8 Protein tertiary structure

Sources for this chapter, which are all recommended reading:

- D.W. Mount. *Bioinformatics: Sequences and Genome analysis*, Cold Spring Harbor Press, Chapter 9: Protein classification and structure prediction. pages 381-478, 2001.
- Nick Alexandrov, Ruth Nussinov, and Ralf Zimmer. *Fast protein fold recognition via sequence to structure alignment and contact capacity potentials.* In Lawrence Hunter and Teri E. Klein, editors, Pacific Symposium on Biocomputing'96, World scientific publishing company, pages 53-72, 1996.
- K.T. Simons, C. Kooperberg, E. Huang and D. Baker, *Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions*, JMB 268:209-225 (1997).

8.1 Hierarchy of protein structure

K.U. Linderstrom-Lang (Linderstrom-Lang & Schnellman 1959) proposed to distinguish four levels or protein structure:

- The *primary structure* is the chemical structure of the polypeptide chain(s) in a given protein, i.e. its sequence of amino acid residues that are linked by peptide bonds.
- The *secondary structure* is folding of the molecule that arises by linking the C=O and NH groups of the backbone together by means of hydrogen bonds.
- The *tertiary structure* is the three dimension structure of the molecule consisting of secondary structures linked by "looser segments" of the polypeptide chain stabilized (primarily) by side-chain interactions.
- The *quaternary structure* is the aggregation of separate polypeptide chains into the functional protein.

Pathway for folding a linear chain of amino acids into a three-dimensional protein structure:

linear polypeptide chain → secondary structures → secondary structures folded into 3D structure

The tertiary structure of proteins is of great interest, as the shape of a protein determines much, if not all, of its function.

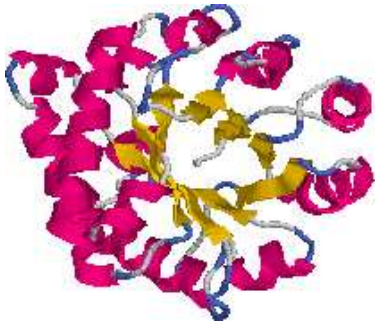
At present, the experimental determination of protein structure via x-ray crystallography is difficult and time-consuming. Hence, we would like to be able to determine the structure of a protein from its sequence. Determining the secondary structure is an important first step.

8.2 The "Holy Grail" of bioinformatics

The *holy grail* of bioinformatics: develop and algorithm that can reliably predict the structure (and thus function) of a protein from its amino-acid sequence!

```
...
IIFIATTNLLGLLPHSFTPTTQLSMNLAMAIPLWA
GAVILAHFLPQGTPTPLIPMLVIIETISLLIQPAL
AVRLTANITAGHLLMGSATLAMTLIIFTILILLTI
LEIAVALIQAYVFTLLVSLYLHDNTPQLNTTVWPT
MITPMLLTLFLITQLKMLPWEPKWADRWLFSTNHK
DIGTLYLLFGAWAGVLGTALSLLIRAELGQPGNLL
GNDHIYNVIVTAHAFVMIFFMVMPIMIGGFGNWLV
PLMIGAPDMAFPRMNNMSFWLLPPSLLLLLASAMV
...
```

$\xrightarrow{?}$



Protein structure prediction consists of three main areas: *fold recognition*, *comparative modeling* and *de novo fold prediction*. We will look at each in the following lectures.

8.3 The fold recognition problem

As of June 2000, more than 12500 protein structures had been deposited in the Brookhaven Protein Data Bank (PDB), and 86500 protein sequence entries were contained in the SWISSProt protein sequence database.

Structural alignments have revealed that there are more than 500 common structural folds that are found in the domains of these protein structures.

The **fold recognition problem** can be formulated as follows:

> Given a protein sequence of unknown structure and a database of representative folds, identify the most plausible fold for the sequence, if there is one, and assess the quality or reliability of the proposed structure.

A mapping of the target sequence on to one of the known structures is called a *sequence-structure* alignment or a *threading*.

8.4 The 123D fold recognition method

In the following, we will discuss one such *threading* method called *123D*, as described in the paper by Alexandrov, Nussinov and Zimmer (1996).

- Using simple empirical potentials, this approach optimizes mappings of residues in the target sequence x onto structural positions of any of the proposed folds.
- The resulting alignments are then evaluated and ranked according to the potential.
- Finally, the statistical significance of the best alignment is estimated in comparison with the other alignments.

8.5 Potentials

The method uses statistically derived *potentials* computed from a non-redundant set of approx. 150 representative protein structures.

An empirical free energy function is used that is given by the sum of three terms:

- *secondary structure* preferences,
- *pairwise contact* potentials, and
- *contact capacity* potentials.

Given a target sequence $x = (x_1, x_2, \ldots, x_m)$ and a proposed fold $y = (y_1, y_2, \ldots, y_n)$, dynamic programming is used to find an optimal threading of x through y, i.e. an an alignment of x to y that minimizes the free energy function.

8.6 Secondary structure preference

Each position $j = 1, \ldots, n$ in the fold y is assigned a secondary structure class $s(j) \in \{\texttt{alpha}, \texttt{beta}, \texttt{other}\}$.

The *secondary structure preference (SSP)* of an amino acid a to be found in a secondary structure of class k is calculated for each of the 20 amino acids as follows:

$$SSP(a,k) := -\log \frac{P^k(a)}{\langle N(a,k) \rangle}, \text{ with } \langle N(a,k) \rangle = \frac{N(a) \times N(k)}{N}.$$

These numbers are obtained by counting occurrences in the given database of known structures:

- $N(a,k)$ is the number of amino acids of type a contained in a secondary structure of class k,

- $\langle N(a,k)\rangle$ is the expected number of residues of type a to be contained in a secondary structure of class k,
- $N(a)$ is the number residues of type a,
- $N(k)$ is the number of residues contained in a secondary structure of class k, and
- N is the total number of amino acids.

