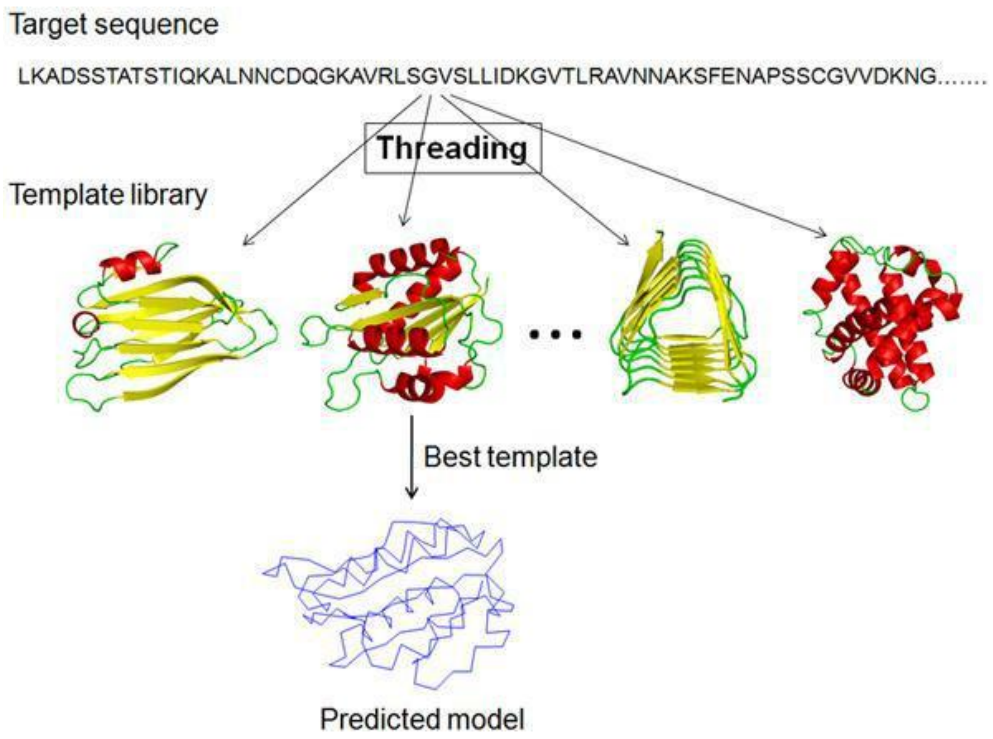
Modeller is a program for comparative protein structure modelling by satisfaction of spatial restraints. It can be described as "Modeling by satisfaction of restraints" uses a set of restraints derived from an alignment and the model is obtained by minimization of these restraints. These restraints can be from related protein structures or NMR experiments. User gives an alignment of sequences to be modelled with known structures. Modeller calculates a model with all non hydrogen atoms. It also performs comparison of protein structures or sequences, clustering of proteins, searching of sequence databases.

THREADING


Percent sequence identity (%) of sequence to template
100
75
50
25
0
HOMOLOGY MODELING using sequence alignment methods
FOLD RECOGNITION (e.g., 3D sequence profiles, other methods)

Threading or Fold recognition is a method to identify proteins that have similar 3D structure (fold), but limited or non existent sequence homology. The threading and sequence-structure alignment approachs are based on the observation that many protein structures in the PDB are very similar. For example, there are many 4-helical bundles, TIM barrels, globins, etc. in the set of solved structures.

As a result of this, many scientists have conjectured there are only a limited number of " unique" protein folds in nature. Estimates vary considerably, but some predict that are fewer than 1000 different protein folds. Thus, one approach to the protein structure prediction problem is to try to determine the structure of a new sequence by finding its best fit" to some fold in a library of structures.



Given a new sequence and a library of known folds, the goal is to _figure out which of the folds (if any) is a good fit to the sequence.

