Computational Molecular Biology

Lecture Notes

by A.P. Gultyaev

Leiden Institute of Applied Computer Science (LIACS)
Leiden University
January 2019

Contents

Introduction

These lectures in Computational Molecular Biology aim to make participants familiar with a number of widely used approaches for computational analysis of biopolymers. In particular, the purpose is to give an idea about the main principles behind the algorithms for analysis of biopolymer structures and functions. Of course, the text below is just some (lecture) notes that may assist in understanding the material given in lectures and should not be considered as independent comprehensive description.

Basic Molecular Biology Introduction:

Biopolymers (DNA, RNA, proteins) are the main components of life on Earth. DNA and RNA molecules are the polynucleotide chains containing four nucleotides, protein polypeptide chains contain 20 residue types (amino acid residues). The main "dogma" of Molecular Biology states that DNA is a carrier of genetic information, which is transcribed into messenger RNA (mRNA) molecules that are translated to proteins responsible for functioning and reproduction of organisms. Furthermore, multiple types of RNA exist that have important functions in the cell and are not protein-coding (non-coding RNAs, ncRNAs)

Genes carrying genetic information in the majority of species (except some viruses) are the regions of double-helical DNA genomes containing two complementary antiparallel polynucleotide strands that have a so-called 5'->3' polarity (following the nomenclature of atoms in monomer units). Each monomer unit (nucleotide) contains phosphate, sugar (desoxyribose) and one of nucleotide bases: adenine (A), thymine(T), guanine (G) or cytosine (C). The complementarity is determined by Watson-Crick base-pairing in AT and GC pairs. Thus a double-helical DNA molecule is described by the sequence of one of its strands, e.g. TAGCGCAGGG... (in this case the complementary strand in this fragment is ...CCCTGCGCTA). DNA replication mechanism ensures the transfer of genetic information upon reproduction, sequence changes occurring due to replication errors (mutations) may lead to disorders or evolutionary development.

Transcription of DNA into RNA is carried out by RNA polymerase complex which uses one of DNA strands as a template. The resulting RNA sequence is determined by Watson-Crick complementarity rules and thus is the copy of complementary DNA strand sequence, except that thymine is substituted by another nucleotide base, uracil (U). Start and termination points of transcription are important elements of gene sequence analysis. Sequence signals that determine transcription starts, are called promoters. Apart from a protein-coding region, which is translated to amino acid sequence, an mRNA molecule contains two untranslated regions "before" (upstream) and "after" (downstream) of the coding sequence. These regions are called 5' untranslated region (5'UTR) and 3'UTR, respectively. Translation of a nucleotide sequence in an amino acid sequence occurs according to the genetic code, with amino acids encoded by triplets of nucleotides (codons). Genetic code is redundant, as the majority of amino acids can be encoded by different codons, frequently with the last 3rd position being so-called "wobble" one. Start-codons are ATG sequences (or AUG in RNA, encoding for methionine amino acid), stop-codons are TAA, TAG and TGA. The translated region between a start- and stop-codon is called open reading frame (ORF). Due to the triplet coding, for any DNA sequence 6 reading frames are possible: 3 frames in each of the complementary strands.

In prokaryotic organisms (those lacking nucleus in their cells), an mRNA molecule may contain an operon consisting of several ORFs coding for several proteins. In eukaryotes, genes have different complexity: mRNA molecules with ORFs are not transcribed directly, but are produced by processing of precursor pre-mRNA molecules. In the processing (RNA splicing), non-coding pre-mRNA regions (introns) are removed, and the rest (exons) are ligated, leading to mature mRNA molecules. A number of eukaryotic genes are characterised by alternative splicing, leading to various exon combinations in mRNAs and, therefore, different encoded proteins.

Biopolymer sequences contain multiple sequence patterns that regulate their functions. Furthermore, these functions depend on 3D structures of biopolymer molecules, in particular, RNA and proteins. Prediction and analysis of biopolymer 3D structures on the basis of one-dimensional sequences are also essential parts of Bioinformatics and ***Computational Molecular Biology***.

Basic Bioinformatics Sequence Analysis Introduction:

Analysis of monomer sequences of biopolymers is of great importance for understanding all living systems, from viruses to humans. Computational analysis of sequence data involves large databases and a number of algorithms developed for sequence comparisons, recognition of functional sequence patterns and processing of experimental sequencing data. Sequence analysis is the core of bioinformatics.

The widely used nucleotide sequence database is GenBank of the retrieval system Entrez provided by The National Center for Biotechnology Information, NCBI (www.ncbi.nlm.nih.gov). Entrez also include curated databases of nucleotide sequences, e.g. Gene, Genome etc. Protein amino acid sequences are mostly derived from nucleotide sequence entries and stored in curated databases, also accessible in Entrez. Sequence data are also available in the resources of European Bioinformatics Institute (EBI). The exchange of sequence information is mostly done using so-called flat-file sequence entry format. It contains a number of datafields, classified into header, features and sequence parts. The main nucleotide sequence databases exchange the data on regular basis, they also use a unified system of accession numbers.