This is very similar to the the singleton characteristic used in the context of secondary structure prediction. Typical values obtained for the SSP potential are shown here (multiplied by 100 for clarity):

	ALA	CYS	ASP	GLU	PHE	GLY	HIS	ILE	LYS	LEU	MET	ASN	PRO	GLN	ARG	SER	THR	VAL	TRP	TYR
ALPHA	-33	44	0	-24	-1	38	4	1	-9	-24	-23	4	78	-22	-17	14	23	20	-2	19
BETA	32	-15	27	33	-18	11	0	-39	9	-1	9	27	6	22	15	7	-21	-45	-12	-22
OTHER	22	-20	-15	7	16	-31	-4	37	3	32	22	-19	-44	10	8	-16	-3	23	12	0

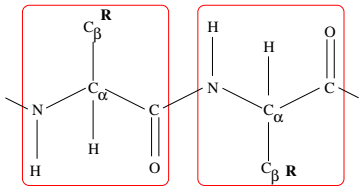
(Alexandrov, Nussinov and Zimmer, 1996)

This confirms the observation that amino acids Ala, Glu, Leu and Met are often found in α helices, whereas Pro, Gly, Ser, Thr and Val occur rarely in them.

Similarly, amino acids Val, Ile, Try and Thr prefer β sheets, whereas Glu, Gln, Lys, Asp, Pro and Cys are seldomly found in them.

8.7 Pairwise contact potentials

Two residues x_i and x_j are defined in be *in contact*, if the distance between their C_β atoms is less than 7.0 Angstrom (S. Miyazawa and R.L. Jernigan 1985).



Note that Glycine has no C_β atom, so fake coordinates must be calculated from the backbone for amino acids of type Gly.

The contact potentials for amino acids a and b are calculated as follows:

$$CP(a,b) := -\log \frac{N(a,b)}{\langle N(a,b)\rangle}, \text{ with } \langle N(a,b)\rangle := \frac{N(a) \times N(b)}{N}.$$

These numbers are obtained by counting occurrences in the given database of known structures:

- $N(a,b)$ is the actual number of residues a and b in contact,
- $\langle N(a,b)\rangle$ is the expected number of contacts between an amino acid of type a and one of type b,

- $N(a) = \sum_z N(a,z)$ and $N(b) = \sum_z N(z,b)$ are the total number of contacts of an amino acid a or b, respectively, and
- $N = \sum_{a,b} N(a,b)$ is the total number of pairs of amino acids in contact.

8.8 Contact capacity potentials

The *contact capacity potential (CCP)* characterizes the ability of a residue to make a certain number of contacts with other residues. Its role is to account for the hydrophobic contribution to the free energy, as obviously, hydrophobic residues will prefer to have more contacts than hydrophilic ones.

For each type a of amino acid, its ability to form r contacts is given by:

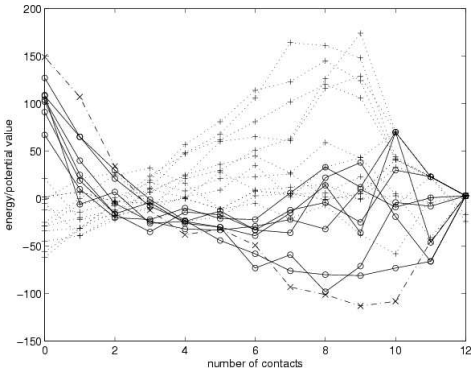
$$CCP(a,r) := -\log \frac{N(a,r)}{\langle N(a,r) \rangle}, \text{ with } \langle N(a,r) \rangle := \frac{N(a) \times NC(r)}{N}.$$

Again, these numbers are obtained by counting occurrences in the given database of known structures:

- $N(a,r)$ is the number of residues of type a having precisely r contacts,
- $\langle N(a,r) \rangle$ is the expected number of residues having r contacts,
- $N(a)$ is the number of residues of type a,
- $NC(r)$ is the number of residues having r contacts, and
- N is the total number of residues.

A contact is called *local*, if less than five residues lie in the sequence between the two residues in contact. In practice, it makes sense to distinguish between *local* and *long-range* contact capacity potentials:

There is a clear correlation between hydrophobicity and long-range contact capacity potentials, indicated here for hydrophobic (circled), polar (dotted) and Cysteine residues (dashed):



(Alexandrov, Nussinov and Zimmer, 1996)

Local contact capacity shows some correlation with secondary structure preferences: obviously, those amino acids that have a preference to be in an α helix tend to have more local contacts.

The contact capacity potential can be refined further in a number of ways, e.g. by considering the secondary structure, the actual distances of contacts, or even the angles of contacts, thus obtaining up to 648 different contact capacity potentials.

8.9 Secondary structure dependent CCP

The ability of residues to make contacts may depend on the secondary structure: residues in α helices have less vacant surrounding space for contacting residues than ones in β strands.

Thus, we obtain six types of SS-contact capacity potentials:

$$\left\{ \begin{array}{c} \text{local} \\ \text{long} - \text{range} \end{array} \right\} \times \left\{ \begin{array}{c} \text{alpha} \\ \text{beta} \\ \text{other} \end{array} \right\}.$$

A significant different between the long-range CCP for α helices and the one for β strands (values $\times 100$) can be observed:

#contacts:	0	1	2	3	4	5	6	7	8	9
ALA	8	35	39	-10	-40	-36	-22	-3	0	5
CYS	95	78	26	-36	-64	-24	-73	-111	-133	5
ASP	-61	-20	16	72	104	115	79	53	40	5
GLU	-60	-20	22	54	95	104	134	99	40	5
PHE	107	11	-44	-49	-19	20	11	54	1	5
GLY	-14	13	32	52	-3	-35	-59	-85	-31	-47
HIS	31	-43	-46	14	35	77	59	-34	40	5
ILE	108	38	-7	-34	-44	-46	-60	-33	-34	-26
LYS	-49	-38	-8	60	134	165	130	122	40	5
LEU	109	27	-21	-41	-41	-34	-12	-22	22	5
MET	82	2	-18	-34	-13	-42	-18	63	-17	5
ASN	-36	-1	6	21	39	50	40	-13	1	5
PRO	-28	10	2	31	37	19	-37	-14	-96	5
GLN	-25	-25	-10	30	33	65	73	85	40	5
ARG	-6	-44	-23	5	80	96	70	87	40	5
SER	-26	1	18	31	8	-5	17	-5	-14	5
THR	-9	13	7	10	12	-31	-18	-16	-21	5
VAL	106	48	21	-38	-48	-61	-79	-62	9	5
TRP	99	-39	-25	-20	-7	-35	81	122	40	5
TYR	100	-5	-33	-31	-33	0	11	70	-11	5