Fold recognition methods include:

• 3D profiles (and protein threading)

- Align sequence to structure

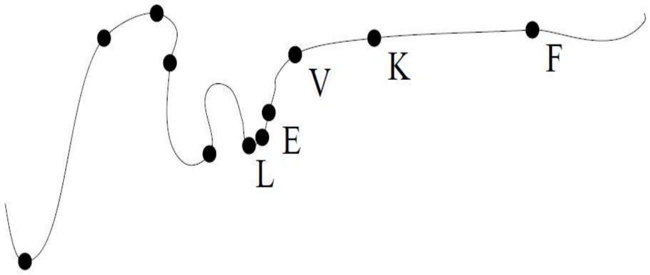
• Profile-based alignment methods that integrate sequence and structural (2D or 3D) information

- e.g., 3D-PSSM or PHYRE software

As a subproblem to fold recognition, we must solve the sequence-structure alignment problem.

Namely, given a solved structure T for a sequence t_1 t_2 ….. t_n = t and a new sequences s_1 s_2 …. s_m = s, we need to find the best match" between s and T. This actually consists of two subproblems:

- Evaluating (scoring) a given alignment of s with a structure T.
- Efficiently searching over possible alignments.



Example: New sequence s=LEVKF, and its best alignment to a particular structure.

There are at least three approaches to the sequence-structure alignment problem.

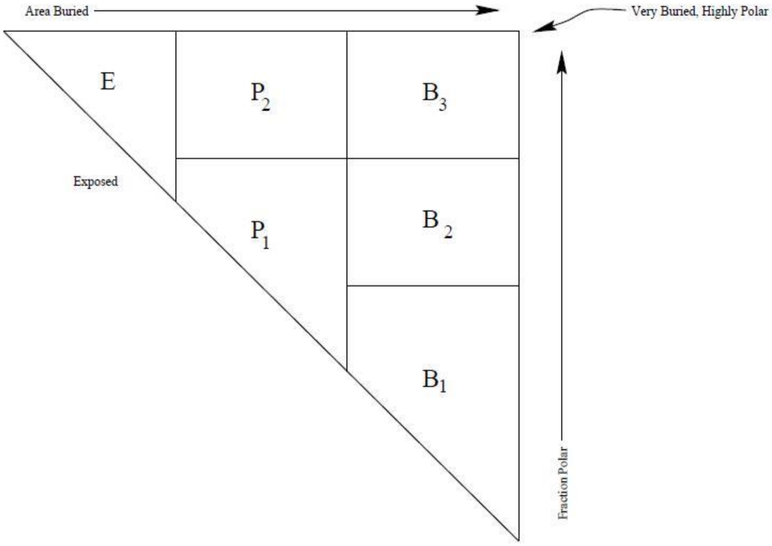
1. The first method is to just use protein sequence alignment. That is, find the best sequence alignment between the new sequence s and the sequence t with structure T. This is then used to infer the structural alignment: if s_i aligns with t_j , s_i's position in the 3D structure is the same as t_j 's. Scoring in this case is based on amino-acid similarity matrices (e.g., you could use the PAM-250 matrix), and the search algorithm is dynamic programming (O(nm) time). This is a non- physical method; that is, it does not use structural information. The major limitation of this method is that similar structures have lots of sequence variability, and thus sequence alignment may not be very helpful. Hidden Markov model techniques have the same problem.

2. The second method we will describe, the 3D profile method, actually uses structural information. The idea here is that instead of aligning a sequence to a sequence, we align a sequence to a string of descriptors that describe the 3D environment of the target structure. That is, for each residue position in the structure, we determine:

_ how buried it is (buried, partly buried or exposed)

_ the fraction of surrounding environment that is polar (polar or apolar)

_ the local secondary structure (α-helix, β-sheet or other)



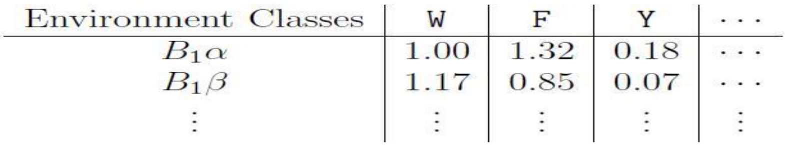
We assign 6 classes of environments to each position in the structure. These environments (E, P1, P2, B1, B2 and B3) depend on the number of surrounding polar residues and how buried the position is. Since there are 3 possible secondary structures for each of these, we have a total of 6 x3 = 18 environment classes.