Sequence alignment is the main approach to compare the sequences and analyse their similarity. The alignment maps the monomers of two or more sequences to each other, what could suggest functional and/ or evolutionary links between them. The alignment task is finding the alignment which is the most likely corresponding to biological relationships between sequences in question. Due to enormous number of all possible alignments even in case of two moderately long sequences, this is not always straightforward. Two essential components of any alignment algorithm are the scoring system and the procedure to calculate highly-scoring alignments. Scoring parameters are defined for matches, substitutions and gaps in alignments. Various scoring systems may be used, the parameters are mostly derived from estimated statistics of mutations in related sequences.

The optimal **pairwise alignment** of two sequences can be found by a **dynamic programming** algorithm that guarantees to find the optimal alignment with the highest score. Dynamic programming is used for finding optimal **global** (Needleman-Wunsch algorithm) and **local** (Waterman-Smith algorithm) pairwise alignments. The optimal local alignment for two sequences is defined as the alignment of their subsequences with the highest score.

One of the frequent alignment tasks is the **sequence database similarity search**. The **query sequence** is aligned to each **subject sequence** in the database, yielding a list of *sequence hits* with relatively high similarity to the query. An application of the optimal alignment computation by dynamic programming is not practical for this task, being rather slow. More efficient approach is to first identify local similarities of short "words" (oligomers) in two sequences, which could be further extended to significant alignments. The most popular algorithm of this kind is BLAST (Basic Local Alignment Search Tool, available at the NCBI resources: www.ncbi.nlm.nih.gov), which implements a strategy to find statistically reliable local alignments.

Alignment of more than two sequences is called **multiple sequence alignment (MSA)**. A number of algorithms is developed for this task. The classical approach is **progressive** MSA. In such an algorithm, first the pairwise alignments are produced, e.g. by dynamic programming. The scores of pairwise alignments are used to cluster the sequences according to their similarities, thus producing a guide tree. The guide tree is used for sequential alignments of (clusters of) sequences, starting from the closely related ones and proceeding to more distant relations. At every step, the alignment of two (clusters of) sequences is actually a pairwise alignment, the difference being is that substitution scores at alignment positions are calculated as average values of the scores for monomers in previously aligned sequences. More complex MSA algorithms have been developed as well, mostly aiming to escape from greedy character of progressive alignment and to solve the problems arising in large datasets of sequences.

None of alignment algorithms can actually guarantee the finding of biologically relevant solution. Therefore alignments are frequently improved by human intervention according to additional data, like protein-coding properties, presence of specific motifs, structures etc.

Identification of regions of high similarity in sequence alignments can lead to recognition of common (consensus) patterns in them, or sequence **motifs**. In turn, a number of algorithms exist to align various descriptors of these motifs to other sequences in order to recognize new pattern-matching domains. The classic form of such a descriptor (*profile*) is a position-specific score matrix (PSSM). Such a matrix has dimensions 4xL for nucleic acids and 20xL for proteins, where L is the motif size. The matrix elements are computed from monomer frequencies in a dataset of sequences assumed to contain the motif. Other types of profiles, e.g. based on Markov models, are also possible. Profile descriptions are used in the databases of nucleotide and amino acid motifs, which can be searched by query sequences.

1. RNA structure prediction

1.1. Thermodynamics of RNA folding and RNA structure predictions

RNA folding occurs in hierarchical way, with secondary structure formed first, followed by the formation of tertiary structure. The secondary structure of an RNA molecule is determined by base-pairing between complementary regions and is usually described as the configuration consisting of double-stranded helices (stems) and loops of different topologies (hairpins, bulges, internal and multibranch loops), formed by various combinations of stems. Tertiary structure is a spatial (3D) conformation of RNA determined by various interactions between atoms within RNA (bases, ribose, phosphates) and with environment (solvent).

Usually the energy content of the 2D structure is very high compared to that of 3D, and the majority of RNA structure prediction algorithms deal with secondary structure prediction without account for 3D-structure elements. The methods for calculating secondary structure are based either on **thermodynamic approach** or **comparative analysis**, or some combination of them. The thermodynamics-based methods may apply free energy minimization (thermodynamics) or folding simulations (kinetics). The approaches for 3D structure predictions are based on stereochemical considerations and approximated energy potentials estimated for loop conformations (e.g. in RNA pseudoknots) and/or for interactions between atomic groups present in RNA.

Free energy ΔG of RNA secondary structure can be computed as the sum of stabilizing free energies determined by base-pairing and stacking of neighboring bases (ΔG < 0) and destabilizing loops (ΔG > 0). Many of these parameters have been estimated from thermodynamic experiments and are available in the Internet (e.g. Turner & Mathews, 2010).

The positive free energies of loops are approximated by logarithmic dependences on their sizes. The approximations of general type

loop ΔG ~ 1.75 RT ln (size)

are used for various loop topologies. This formula is derived from size dependence of conformational entropy (in assumption that enthalpy of loop formation is zero). Some specific sequences, in particular, tetraloops, are extrastable, and usually this is taken into account by addition a negative term in the formula.