long-range CCP for α helices

#contacts:	0	1	2	3	4	5	6	7	8	9	10	11	12
ALA	9	26	23	10	-5	0	-9	-19	-31	-3	-3	9	-24
CYS	197	124	71	8	-1	-4	-18	-51	-67	-77	-47	-48	3
ASP	-59	-59	-38	-4	-6	19	68	104	89	58	47	9	-31
GLU	-55	-41	-58	-28	3	26	92	91	123	103	9	9	3
PHE	91	35	24	7	-4	-19	-33	-21	18	-14	47	9	3
GLY	-74	-21	9	17	33	0	25	4	10	-8	-67	-14	3
HIS	7	-1	13	-43	-21	-6	45	22	27	76	47	9	3
ILE	153	105	55	30	19	1	-57	-40	-73	-39	24	-13	3
LYS	-40	-50	-56	-28	2	9	89	125	119	141	18	9	3
LEU	101	90	55	21	-3	-33	-36	-22	-35	4	-7	-12	3
MET	86	50	12	7	-9	-17	-45	-2	10	66	47	9	3
ASN	-41	-63	-42	0	5	34	48	54	52	18	47	9	3
PRO	-17	-54	-19	-6	-10	43	24	29	39	-3	9	9	3
GLN	-36	-48	-25	-41	-12	50	47	53	114	87	47	9	3
ARG	-24	-9	-5	-45	-27	-2	66	51	108	141	47	9	3
SER	-42	-42	-28	1	-3	36	40	27	17	29	-23	9	3
THR	-24	-6	-26	-23	0	5	42	27	20	16	47	9	3
VAL	81	77	64	33	8	-14	-28	-54	-43	-64	-37	-7	3
TRP	127	65	35	-13	-20	20	-40	-35	10	13	-36	9	3
TYR	103	52	23	41	-10	-35	-41	-10	17	-28	47	9	3

long-range CCP for β strands

(Alexandrov, Nussinov and Zimmer, 1996)

8.10 Alignment using CCPs and sequence information

Given a target protein sequence $x = (x_1, \ldots, x_m)$ and a sequence $y = (y_1, \ldots, y_m)$ whose fold is known, we use dynamic programming to determine the best scoring alignment between the two sequences.

Recall that slightly different algorithms are used, depending one whether one is interested in the best global-, local or e.g. overlap alignment. All three are variations of the following

recurrence:

$$\begin{aligned} M(i,j) &= \max\left(M(i-1,j-1), I_x(i-1,j-1), I_y(i-1,j-1)\right) \\ &\quad +\text{match}(x_i, y_j), \\ I_x(i,j) &= \max_{k<i}\left(M(i-k,j) - g_x(i,j,k)\right), \\ I_y(i,j) &= \max_{k<i}\left(M(i,j-k) - g_y(i,j,k)\right). \end{aligned}$$

This recursion defines the maximal score $M(i,j)$ of the alignments of the i- and j-prefixes of the two sequences x and y.

To take the introduced potentials into account, the term for single matches can be modified as follows:

$$\text{match}(i,j) = \alpha \times s(i,j) + \beta \times l(i,j) + \gamma \times cc(i,j),$$

where

- $s(i,j)$ is the sequence score of substituting y_j by x_i using an appropriate BLOSUM or PAM matrix,
- $l(i,j) = SSP(x_i, s(j))$ scores the local preference of the i-th amino acid x_i to be in the secondary structure class $s(j)$ of the amino acid at site j of the folded sequence y, and
- $cc(i,j) = CCP(x_i, nc(j))$ denotes the contact capacity score achieved when mapping x_i to position j, i.e. the energy assigned to amino acid x_i to have $nc(j)$ contacts, where $nc(j)$ is the number of contacts that position j in the folded sequence actually has.

The variables α, β and γ are weighting factors relating the different contributions, usually set to 0 or 1 to turn different contributions off and on.

For smoothing, an averaged match score can be obtained by averaging over a window of length $2w+1$ centered at the match in question, usually with $w = 3$:

$$\overline{\text{match}}(i,j) := \frac{1}{2w+1} \sum_{k=-w}^{w} \text{match}(i+k, j+k).$$

Gaps in the alignment are penalized using an affine gap score with gap open penalty d and gap extension penalty e:

$$g(i,j,k) = \begin{cases} \sigma(d+ke) & \text{if } s(j) \in \{\texttt{alpha}, \texttt{beta}\}, \\ (d+ke) & \text{otherwise.} \end{cases}$$

Typical values used are $d = 10$ to 80 and $e = \frac{d}{10}$. The value of σ is either 1 or 10, depending on whether gaps for secondary structures are to be weighted.

8.11 Assessing potentials and threading methods

There are a number of tests that can be used to assess the performance of the potentials and associated optimization procedures for fold recognition:

The *shuffle* test is a simple statistical test: given a *native* score for an optimization problem, consider many permutations of the input problem and then express the native score in terms of standard deviations of the randomized scores.

In the *Sippl* test (M. Sippl 1990), the given target sequence is aligned against all sequences in a given fold database in all possible gap-free ways. The score for all these evaluations are computed and then the native score in expressed in terms of standard deviations. In effect, the Sippl test performs threading without gaps.

The *threading test* is to use the respective method and potential, try align a given sequences as well as possible to any fold of a fold database, evaluate, score and rank the resulting alignments. The native (identity) threading should be the best alignment of the sequence onto its native fold, and the score of this combination should be better than the score of all non-native combinations.

To perform the shuffle test or Sippl test, all that is needed is a program that can evaluate a given threading. The threading test is the most realistic test, but requires a full implementation of the proposed method.