For each position in the structure, we categorize it into one of 18 environment classes using these characteristics. Because we are using environmental variables, this adds a physical dimension to the problem. The key observation is that different amino acids prefer different environments.

For all proteins in the PDB, we can tabulate the number of times we see a particular residue in a particular environment class, and use this to compute a score for each environment class and each amino acid pair. In particular, we compute a log-odds score of

$$\text{score}_{ij} = \ln\left(\frac{Pr(\text{residue } j \text{ in enviroment } i)}{Pr(\text{residue } j \text{ in any enviroment})}\right)$$

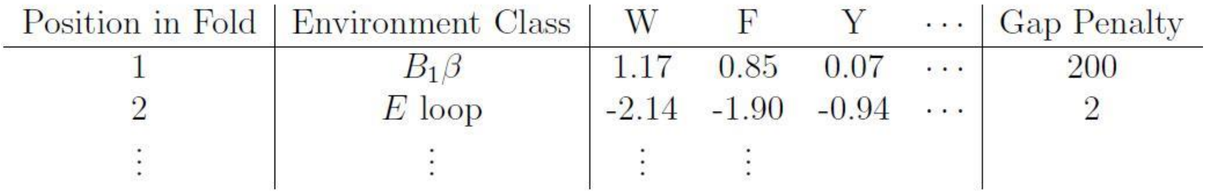
The denominator is obtained from amino acid frequencies present in the PDB This gives us an 18x20 table as follows:



Environment Classes	W	F	Y	⋯
$B_1\alpha$	1.00	1.32	0.18	⋯
$B_1\beta$	1.17	0.85	0.07	⋯
⋮	⋮	⋮	⋮	⋮

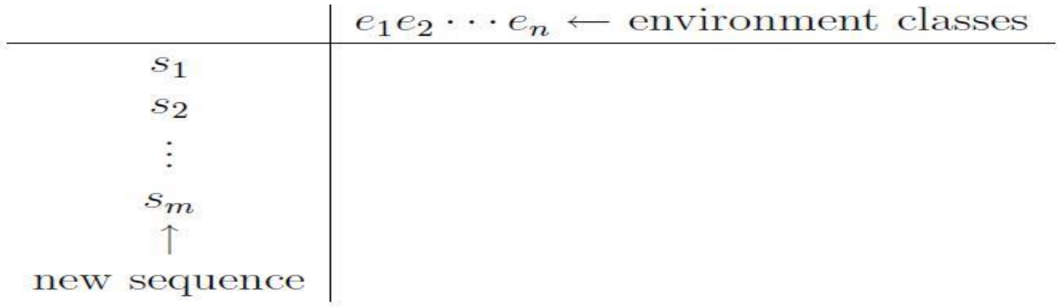
Then we can build a 3D profile for a particular structure using this table. Namely, for each position in our structure, we determine its environment class, and the score of a particular amino acid in this position depends on the table we built above.

Thus, for example, if the first position in our structure has environment class B1β, the score of having a tyrosine (Y) in that position is 0.07. Thus, for example, if there are n positions in our structure, we build a table as follows:



Position in Fold	Environment Class	W	F	Y	⋯	Gap Penalty
1	$B_1\beta$	1.17	0.85	0.07	⋯	200
2	E loop	-2.14	-1.90	-0.94	⋯	2
⋮	⋮	⋮	⋮			

Then to align a sequence s with a structure, we align the sequence with the descriptors of the 3D environment of the target structure. To find the best alignment, we use a 2D dynamic programming matrix as for regular sequence alignment:



$$\begin{array}{c|l} & e_1 e_2 \cdots e_n \leftarrow \text{environment classes} \\ \hline s_1 & \\ s_2 & \\ \vdots & \\ s_m & \\ \uparrow & \\ \text{new sequence} & \end{array}$$

Thus, to use the 3D profile method for fold recognition, for a particular sequence we calculate its score (using dynamic programming) for all structures. Signifcance of a score for a particular structure is given by scoring a large sequence database against the structure and calculating

$$z_{-\text{score}} = \frac{\text{score} \ -\mu}{s}$$

Where µ is the mean score for that structure, and s is the standard deviation of the scores.