The stacking free energies in the stems are calculated according to the nearest-neighbor rules. In helices, the stabilizing free energies, determined by H-bonds and stacking effects, depend on combinations of adjacent base pairs rather than on single pairs. Thus, for calculating the energy of a helix with 5 base pairs 4 stacking values have to be added, see example:

```
 G G
A   A
A   U       ΔG_loop(N=8) = + 5.5 kcal/mol
A   A
 C-G      ΔG_mismatch(CG/AA) = - 1.5 kcal/mol
 U-A      ΔG_stacking(UA/CG) = - 2.4 kcal/mol
 A-U      ΔG_stacking(AU/UA) = - 1.1 kcal/mol
 C-G      ΔG_stacking(CG/AU) = - 2.1 kcal/mol
 G.U      ΔG_stacking(GU/CG) = - 2.5 kcal/mol
```

Stacking between the planar rings of nucleotide bases occurs also in the mismatches and single unpaired nucleotides adjacent to stems. These values also depend on nearest neighbors and are tabulated taking this into account. RNA folding free energies may depend on other contributions such as those arising from pseudoknot configurations, specific sequence motifs and coaxial stacking of helices.

For a *given secondary structure*, it is easy to compute its free energy by simply adding all values for individual structural elements. However, the number of all possible structures for *a given sequence* is very large, and in order to find the one with the lowest free energy the dynamic programming is used. The algorithm used for **energy minimization by dynamic programming** is somewhat similar to the algorithm used for alignment problem. The program mfold (http://mfold.rna.albany.edu/) is the most frequently used.

Using dynamic programming algorithm, it is also possible to compute the equilibrium partition function Q:

$$Q = \sum_{s} \exp [- \Delta G(S) / kT]$$

as the sum over all possible structures S.
Knowing the partition function, one can compute the probability of a given structure S:

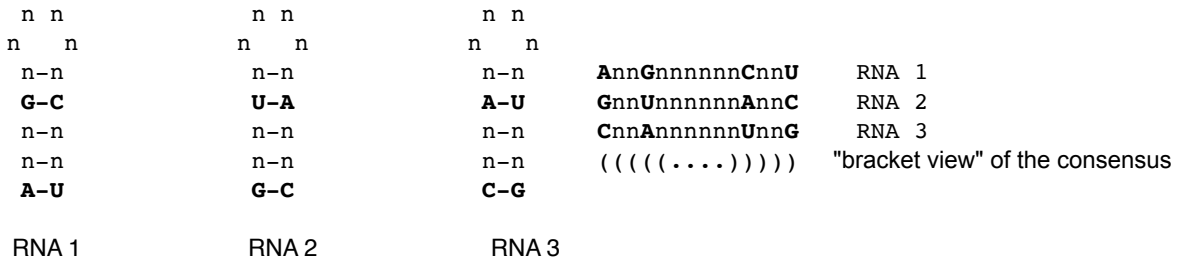
$$P(S) = \frac{\exp [- \Delta G(S) / kT]}{Q}$$

Furthermore, it is possible to calculate the probability of a given base pair.

Apart from the global minimum conformation, analysis of **probable alternative structures** is frequently necessary. For instance, mfold can calculate suboptimal structures, within some ΔG increment of the energy minimum. A convenient way to overview the most likely suboptimal structures is to use *energy dot plots* that contain superpositions of possible foldings (e.g. base pairs from all structures with free energies less than some value). Dot plot is a two-dimensional plot where the sequence positions are on the two coordinates so as any base-pair can be represented by a dot and a helix by a diagonal region.

1.2. Comparative RNA analysis

Prediction of structure using comparative analysis is based on the assumption that functionally related sequences should have similar structures. Comparisons of RNA secondary structures usually exploit a principle of nucleotide **covariations** (coordinated variations), in order to derive a consensus secondary structure for related RNA molecules.



Many models of RNA secondary structures such as rRNAs, RNase P etc. were deduced mostly from comparative analysis. Base covariations are not necessary determined by Watson-Crick pairing, but can be a consequence of **non-canonical base pairs** (e.g.G•A, C•C etc.) that may be found in both helical duplexes and long-range interactions.

A significance of covariations as evidence for base-pairing can be estimated using **mutual information (MI)** calculations, based on Shannon's entropies of paired positions in alignments. For two columns *x* and *y* of multiple sequence alignment the MI value *M(x,y)* is computed as follows:

$$M(x,y) = \sum_{b_x, b_y \in (A,G,C,U)} f(b_x b_y) \cdot log_4 \frac{f(b_x b_y)}{f(b_x) f(b_y)}.$$

Here $f(b_x)$ and $f(b_y)$ are the nucleotide frequencies at positions *x* and *y*, and $f(b_x b_y)$ are the nucleotide combination frequencies. The formula can be rewritten as M(x,y) = H(x) + H(y) - H(x,y), where Shannon's entropy H is defined as

$$H = -\sum f(b) \cdot log_4 f(b)$$

The *M(x,y)* values are in the interval [0,1] and measure a correlation between variations at positions x and y, with value 1 corresponding to ideal covariation and 0 to absence of any correlation. Several other measures, derived from *M(x,y)*, can be used for correlation statistics as well. For instance, normalized MI values M(x,y)/H(x) and M(x,y)/H(y) may compensate for biases in datasets of aligned sequences. On the other hand, MI values alone are frequently not sufficient to conclude about pairwise interactions because high correlation values could be also determined by speciation. Some algorithms are designed to distinguish covariations determined by physical contacts, for instance, by identification of independent covariation events. Similar approaches are also used in protein structure prediction (see section 2.5).