The Sippl test was applied to 167 sequences, of which 139 were longer than 60 residues, yielding the following percentages correct fold recognitions:

	using SSCCP+ SSP	DCCP	ACCP
all 167 sequences	86.8%	83.8%	89.2%
139 ($>$ 60 residues)	94.2%	93.5%	97.1%

Here, SSCCP+SSP means using the secondary-structure dependent contact capacity potential, DCCP means using a potential taking contact distances into account and ACCP means using a potential taking contact angles into account.

Addition experiments using the Sippl, shuffle and threading test show that the approach indicated here produces useful results.

The program 123D runs on the web at:

`http://cartan.gmd.de/ToPLign.html`

8.12 De novo structure prediction

As we have seen, threading methods can be used to identify a suitable structure for a given protein sequence in a database.

Alternatively, sequence alignment can be used to identify a family of homologous proteins that have similar sequences ($\geq$ 30% sequence identity, say) and thus similar three-dimensional structures. The structure of members of a family or even super family can often be determined in this way.

However, because $40 - 60\%$ of proteins in newly sequenced genomes do not have significant sequence homology to proteins of known structure, the problem of determining new structures is important.

The **de novo protein folding problem** can be formulated as follows:

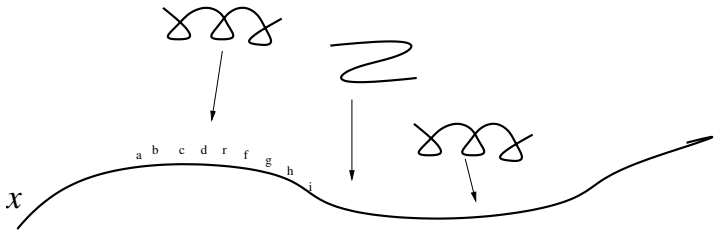
> Given a protein sequence of unknown tertiary structure, determine its structure *in the absence* of close homologs of known structure.

8.13 De novo prediction using ROSETTA

The most successful *de novo* fold prediction program at present is probably ROSETTA, due to David Baker and others (see `http://depts.washington.edu/bakerpg`).

ROSETTA is based on the assumption that the distribution of conformations sampled for a given nine residue segment of the target sequence x is reasonably well approximated by the distribution of structures adopted by the segment (and closely related sequences) in known protein structures.

The predicted structure is pieced together from the structures corresponding to the segments:



In the following we will discuss the general approach of ROSETTA as described in K.T. Simons, C. Kooperberg, E. Huang and D. Baker (1997). Much work has been done to improve the algorithm in the last five years.

As in the case of threading, structures are represented using a simplified model consisting of the heavy atoms of the main-chain and the C_β atoms of the side-chain. (For glycine residues, a fake C_β atom is used). All bond lengths and angles are held constant according to the ideal geometry of alanine, the only remaining variables are the backbone torsion angles ϕ and ψ.

Initial studies showed that there is a stronger correlation between local sequence and local structure for nine residue fragments than for other fragment lengths less than 15.

Hence, ROSETTA attempts to build structures from segments of 9 residues.

8.14 Nearest neighbors

ROSETTA uses a database of sequences with known structures selected from PDB. This is used to compute frequency distributions for all 20 amino acids at each position i of any nine-residue segment of each of the given sequences, additionally using multiple alignments from other databases and pseudo counts.

Every nine-residue segment x' in the given sequence $x = (x_1, x_2, \ldots, x_L)$ is compared with

every nine-residue segment y' of every sequence y in the database using the amino-acid frequency distributions at each position in the two segments:

$$DIST := \sum_{i=1}^{9} \sum_{a} |y'(a,i) - x'(a,i)|,$$

where $x'(a,i)$ or $y'(a,i)$ is the frequency of amino acid a at position i of x' or y', respectively.

Using this distance measure, for each nine-residue segment x' in the target sequence x, the set $N(x')$ of 25 *nearest neighbor* segments in the database are identified.

By applying this computation to target sequences with known structures, it can be verified that the structural similarity of the 25 neighbors to the structure of the target segment is higher than would be expected by chance alone:

Sequence information used to find fragments	% similar
Multiple sequence alignment	20.8
Single sequence	17.5
Random sequence	8.0

So, for example, in the case that the frequency distributions are obtained using multiple alignments, on average, 20.8% of the 25 nearest neighbors determined for a nine-residue segment x' lie within 1 Angstrom "distance matrix error" from the native structure of the segment.

8.15 Ideal bonds and torsion angles

The conformation of each nine-residue segment x' in the target sequence x is chosen from the list of *template* structures adopted by the 25 nearest neighbors.

As the template structures come from PDB, they are not based on ideal bonds and angles. Hence, the torsion angles found in the template structures can not be used directly, as this would lead to significant inaccuracies.

To address this problem, for each such template structure t, a random torsion angle search is performed to find a new conformation t' that assumes ideal bond lengths and angles and has a low *rmsd* (root mean squared distance) to t.

Torsion angles for the nearest neighbors were taken from these idealized structures.

8.16 The consistency of structure

Consider a nine-residue segment x' of x. If all 25 nearest neighbor sequences in $N(x')$ have very similar structures, then one can expect that a structure prediction for x' based on these similar structures will be more reliable than in the case when these structures are very varied.

Indeed, Such a correlation between the amount of structure variation in a given set of templates and the reliability of the prediction has been shown to hold.

This observation could be used to choose optimal fragment sets for building up a structure from the many local segments with different boundaries and lengths which cover each position in the sequence.

8.17 Simulated annealing

The *simulated annealing* method employs a *temperature* that cools over time (Van Laarhoven and Aarts, 1987). At high temperatures the search can move more easily to solutions whose score is less optimal than the score of the current solution. As the temperature decreases, the search becomes more and more directed toward better solutions.

I.e., let T_i denote the current solution at step i and let $z(T_i)$ denote the goodness of T_i (e.g., $-PS(T)$). In hill climbing, a move to T_{i+1} is acceptable, if $z(T_{i+1}) \geq z(T_i)$. In simulated annealing, *any* new solution is accepted with a certain probability:

$$Prob(\text{accepting solution } T_{i+1})$$

$$= \begin{cases} 1 & \text{if } z(T_{i+1}) \geq z(T_i) \\ e^{-t_i(z(T_{i+1})-z(T_i))} & \text{otherwise,} \end{cases}$$

where t_i is called the *temperature* and decreases over time.