The advantages of the 3D profile method over regular sequence alignment is that environmental tendencies may be more informative than simple amino acid similarity, and that structural information is actually used. Additionally, this is a fast method with reasonably good performance. The major disadvantage of this method is that it assumes independence between all positions in the structure.

3. Our third method for sequence-structure alignments uses contact potentials. Most "threading" methods today fall into this category.

Typically, these methods model interactions in a protein structure as a sum over pairwise interactions.

One formalization of the problem is:

Given: a structure P with positions p_1; p_2;……; p_n, and a sequence s_1;……..; sm.

Find: t_1; t_2……..; tn (where 1< t1 < t2 < _ _ _ < tn ≤ m and t_i indicates the index of the amino acid from s that occupies p_i) such that

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \text{score}\left(i, j, s_{t_i}, s_{t_j}\right)$$

is maximized.

This problem is NP-complete for pairwise interactions. (If the contact graph for a structure is planar, there are approximation algorithms for this problem. However, in practice, they are not used because most graphs would not be planar and heuristics are thought to give better solutions.) One approach commonly used to find threadings is to disallow gaps into core segments (such helices and sheets), and to put lower and upper bounds on distances between core segments. Some algorithms also use exhaustive enumeration and branch and bound techniques to find the best threading. Alternatively, some approaches give up the guarantee of finding the best threading, and use fast heuristics instead.

The score functions come from database-derived pairwise potentials. The general idea is to define a cutoff parameter for contact" (e.g., up to 6 Angstroms), and to use the PDB to count up the number of times amino acids i and j are in contact:

$$\text{score}_{ij} = \ln\left(\frac{Pr(i,j|\text{ contact })}{\text{normalization}}\right).$$

There are several methods to do this normalization. For example, in [2], normalization is by expected frequencies.

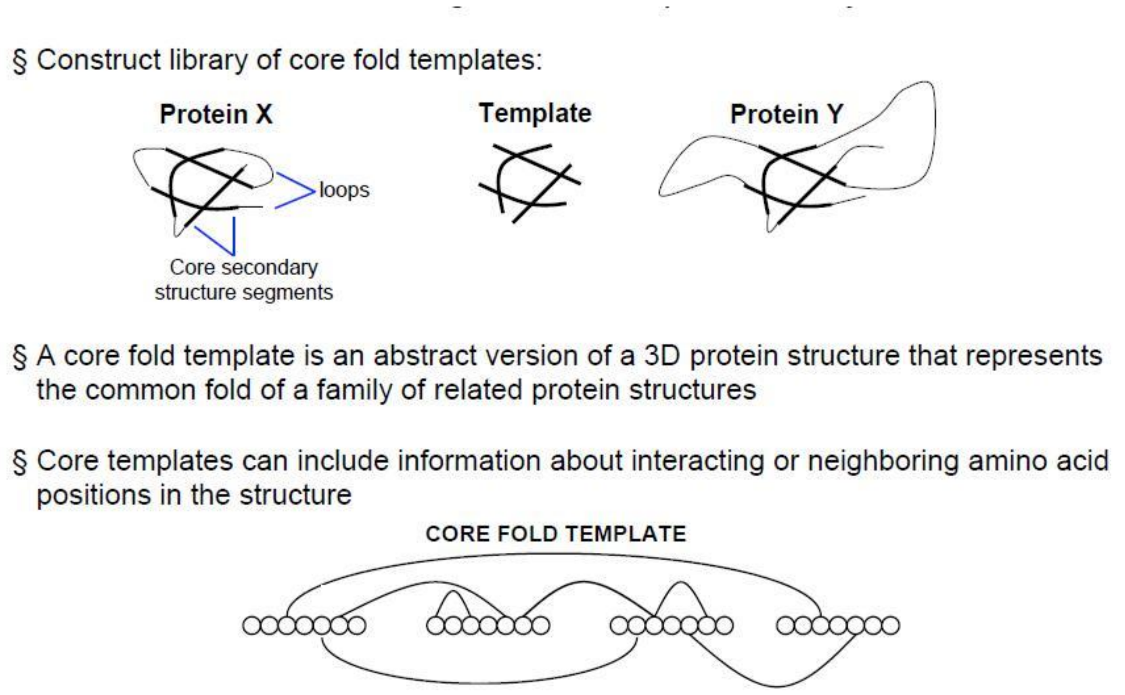
Additionally, there are many variations in defining the potentials. For example, in addition to pairwise potentials, some researchers consider single residue potentials as well (e.g., to take into account hydrophobicity or secondary structure), or distance-dependent intervals (e.g., counting up pairwise contacts separately for intervals within 1 Angstrom, between 1 and 2 Angstroms, etc.).