1.3. Detecting conserved structures in related RNA sequences

A number of algorithms is designed to predict most likely conserved structures in datasets of related RNA molecules. For instance, the program RNAalifold of ViennaRNA Web Services (http://rna.tbi.univie.ac.at) predicts the consensus secondary structure for a set of aligned sequences using modified dynamic programming algorithm that adds a covariance term to the standard energy model. The program calculates the probabilities of separate base pairs from the ensemble of suboptimal structures, taking base pair conservation into account. The conservation of base pairs is estimated by a measure that includes a term depending on (co)variation of presumably paired nucleotides in alignment and a term derived from stacking energies with neighboring base pairs.

If RNA sequences are not reliably aligned, the algorithms computing both the alignment and consensus structure are used. Usually they iterate back and forward from the steps improving the structure-based alignment and retrieving the consensus structures from the alignment. Of course, due to complexity of the problem, such algorithms are less accurate than those based on reliable alignments.

1.4. RNA motif search

Several programs have been developed for genome-wide search of known structural motifs (e.g. tRNAs, snoRNAs, iron-responsive elements etc.) using descriptors for consensus structures. Different descriptors with various sequence and structure requirements (scoring functions or threading potentials) may be combined. For instance, below a descriptor in a so-called bracket notation for the consensus structure consisting of a conserved hairpin with a bulge:

```
(((((.(((((......)))))))))))
NNNNNCNNNNNCAGWGHNNNNNNNNNN
```

where N is any nucleotide; W is A or U; H is C, A or U; base-pairs are indicated by parentheses. Such kind of descriptor can be threaded along the sequences in the database to search for matching sequences.

More efficient and flexible descriptions of RNA structural motifs are based on covariance models (CM) derived from alignments of related RNA molecules. Such covariance models, that serve as mathematical descriptions of RNA structures, can be used as queries for the search in a given sequence, genome or database. The idea is somewhat comparable to the use of profiles (PSSMs) in the protein motif search. For instance, such an approach is implemented in the database of RNA families (Rfam, http://rfam.sanger.ac.uk/), where each family is stored in the form of an alignment (“SEED” alignment) that specifies a covariance model used for the search for new family members.

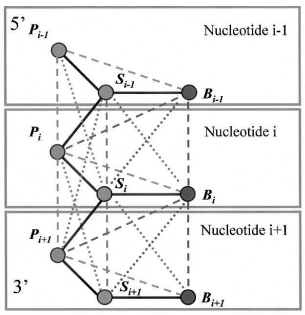
For instance:

```
M14879/224-175          AAACAGAGAAGUCAACCAGAGAAACACACGUUGUG..GUAUAUUACCUGGUA
M17439/226-177          AAACAGAGAAGUCAACCAGAGAAACACACGUUGUG..GUAUAUUACCUGGUA
M21212/157-106          CAACAGCGAAGCGGAACGGCGAAACACACCUUGUGUGGUAUAUUACCCGUUG
#=GC SS_cons            ..............<<<<<.......<<<...>>>..........>>>>>.
```

New sequences (database hits) are aligned to the CM, leading to an extended FULL alignment. The process can be iterated, if some sequences from the FULL are taken to the SEED in order to refine it (this is done in curated way).

1.5 3D RNA structure prediction

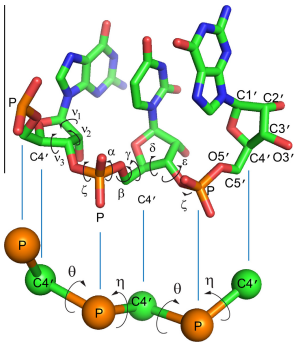
In recent years, the promising approaches for prediction of RNA three-dimensional (3D) structure using molecular dynamics (MD) have been developed. The MD simulations are very demanding computationally, even when they use simplified structural models e.g. so-called “bead-on-a-string” **coarse-grained polymer models** like the one shown below (Ding et al., 2008). In this model, the structure of the polynucleotide chain is reduced to three beads: sugar (S), phosphate (P) and base (B). The RNA folding is simulated using the 3D constraints and the energy functions describing interactions between these units.



(Ding et al., 2008)

Depending on the approximation level, a "bead" or "grain" in a coarse-grained model can be any unit for which interactions with other unit can be defined: an atom, a group of atoms, a nucleoside, or even a helix or an independently folded domain. A number of 3D RNA models of structural RNAs were built sequentially, with the first step being secondary structure prediction followed by rearrangement of secondary structure units according to most likely interactions between them. Such interactions can be derived from nucleotide covariations in secondary structure-based alignments and serve as structural constraints for 3D modeling.

Obviously, **all-atom models of RNA folding** are more complex and computationally demanding. Below the all-atom structure of the GUG sequence is compared to coarse-grained representation of polyphosphate backbone with virtual bonds connecting phosphate and sugar C4' carbon atoms.



(Dawson et al., 2016)

All-atom simulations of RNA folding may use different potential energy functions, so-called **force fields**, derived from various approximations of quantum mechanics. Simulation of RNA folding can be done using Molecular Dynamics that yields possible trajectories that lead to the folded structure(s). In general, all-atom Molecular Dynamics simulations of RNA folding are computationally demanding.