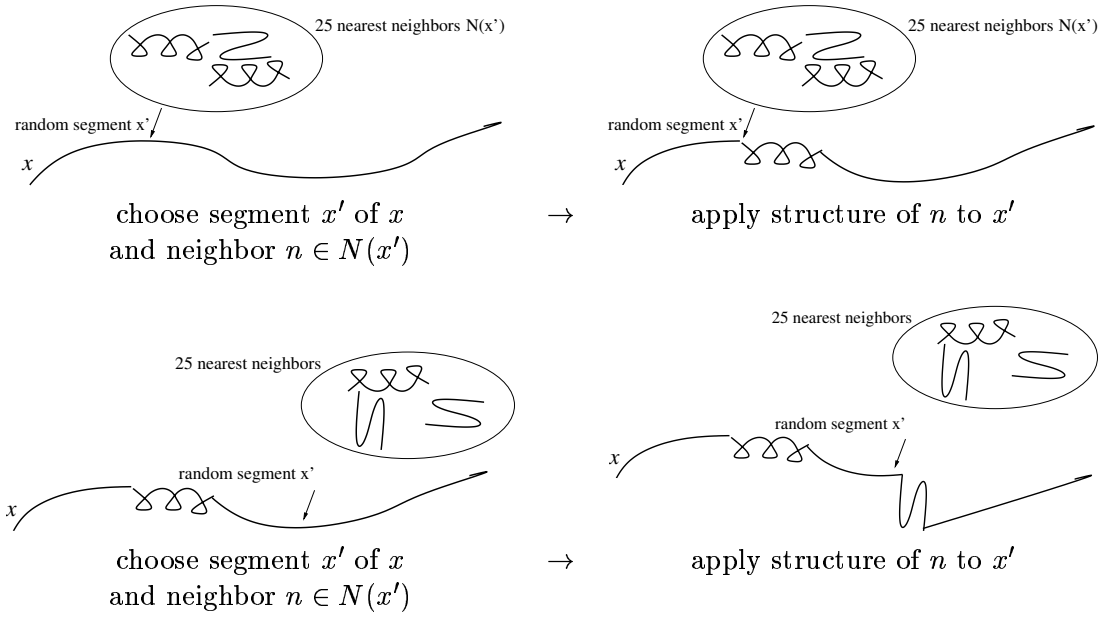
8.18 The ROSETTA algorithm

Structures are generated using a simulated annealing algorithm, employing a temperature that decreases linearly from 2500 to 10 over 10000 iterations (i.e. attempted moves).

Given a protein sequence x. The starting configuration is a fully extended chain. Repeat the following *move* 10000 times:

1. Randomly choose a nine-residue segment x' of x.
2. Randomly choose a nearest neighbor sequence $y' \in N(x')$.
3. Replace the torsion angles associated with x' by the ones associated with y'.
4. If the move puts two atoms closer than 2.5 Angstrom together, reject it.
5. If the move results in a score that does not meet the simulated annealing criterion, reject it.

Here is an illustration of the main move used in the algorithm:



8.19 Scoring

Obviously, the reliability of the results obtained using the ROSETTA algorithm is crucially dependent on the score function. We will discuss a simple form of it.

In probabilistic terms, we would like to maximize the probability of the predicted structure, given the sequence, which, using Bayes theorem, is:

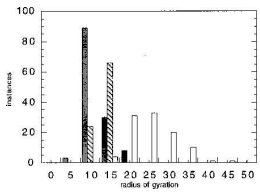
$$P(structure \mid sequence) = P(structure) \times \frac{P(sequence \mid structure)}{P(sequence)}.$$

In the comparison of different structures for the same sequence, $P(sequence)$ is constant and can be ignored.

In the context of threading, it is simplest to assume $P(structure) = \frac{1}{(\text{number of structures})}$.

In the context of de novo folding, not all generated structures are equally likely to be proteins, for example, conformations with unpaired β strands are quite unlikely. The term $P(structure)$ could be used to capture all features that distinguish protein structures from random chain configurations.

A simple choice is the following: set $P(structure) = 0$, for configurations in which atoms overlap. Otherwise, set $P(structure) = e^{-\kappa^2}$, where κ is the *radius of gyration* of the structure, which is small for compact structures.



(Simons et al. (1997))

Here we see the radius of gyration of obtained for structures generated using no scoring function (open bars), or κ^2 (hatched bars), and, for comparison, the values from structures selected from PDB (solid bars).

Now, let us consider the term $P(sequence \mid structure)$. In a very simple model, the conditional probability of seeing a particular amino acid a at a particular position i in a sequence will depend on the *environment* $E(i)$ of the position in the structure. In this case:

$$P(sequence \mid structure) \approx \prod_i P(a_i \mid E_i).$$

Another approach assumes the independence of pairs rather than individual amino acids:

$$P(sequence \mid structure) \approx \prod_{i<j} P(a_i, a_j \mid r_{ij}),$$

where r_{ij} denotes the distance between residues i and j. Applying Bayes we get:

$$P(a_i, a_j \mid r_{ij}) = P(a_i, a_j) \times \frac{P(r_{ij} \mid a_i, a_j)}{P(r_{ij})}.$$

The first factor in the previous term is independent of structure. Putting these results together, we get:

$$P(structure \mid sequence) \approx e^{-\kappa^2} \times \prod_{i<j} \frac{P(r_{ij} \mid a_i, a_j)}{P(r_{ij})},$$

and a *scoring function* is given by $-\log$ of this expression.