A general paradigm of protein threading consists of the following four steps:

1. Construct a library of core fold templates
2. A scoring (or objective) function is used to evaluate the placement of a sequence in a core template
3. Search for optimal alignments between the sequence and each core fold template
4. Select the core fold template that best aligns (fits) with the protein sequence
    - The 3D model is derived from the optimal alignment (or ‘threading’) of the sequence to the best scoring structural template

The construction of a structure template database

Select protein structures from the protein structure databases as structural templates. This generally involves selecting protein structures from databases such as PDB, FSSP, SCOP, or CATH, after removing protein structures with high sequence similarities.

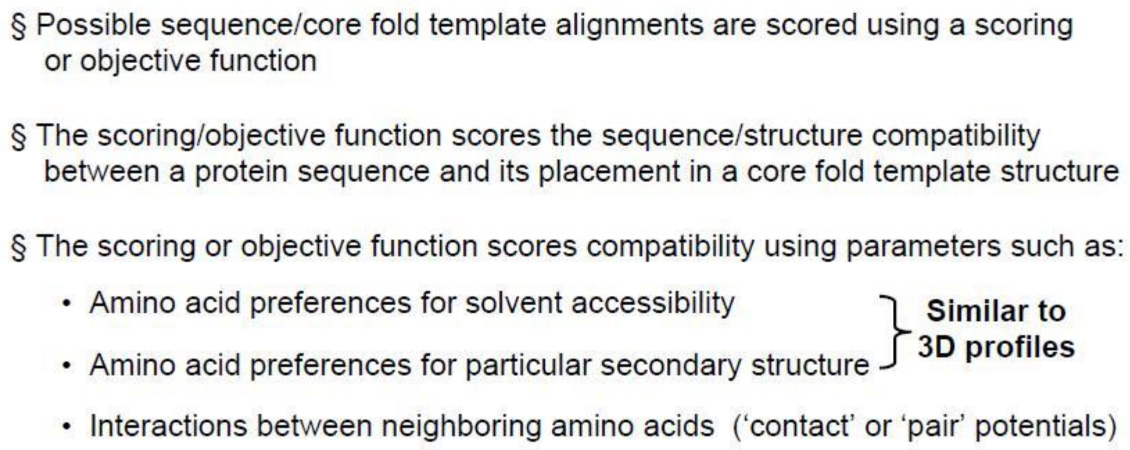


§ Construct library of core fold templates:

§ A core fold template is an abstract version of a 3D protein structure that represents the common fold of a family of related protein structures

§ Core templates can include information about interacting or neighboring amino acid positions in the structure

The design of the scoring function



§ Possible sequence/core fold template alignments are scored using a scoring or objective function

§ The scoring/objective function scores the sequence/structure compatibility between a protein sequence and its placement in a core fold template structure

§ The scoring or objective function scores compatibility using parameters such as:

- Amino acid preferences for solvent accessibility
- Amino acid preferences for particular secondary structure
- Interactions between neighboring amino acids ('contact' or 'pair' potentials)