2. Protein structure prediction

The development of algorithms for protein structure prediction from sequence has relatively long history. Various **secondary (2D) structure prediction** algorithms consider the residues in a polypeptide chain to be in three (helix, strand, coil), four (helix, strand, coil, turn) or even eight states and try to predict the location of these states. The majority of methods for protein structure prediction are based on empirical (statistical) approaches. They try to extrapolate the statistics, revealed in the known structures, to other proteins.

The most simple observation:
- *Ala, Gln, Leu and Met are commonly found in α helices;*
- *Pro, Gly, Tyr and Ser usually are not;*
- *Pro – a "helix-breaker" (bulky ring prevents the formation of n/n+4 H-bonds)*

The algorithm of Chou and Fasman (1978) was a widely used approach in 1970's-1980's. It was based on the calculation of a moving average of values that indicated the probability (or propensity) of a residue type to adopt one of three structural states, α-helix, β-sheet and turn conformation. The probabilities were simply the frequencies of a given residue type to be observed in a particular secondary structure, normalized by the frequency expected by chance. Some additional heuristic rules were added that attempted to determine the exact ends of secondary structure elements.

In the further developments of these ideas, the algorithm of Garnier et al. (1978) introduced an estimate of the effect that residues within the region eight residues N-terminal to eight residues C-terminal of a given position have on the structure of that position. For each of 20 amino acids, a profile (17 residues long) was defined that quantifies the contribution this residue makes towards the probabilities of other residues to be in one of four states, α-helix, β-sheet, turn and coil. Thus, for every residue the values from ± 8 positions are added (four values from each), so as four probability profiles are produced, and at any position the highest profile value predicts the structure.

Further developments of prediction methods (until the early 1990s) were based on implementing various rules for pattern recognition, in particular, the **hydrophobicity patterns**, periodicity of helices, different ways for calculating propensities in windows of variable sizes (3-51 residues). However, it seemed that prediction accuracy of such approaches stalled at levels slightly above 60% (percentage of residues predicted correctly in one the three states: helix, strand, and other). Next generation of methods incorporated **multiple alignments** into predictions. They incorporated a concept that homologous proteins should have similar structures. For example, all naturally evolved proteins with more than 35% pairwise identical residues over more than 100 aligned residues have similar structures (Rost, 1999). Predictions were further improved by application of **neural networks**, trained on known structures.

Multiple algorithms for **protein 3D structure prediction** are developed. The modern approaches for protein tertiary structure prediction can be divided into three general strategies:

(1) **Comparative (homology) modeling**. If there is a clear sequence homology between the target and one or more known structures, an algorithm tries to obtain the most accurate structural model for the target, consistent with the known set.
(2) **Fold recognition**. Algorithms that try to recognize a known fold in a domain within the target protein.

(3) ***Ab initio* methods**. Modeling of structures using energy calculations, considering both secondary and tertiary structures.

There are overlaps between the methodologies.

2.1. Homology modeling

Homology modeling approach assumes that a sequence similarity between a target protein and at least one related protein with known structure (the template) implies the 3D-similarity as well, and therefore allows one to extrapolate template structures to the target sequence.

Main steps in comparative protein structure modeling

(Fiser *et al*., 2001)

1. Identify related structures.
2. Select templates.
3. Align target with templates.
4. Build a model for the target (using information from template structures).
5. Evaluate the model.
6. If model is not satisfactory, repeat the steps 2-5 or 3-5.

Identification of structures related to the target sequence is usually done by searching the database of known protein structures (PDB) using the target sequence as the query. For instance, a specific algorithm for template search in PDB is PDB-BLAST that not only builds a multiple alignment using target as a query, but also constructs similar multiple alignments using all found potential templates as queries (here each alignment is called sequence profile, and it can be converted into a matrix). The templates are found by comparing the target sequence profile with each of the template sequence profiles (e.g. by dynamic programming method). This allows to capture essential sequence motifs for a fold to be predicted. Selection of optimal templates is guided by several criteria, such as overall sequence similarity to the target and the quality of the experimentally determined structure. It is not necessary to select only one template: alignments of the target to different templates may be used for model building. These alignments may be refined after template selection. Various algorithms are used for model building, e.g. based on "rigid bodies" or spatial restraint satisfaction (based on core conserved structures obtained from aligned template structures). There are systems for (web-based) automated homology modeling that are able to predict the structures for one or many sequences without human intervention. For instance, **SWISS-MODEL (**http://swissmodel.expasy.org)**,** one of the first servers for protein structure predictions, initiated in 1993 and accessible via the ExPASy web server.

2.2 Fold recognition

In case of relatively low *sequence* similarity of a target protein to the known structures, an attempt may be done to recognize a known *fold* within the target by a search for an optimal sequence-to-structure compatibility. Mostly this is done by **threading algorithms**: a target sequence is threaded through templates from the structure database and alternative **sequence-structure alignments** are scored according to some measure of compatibility between the target sequence with the template structures. The scoring is done using **threading potentials**, for instance so-called **knowledge-based** or **mean-force** potentials. These potentials do not consider physical nature of interactions in proteins and are derived from the statistics of the databases of known structures.