A more detailed scoring function can be based on the following equation:

$$P(a_1, a_2, \ldots, a_n \mid structure) \approx$$

$$\prod_i P(a_i, \mid E_i) \times \frac{P(a_i, a_j \mid r_{ij}, E_i, E_j)}{P(a_i, \mid r_{ij}, E_i, E_j) P(a_j \mid r_{ij}, E_i, E_i)},$$

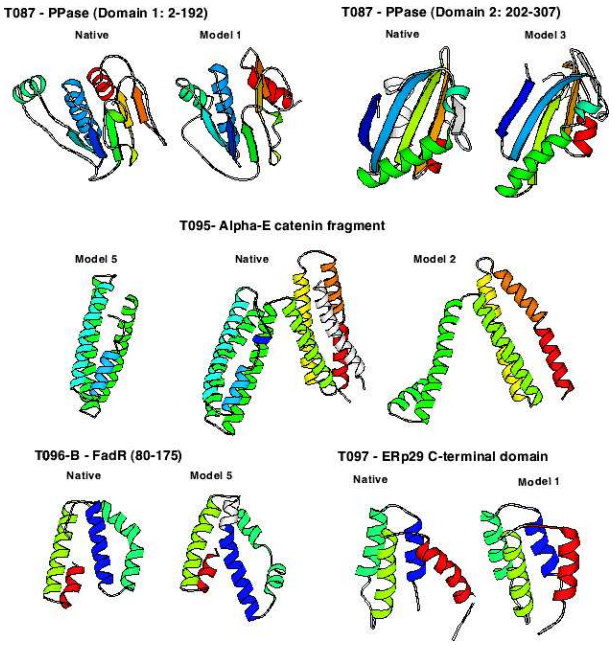
where E_i can represent a variety of features of the local structural environment around residue i. There are many more details that could be mentioned here, but we will skip them.

8.20 Performance

Every couple of years, a competition takes place called CASP (critical assessment of techniques for protein structure prediction, see `http://predictioncenter.llnl.gov/` for details) in which protein sequences are provided as target sequences for protein folding algorithms. These are sequences for which the three-dimensional structure has already been

determined experimentally, but has not yet been published. Many groups working on protein folding submit models for the target sequences.

In the latest such competition, CASP4 in 2000, the ROSETTA method performed very well: Large segments were correctly predicted (more than 50 residues superimposed within an rmsd of 6.5 Angstrom) for 16 of 21 domains under 300 residues for which models were submitted. Models with the global fold largely correct were produced for several targets with new folds, and for several difficult fold recognition targets, the Rosetta models were more accurate than those produced with traditional recognition models.



(R. Bonneau, J. Tsai, I. Ruczinski, D. Chivian, C. Rohl, C.E.M. Strauss and D. Baker, Rosetta in CASP4: Progress in *ab initio* protein structure prediction, manuscript.)

8.21 Classification of protein structures

There are a number of databases that classify protein structures, each based on slightly different concepts. These include:

- SCOP: Structural Classification Of Proteins (`http://scop.mrc-lmb.cam.ac.uk/scop/`),
- CATH: Class, Architecture, Topology and Homologous superfamily (`http://www.biochem.ucl.ac.uk/bsm/cath_new/index.html`),
- FSSP: Fold Classification based on Structure-Structure alignment of Proteins (`http://www2.ebi.ac.uk/dali/fssp/`), and
- DALI domain dictionary. (`http://www2.ebi.ac.uk/dali/domain/`).

8.22 SCOP

This description of the SCOP database closely follows the original paper:

- A.G. Murzin, S.E. Brenner, T. Hubbard and C. Chothia, *SCOP: A structural classification of proteins database for the investigation of sequences and structures*, J. Mol. Biol. 247:536-540 (1995).

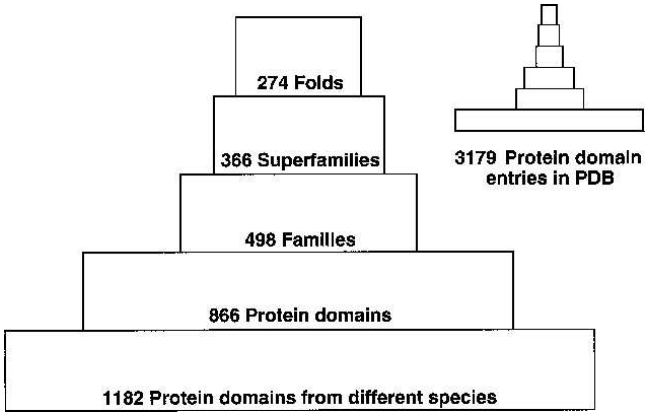
The goal of the SCOP database is to provide a detailed and comprehensive description of the *structural* and *evolutionary* relationships of the proteins of known structure. It also provides for each entry links to coordinates, images of the structure, interactive viewers, sequence data and literature references.

The *homology search* permits users to enter a sequence and obtain a list of any structures to which it has significant levels of sequence similarity.

The *key word search* finds matches from both the text of the SCOP database and the headers of Brookhaven Protein Database structure (PDB) files.

In SCOP, the unit of classification is usually the protein domain. Small and most medium size proteins are treated as a whole. The domains in large proteins are usually treated separately.

When SCOP was introduced in 1995, PDB contained 3179 domains. The SCOP classification of proteins is hierarchical:



(Source: Murzin *et al.* (1995).)

In the following we list the different units of classification.

Family: Proteins are clustered together into families if they appear to have a common evolutionary origin, i.e. if they have residues identities of 30% or more, or they have lower identities, but their functions and structures are very similar, e.g. globins with sequence identities of 15%. Families are subclassed by species.

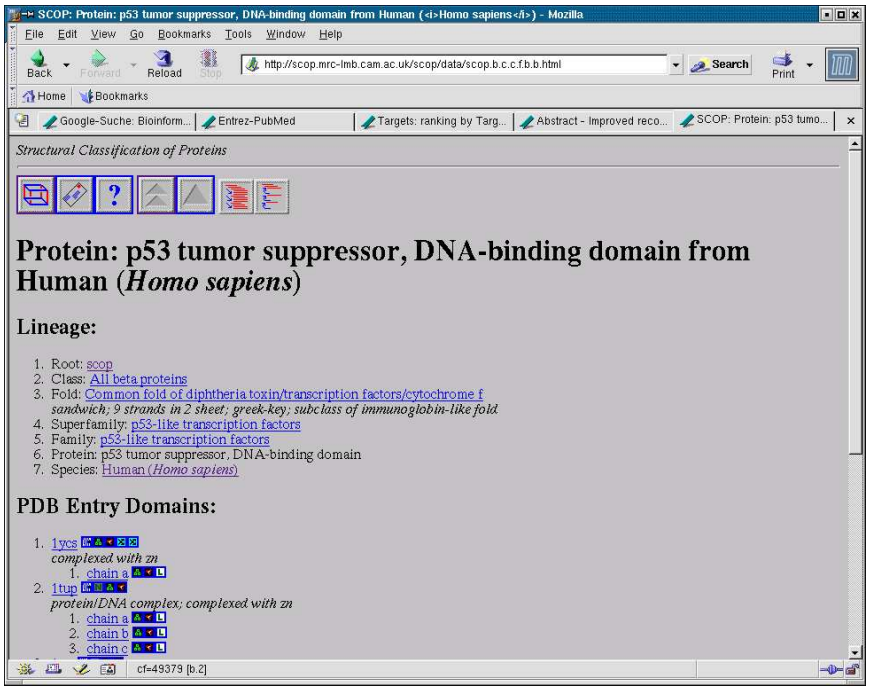
Superfamily: Families, whose proteins have low sequence identities, but whose structures and functional features suggest a common evolutionary origin, are placed together in superfamilies.