Similar to 3D profiles

Threading alignment

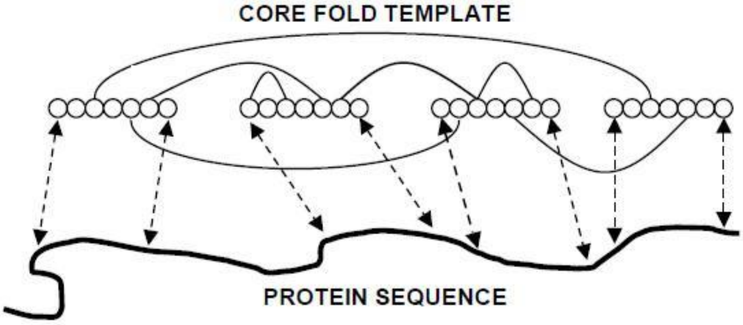
Align the target sequence with each of the structure templates by optimizing the designed scoring function. This step is one of the major tasks of all threading-based structure prediction programs that take into account the pairwise contact potential; otherwise, a dynamic programming algorithm can fulfill it.

§ If interaction terms between neighboring amino acids are not allowed

- Dynamic programming methods will efficiently find the optimal alignment between the sequence and core fold template

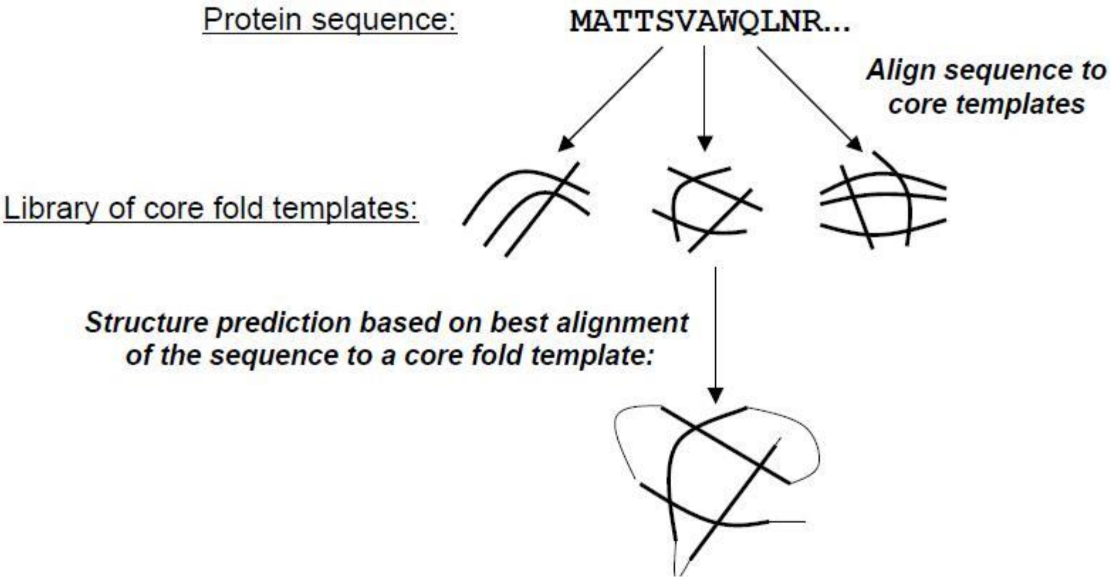
§ If interaction terms between neighboring amino acids in the structure are allowed

- Heuristic methods
  - Fast, but may not find optimal alignment
- Exact methods (e.g., branch & bound, Lathrop and Smith (1996) *J. Mol. Biol.* 255:641-665)
  - Will find the optimal alignment, but can take exponential time