Knowledge-based (database-derived) mean force potentials incorporate all forces (electrostatic, van der Waals etc) acting between atoms as well as the influence of the environment (solvent). For the interaction between two residues (a,b) with a sequence separation k and distance r between specified types of atoms (e.g. C**β**→C**β**, C**β**→N etc.) a general definition of the potential is (“inverse Boltzmann equation”)

$$E^{ab}_k(r) = - RT \ln [f^{ab}_k(r)],$$

where the occurrence frequency $f^{ab}_k(r)$ is obtained from a database of known structures.

A definition of the reference state is very important. A convenient choice for the reference state is

$$E_k(r) = - RT \ln [f_k(r)] ,$$

where $f_k(r)$ is an average value over all residue types.

Thus the potential for the specific interaction of residues is

$$\Delta E^{ab}_k(r) = E^{ab}_k(r) - E_k(r) = - RT \ln [f^{ab}_k(r) / f_k(r)]$$

A **solvation potential** for an amino acid residue a is defined as:

$$\Delta E^{a}_{solv}(r) = - RT \ln (f^{a}(r) / f(r)),$$

where

r is the degree of residue burial,
$f^{a}(r)$ is the frequency of occurrence of residue a with burial r,
f(r) is the frequency of occurrence of an arbitrary residue with burial r.

Obviously, threading algorithms are more complex than sequence-sequence alignments and require some approximations. For instance, in **ungapped threading**, the query sequence is mounted over an equally long part of template fold. The total alignment score is easily computed as sum of pairwise potentials for all query residues. Ungapped threading is mostly used not for real predictions, but rather for testing and adjusting the energy functions.

Treatment of gaps in sequence-structure alignments presents a problem because a score that should be computed for a given residue in a query sequence, assumed to be aligned to a residue in a template structure, would depend not only on the type of these two residues (as in sequence-sequence alignments), but on the gaps that may be introduced at other alignment positions. Here usually a so-called **frozen approximation** is used (e.g. Sippl, 1993). In frozen approximation, an appropriate comparison matrix of the size N×M (N residues in the template and M in the query) is calculated by replacing the amino acids in the template structure with amino acids from the target sequence one at a time. The rest of the structure is kept intact, and it is assumed that the field created by the native protein will also favour the correct replacement. Although it is a very crude approximation, it is rather efficient, despite the fact that it does not solve the full threading problem.

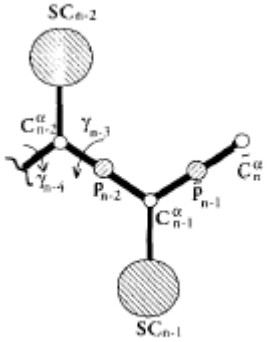
The subsequent steps are equivalent to sequence-sequence alignment: the scores in the comparison matrix may be used for calculating dynamic programming matrix leading to the final alignment.

Often in fold recognition methods several scores are produced, related to different aspects of the sequence-structure alignment, some of the most important: initial sequence profile alignment score, number of aligned residues, length of target sequence, length of template sequence, pairwise energy sum, solvation energy sum. Each of these scores separately may be not sufficient, but they can be used to calculate some determinant score.

2.3. *Ab initio* protein structure prediction

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 C^{α} atoms with attached SCs and peptide group centers (p) centered between two consecutive C^{α} atoms.
- The distance between successive C^{α} 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 C^{α} - C^{α} - C^{α} 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.



(Liwo *et al*., 1993)

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 basic 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 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.

2.4. 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.**

(D. Baker & coworkers)

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.

In such a model, folding can be considered as low-energy combinations of conformations of the local segments and their relative orientations. For instance, local conformations can be 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, initially 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.

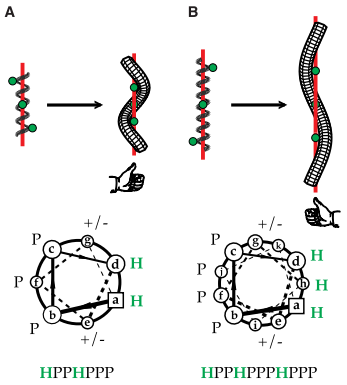
Obviously, none of prediction approaches is ideal. Therefore it is reasonable to try to combine the best features of many different procedures or to derive a consensus, **meta-prediction**. For instance, the **3D-Jury system** generated meta-predictions using models produced by a set of servers. The algorithm scored various models according to their similarities to each other.

2.5. Inference of amino acid residue contacts from covariations

The information on 3D contacts between amino acid residues in protein native structures can be also obtained from the data on coordinated substitutions in the families of structurally similar proteins. The approach resembles the covariation analysis of RNA structural models (section 1.2). Alignments of related protein amino acid sequences are compiled in several databases, e.g. protein families database Pfam. In such families, the pairs of amino acid residues located close to each other in similar 3D structures may co-evolve. Thus, pairwise correlations of amino acid variations could be used to infer the contacts that determine 3D structure. Similar to studies on RNA covariations, mutual information (MI) turned out to be a poor measure of 3D contact probability, because correlated monomer pairs do not necessarily directly interact. Nevertheless, several algorithms using various measures for contact prediction have been used to derive 3D structural constraints in protein families. For instance, approaches to identify maximally informative couplings in global maximum entropy or Bayesian network models over the whole lengths of alignments turned out to be better predictors of 3D contacts as compared to local statistics like MI values. Such 3D contacts can be used as structural constraints in protein structure predictions.