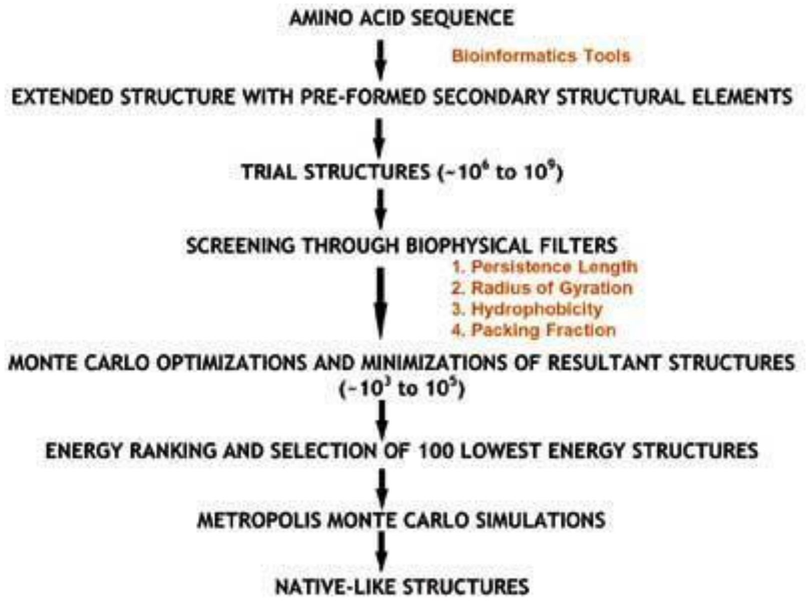


Threading prediction

Select the threading alignment that is statistically most probable as the threading prediction. Then construct a structure model for the target by placing the backbone atoms of the target sequence at their aligned backbone positions of the selected structural template.


Protein sequence:
MATTSVAWQLNR...
Align sequence to core templates
Library of core fold templates:
Structure prediction based on best alignment of the sequence to a core fold template:

AB INITIO PREDICTION METHOD


AMINO ACID SEQUENCE
Bioinformatics Tools
EXTENDED STRUCTURE WITH PRE-FORMED SECONDARY STRUCTURAL ELEMENTS
TRIAL STRUCTURES (~10⁶ to 10⁹)
SCREENING THROUGH BIOPHYSICAL FILTERS
1. Persistence Length
2. Radius of Gyration
3. Hydrophobicity
4. Packing Fraction
MONTE CARLO OPTIMIZATIONS AND MINIMIZATIONS OF RESULTANT STRUCTURES
(~10³ to 10⁵)
ENERGY RANKING AND SELECTION OF 100 LOWEST ENERGY STRUCTURES
METROPOLIS MONTE CARLO SIMULATIONS
NATIVE-LIKE STRUCTURES

Ab initio, or de novo approaches predict a protein structure and folding mechanism from knowledge only of its amino acid sequence. Often the term ab initio is interpreted as applied to an algorithm based entirely on physico-chemical interactions. On the other hand, the most successful ab initio methods utilize information from the sequence and structural databases in some form. Basic idea of an ab initio algorithm: search for the native state which is presumably in the minimum energy conformation. Usually an ab initio algorithm consists of multiple steps with different levels of approximated modeling of protein structure.

For a consideration of side chains in ab initio predictions, a so-called united residue approximation (UNRES) is frequently used:

- Side chains are represented by spheres ("side-chain centroids", SC). Each centroid represents all the atoms belonging to a real side chain. A van der Waals radius is introduced for every residue type.

- A polypeptide chain is represented by a sequence of Cα atoms with attached SCs and peptide group centers (p) centered between two consecutive Cα atoms.

- The distance between successive Cα atoms is assigned a value of 3.8 Å (a virtual-bond length, characteristic of a planar trans peptide group CO-NH).

- It is assumed that Cα - Cα - Cα virtual bond angles have a fixed value of 90° (close to what is observed in crystal structures). - The united side chains have fixed geometry, with parameters being taken from crystal data.

The only variables in this model of protein conformation are virtual-bond torsional angles γ.

The energy function for the simplified chain can be represented as the sum of the hydrophobic, hydrophilic and electrostatic interactions between side chains and peptide groups (potential functions dependent on the nature of interactions, distances and dimensions of side chains). The parameters in the expressions for contact energies are estimated empirically from crystal structures and all-atom calculations.

An example of the algorithm for structure prediction using UNRES:

1. Low-energy conformations in UNRES approximation are searched using Monte Carlo energy minimization. A cluster analysis is then applied to divide the set of low-energy conformations whose lowest-energy representatives are hereafter referred to as structures. Structures having energies within a chosen cut-off value above the lowest energy structure are saved for further stages of the calculation.

2. These virtual-bond united-residue structures are converted to an all-atom backbone (preserving distances between α-carbons).

3. Generation of the backbone is completed by carrying out simulations in a “hybrid” representation of the polypeptide chain, i.e. with an all-atom backbone and united side chains (still subject to the constraints following the UNRES simulations, so that some or even all the distances of the virtual-bond chain are substantially preserved). The simulations are performed by a Monte Carlo algorithm.

4. Full (all-atom) side chains are introduced with accompanying minimization of steric overlaps, allowing both the backbone and side chains to move. Then Monte Carlo simulations explore conformational space in the neighborhood of each of the low-energy structures.

Monte Carlo algorithms start from some (random) conformation and proceed with (quasi)randomly introduced changes, such as rotations around a randomly selected bond. If the change improves energy value, it is accepted. If not, it may be accepted with a probability dependent on energy increase. The procedure is repeated with a number of iterations, leading to lower energy conformations. A function defining higher energy acceptance probability is usually constructed 25 with a parameter that leads to lower probabilities in the course of simulation ("cooling down" the simulation) in order to achieve convergence and stop the algorithm.

Combinations of approaches

Many of the modern packages for protein structure predictions attempt to combine various approaches, algorithms and features. One of the most successful examples is Rosetta - ab initio prediction using database statistics.

Rosetta is based on a picture of protein folding in which local sequence fragments (3-9 residues) rapidly alternate between different possible local structures. The distribution of conformations sampled by an isolated chain segment is approximated by the distribution adopted by that sequence segment and related sequence segments in the protein structure database. Thus the algorithm combines both ab initio and fold recognition approaches.

Folding occurs when the conformations and relative orientations of the local segments combine to form low energy global structures. Local conformation are sampled from the database of structures and scored using Bayesian logic:

P(structure | sequence) = P(structure) x P(sequence | structure) / P(sequence).

For comparisons of different structures for a given sequence, P(sequence) is constant. P(structure) may be approximated by some general expression favouring more compact structures. P(sequence | structure) is derived from the known structures in the database by assumptions somewhat similar to those used in fold recognition, for instance by estimating probabilities for pairs of amino acids to be at particular distance and computing the probability of sequence as the product over all pairs).

Non-local interactions are optimized by a Monte Carlo search through the set of conformations that can be built from the ensemble of local structure fragments.

In the standard Rosetta protocol, an approximated protein representation is used: backbone atoms are explicitly included, but side chains are represented by centroids (so-called low-resolution refinement of protein structure). The low-resolution step can be followed by high-resolution refinement, with all-atom protein representation. Similar stepwise refinement protocols can be used to improve predictions yielded by other methods, for instance, in loops (variable regions) of homology-modeling structures.

In recent CASP experiments (Critical Assessment of Structure Prediction), the Rosetta approach turned out to be one of the most successful prediction methods in the novel fold category. Obviously, none of prediction approaches is ideal. Therefore it is reasonable to try to combine the best features of many different procedures or to derive a consensus, meta-prediction. For instance, the 3D-Jury system generated metapredictions using models produced by a set of servers. The algorithm scored various models according to their similarities to each other.

Predictions of coiled coil domains and transmembrane segments

Special algorithms have been developed for domains characterized by special types of interactions. The coiled coil domains are very stable structures formed by regular arrangement of hydrophobic and polar residues in adjacent α- helices. This is possible in the amino acid sequences containing repeats of seven residues (heptads) with hydrophobic residues located at the first and the fourth positions of the heptad and preferences for polar residues at positions 5 and 7. It is possible to design an algorithm that would take into account stabilizing interactions in coiled coils to predict such conformations. Transmembrane proteins contain α-helical segments buried in the membranes. Due to the specific hydrophobic environment in a membrane, protein folding occurs differently as compared to globular proteins folded in the polar water environment. This leads to special folding algorithms, mostly based on known statistics of amino acid frequencies in transmembrane α-helices. Efficient modern algorithms use probabilistic approaches such as Markov models and Bayesian approach.