2.6. Predictions of coiled coil domains and transmembrane segments

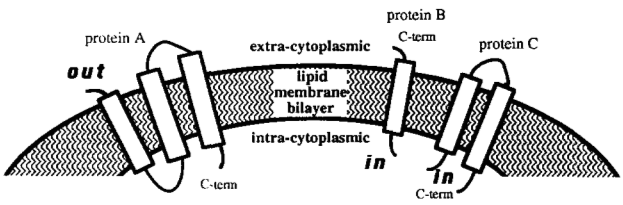
Special algorithms have been developed for domains characterized by special types of interactions. The **coiled coil** domains are very stable structures formed by regular arrangement of hydrophobic and polar residues in adjacent α-helices. In the left-handed supercoil, each of the helices contain repeats of seven residues, usually denoted $(a\text{-}b\text{-}c\text{-}d\text{-}e\text{-}f\text{-}g)_n$ and $(a'\text{-}b'\text{-}c'\text{-}d'\text{-}e'\text{-}f'\text{-}g')_n$. In normal α-helix, each residue would rotate about 100° around helix axis, thus 7 residues would rotate 700° (20° less than two full turns). Two slightly left-handed supercoiled (20° every 7 residues) helices can face each other at the axis of superhelical rotation with the same positions of the repeats. In left-handed coiled coils, the residues a and d are usually nonpolar (e.g. Leu, Val, Ile), yielding hydrophobic interactions with a' and d', while e and g are charged (e.g. Glu, Lys), maintaining electrostatic interactions. Positions b, c and f are typically hydrophilic. In the right-handed supercoil, there are repeats of 11 residues (11 × 100° = 1100°, that is, 20° more than three full turns). Here in the repeat $(a\text{-}b\text{-}c\text{-}d\text{-}e\text{-}f\text{-}g\text{-}h\text{-}i\text{-}j\text{-}k)_n$ positions a, d and h are hydrophobic. Two types of supercoils are shown below to illustrate how supercoiling brings repeat units (heptads or undecatad) in identical positions relative to the superhelix axis, as seen in the helical wheel projections.



(Harbury et al., 1998)

The most simple approach to predict coiled coils was based on the frequencies of amino acids found in each of the seven positions in the heptad repeats contained in the database. These frequencies are used to calculate the scores for a given sequence that determine the probability to form a left-handed coiled coil. Such an approach can be further improved to include the frequencies of each pair of residues in each pair of heptad positions. Furthermore, an extension of this algorithm allows to identify three-stranded coiled coils as well.

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. In the first approximation, the sequences of transmembrane proteins can be represented as transmembrane helical segments of high hydrophobicity alternating with the hydrophilic loops inside or outside the membrane. 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.



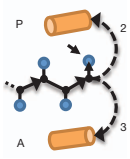
(Rost et al., 1996)

3. Biomolecular design

The so-called inverse folding task is the search for a sequence that will fold into a desired structure. Unlike the folding problem with presumably single solution (native state), the inverse problem may have many solutions. There are examples of structurally similar biopolymers (proteins or RNAs) with very different monomer sequences.

Computationally, inverse folding (or design) is not an easy problem. Importantly, it is not sufficient to find a sequence that may fold into a given topology, it is also necessary to ensure that the sequence would not fold into any alternative structure (because of lower free energy). Refinement of designed structure can be based on probabilistic algorithms that gradually improve the quality of solution (sequence folded into unique structure).

The first successful protein designs were restricted by relatively simple topologies such as coiled coils. A breakthrough was achieved in 2003, when a novel 93-residue fold was designed using a computational strategy with multiple iterations back and forwards between sequence design and structure refinement (Rosetta) in order to produce a desired topology. Recently a significant progress was achieved in both algorithms for protein design and understanding of important elements of protein structure that may by used as standard "building blocks". For instance, the units with two secondary structure elements connected by loops, such as ββ-, βα- or αβ-units, have preferred orientations depending on loop lengths, what can considerably decrease computational complexity of protein design and guide it towards stable solutions. An example of fundamental rule in βα-units:



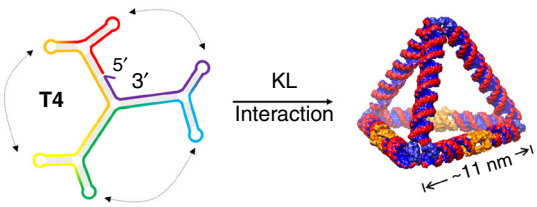
Loop = 2 => orientation P is more likely.
Loop = 3 => orientation A is more likely.

(Koga et al., 2012)

It has been also noted that the complexity of natural proteins has been partially evolved from combinations of ancestral folded peptides. A strategy to compose functionally active proteins from a pool of well-characterized structured domains can be also applied in computational protein design.

Inverse folding may be also used for design of RNA secondary structures, with the same main principle: searching for a sequence with the lowest free energy conformation consistent with the predefined structure. Such a search can be based on minimization of structural difference between the minimum free energy conformation of designed sequence and the target structure. Computational approaches for RNA design can also exploit the usage of previously known folded blocks as initial structures that can be improved by a design algorithm. Apart from the efforts to design stable equilibrium RNA conformations, the protocols for RNA switches (riboswitches) between alternative conformations were developed as well. Such inducible switches have a number of interesting applications, e.g. in making RNA sensors or molecular gates coding for Boolean operators AND, OR, NOR in "ribocomputing" devices.

Design of nucleic acid structures (RNA and DNA) is increasingly used in nanobiology. In so-called RNA or DNA origami the nanostructures can be assembled in programmable way from the building blocks that are base-paired to form diverse shapes like a cube or more complex configurations, nanocages etc. An example of using "kissing" loop (KL) interactions to design a 3D RNA tetrahedron:



(M. Li et al., 2018)

4. Molecular docking and computation of protein-protein interactions

Molecular docking strategies identify the orientations of molecules that are optimal for their interactions. The application of these approaches for protein-ligand or protein-protein interactions play an important role in drug discovery.

In the most simple approximation of protein-ligand interaction, molecular docking is based on lock-and-key assumption that considers both protein and ligand being **rigid bodies** with affinity proportional to geometric fit between them. The fit is searched in six-dimensional rotational/translational space. Rigid body docking can be also carried out with account of binding free energy. The binding free energy potentials can be calculated as the sum of van der Waals, electrostatic and hydrogen-bonding interaction energies. Additional factors such as interactions with solvent can be included as well. Rigid body docking algorithms consider large numbers of docked conformations that can be scored using calculated free energies. Two main components of a docking protocol are a scoring function and a strategy of searching for highly-scoring configurations. The search can be carried out by various algorithms, e.g. Molecular Dynamics, Monte Carlo simulations, genetic algorithms, evolutionary programming.

Rigid body docking is sometimes less accurate when applied to protein crystal structures obtained with unbound proteins. The main reason is that the conformations of both receptor and ligand change upon the binding. This is taken into account by **"induced-fit" or flexible docking**. Obviously, flexible docking involves multiple degrees of freedom and in general more computationally demanding. Different approximations can be used. For instance, a flexible ligand docking into rigid receptor, or limiting flexibility only to side chains with a rigid backbone.

The information on **protein-protein interactions** is very important for understanding their function. Interactions between proteins can be classified as *physical* (direct interactions) and *functional* ones (involvement in the same process). Interacting proteins could be also combined in a *protein network* (based on both physical and functional interactions).

Prediction of structures of protein-protein complexes is computationally more demanding than modeling of receptor-ligand complexes. Rigid-body docking has been applied for protein dimers and some other protein-protein interactions. However, conformational changes in proteins upon binding are an important and challenging problem for accurate structure prediction of protein complexes. Various approximations and search programs can be used. For instance, displacement of rigid bodies combined with optimization of side-chain conformations by Monte Carlo procedure turned out to be successful predictions. The algorithms for multimeric threading, somewhat similar to fold recognition structure predictions, were also developed.

Functional associations between proteins are derived from different sources. A number of computational procedures is designed for integration of this knowledge. Thus, the interactions can be derived directly from experimental data (available in primary databases), information on metabolic pathways available in curated databases, automated text-mining of PubMed or collections of full-text articles, the databases containing gene (co)expression data, orthology relations etc.

Recommended reading and Internet Resources

Basic Bioinformatics:

Sayers EW *et al*. (2019). Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res*. **47**:D23-D28.
(Reference source/summary/update of NCBI resources, in particular, Entrez retrieval system.)

Eddy SR (2004a). What is dynamic programming? *Nature Biotechnology* **22**:909-910

Eddy SR (2004b). Where did the BLOSUM62 alignment score matrix come from? *Nature Biotechnology* **22**:1035-1036.

Altschul SF *et al*. The statistics of sequence similarity scores. (BLAST tutorial). http://www.ncbi.nlm.nih.gov/BLAST/tutorial/Altschul-1.html

Thompson JD, Higgins DG & Gibson TJ (1994). CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.* **22**:4673-4680.

RNA secondary structure prediction:

Lorentz R *et al*. (2016). Predicting RNA secondary structures from sequence and probing data. *Methods* **103**:86-98.

Bernhart SH, Hofacker IL, Will S, Gruber AR, Stadler PF (2008). RNAalifold: improved consensus structure prediction for RNA alignments. *BMC Bioinformatics* **9**:474.

Protein secondary structure prediction:

Fiser A (2010). Template-based protein structure modeling. *Methods Mol Biol* **673**:73-94.

Baker D & Sali A (2001). Protein structure prediction and structural genomics. *Science* **294**:93-96.

Rohl CA, Strauss CEM, Misura KMS & Baker D (2004). Protein structure prediction using Rosetta. *Methods Enzymol* **383**:66-93.

Biomolecular design:

Koga N *et al.* (2012) Principles for designing ideal protein structures. *Nature* **491**:222-227.

Green AA, Silver PA, Collins JJ, Yin P (2014) Toehold switches: de-novo-designed regulators of gene expression. *Cell* **159**:925-939.

Han D *et al*. (2017) Single-stranded DNA and RNA origami. *Science* **358**(6369):eaao2648.

Internet resources:

NCBI homepage: http://www.ncbi.nlm.nih.gov/

European Bioinformatics Institute: http://www.ebi.ac.uk

The mfold Web Server: http://unafold.rna.albany.edu/?q=mfold/RNA-Folding-Form

ViennaRNA Web Services: http://rna.tbi.univie.ac.at

The ExPASy (Expert Protein Analysis System): http://www.expasy.org