Common fold: Superfamilies and families are defined as having a common fold if their

proteins have the same major secondary structures in the same arrangement with the same topological connections.

Class: For the convenience of users, the different folds are currently grouped into 11 classes:

1. All alpha proteins (151).
2. All beta proteins (111).
3. Alpha and beta proteins (α/β) (117). Mainly parallel beta sheets (β-α-β units).
4. Alpha and beta proteins ($\alpha + \beta$) (212). Mainly anti-parallel beta sheets (segregated alpha and beta regions).
5. Multi-domain proteins (α and β) (39). Folds consisting of two or more domains belonging to different classes.
6. Membrane and cell surface proteins and peptides (12). Does not include proteins in the immune system.
7. Small proteins (59). Usually dominated by metal ligand, heme, and/or disulfide bridges.
8. Coiled coil proteins (5). Not a true class.
9. Low resolution protein structures (17). Not a true class.
10. Peptides (95). Peptides and fragments. Not a true class.
11. Designed proteins (36). Experimental structures of proteins with essentially non-natural sequences. Not a true class.



8.23 CATH

CATH is a hierarchical classification of protein domain structures, which clusters proteins at four major levels, defined on the CATH website as follows:

- **Homologous superfamily:** This groups together protein domains which are thought to *share a common ancestor* and can therefore be described as homologous. Similarities are identified first by sequence comparisons and subsequently by structure comparison using SSAP. Structures are clustered into the same homologous superfamily if they satisfy one of the following criteria:
  - Sequence identity $\geq 35\%$, 60% of larger structure equivalent to smaller,
  - SSAP score ≥ 80.0 and sequence identity $\geq 20\%$, 60% of larger structure equivalent to smaller,
  - SSAP score ≥ 80.0, 60% of larger structure equivalent to smaller, and domains which have related functions.

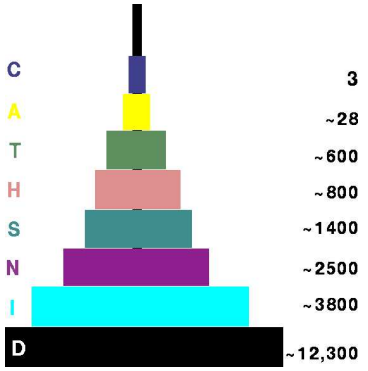
- **Topology** or fold family: Structures are grouped into fold families depending on both the *overall shape and connectivity of the secondary structures.* This is done using the structure comparison algorithm SSAP. Parameters for clustering domains into the same fold family have been determined by empirical trials throughout the database. Structures which have a SSAP score of 70 and where at least 60% of the larger protein matches the smaller protein are assigned to the fold family. Some large families are subclassed using a higher score threshold.

- **Architecture** describes the *overall shape of the domain structure* as determined by the orientations of the secondary structures but ignores the connectivity between the secondary structures. It is currently assigned manually using a simple description of the secondary structure arrangement e.g. barrel or 3-layer sandwich.

- **Class** is determined according to the *secondary structure composition* and packing within the structure. It can be assigned automatically for over 90% of the known structures. For the remainder, manual inspection is used and where necessary information from the literature taken into account. Three major classes are recognized;
  - mainly-alpha,
  - mainly-beta and
  - alpha-beta.

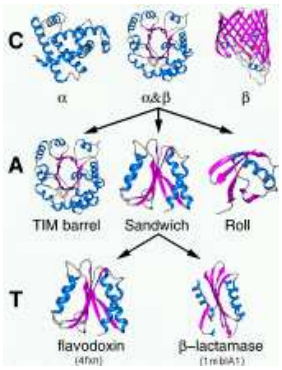
  The last class (alpha-beta) includes both alternating α/β structures and $\alpha + \beta$ structures, as originally defined by Levitt and Chothia (1976). A fourth class is also identified which contains protein domains which have low secondary structure content.

The following pyramid plot shows the number of groups identified at each level in the CATH database. Characters on the lefthand-side gives the CATH levels: Class; Architecture, Topology; Homologous superfamily; Sequence family - 35% sequence identity; Near-identical - 95% sequence identity; Identical - 100% sequence identity; Domain entry:



(http://wserv1.dl.ac.uk/CCP/CCP11/newsletter/vol2_3/orengo/)

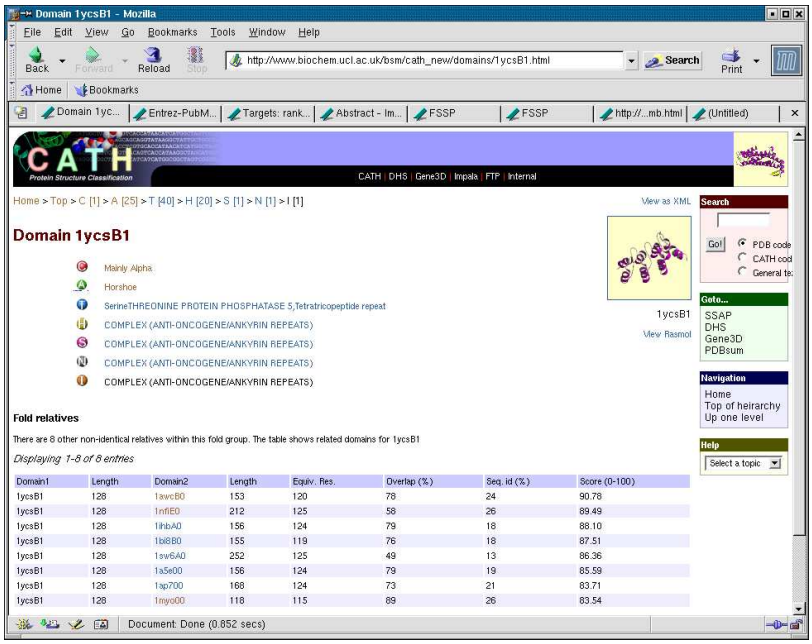
This figure illustrates the hierarchical nature of CATH:



(http://www.biochem.ucl.ac.uk/bsm/cath_new/cath_info.html)

References:

- Orengo, C.A., Michie, A.D., Jones, S., Jones, D.T., Swindells, M.B., and Thornton, J.M. *CATH- A Hierarchic Classification of Protein Domain Structures.* Structure. 5(8):1093-1108 (1997).
- Pearl, F.M.G, Lee, D., Bray, J.E, Sillitoe, I., Todd, A.E., Harrison, A.P., Thornton, J.M. and Orengo, C.A. (2000) Assigning genomic sequences to CATH Nucleic Acids Research. 28(1):277-282 (2000).



8.24 The FSSP database

The FSSP database includes all protein chains from the Protein Data Bank which are longer than 30 residues. The chains are divided into a *representative set* and *sequence homologs* of structures in the representative set.

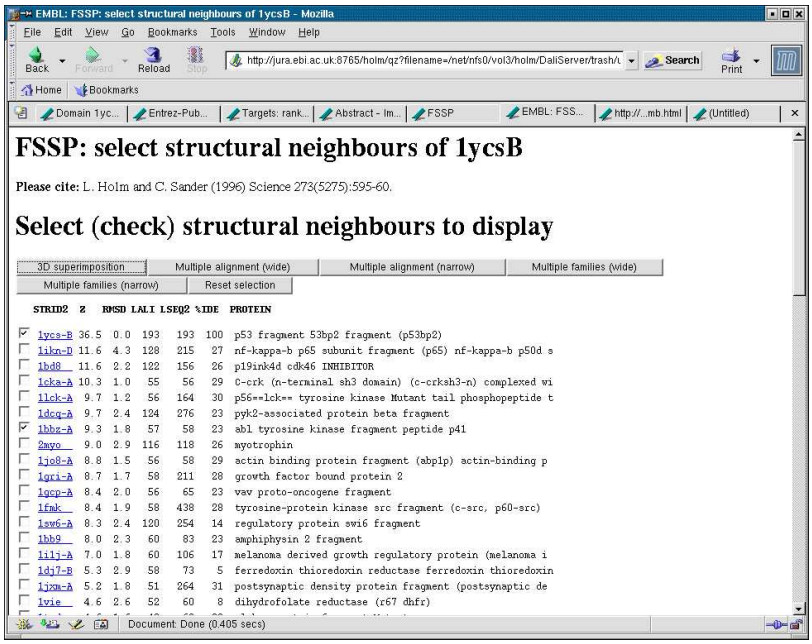
Sequence homologs have more than 25% sequence identity, and the representative set contains no pair of such sequence homologs. An all-against-all structure comparison is performed on the representative set.

The resulting alignments are reported in the FSSP entries for individual chains.

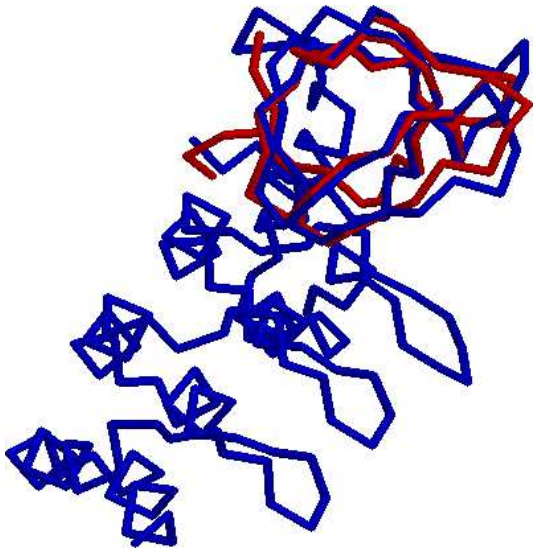
In addition, FSSP entries include the structure alignments of the search structure with its sequence homologs.

Reference:

- L. Holm and C. Sander, *Mapping the protein universe*, Science 273:595-602 (1996).



The structural alignment obtained from FSSP between 1ycs-B and 1bbz-A:



8.25 The DALI domain dictionary

In the "Dali domain dictionary", structural domains are delineated automatically. Each domain is assigned a *domain classification number* $DC\_l\_m\_n\_p$ representing

- a fold space attractor region (l),
- a globular folding topology (m),
- a functional family (n) and

- a sequence family (p).

Based on the Dali website, the levels of the classification can be described as follows:

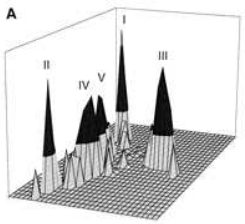
- **Sequence families:** The finest level of classification is a representative subset of the Protein Data Bank extracted using a 25% sequence identity threshold. All-against-all structure comparison are carried out within the set of representatives. Homologs are only shown aligned to their representative.

- **Functional families:** The next level of classification infers *plausible evolutionary relationships* from strong structural similarities which are accompanied by functional or sequence similarities. Functional families are branches of the fold dendrogram where all pairs have a high average neural network prediction for being homologous. The neural network weighs evidence coming from: overlapping sequence neighbors as detected by PSI-Blast, clusters of identically conserved functional residues, E.C. numbers, Swissprot keywords. The threshold for functional family unification was chosen empirically and is conservative; in some cases the automatic system finds insufficient numerical evidence to unify domains which are believed to be homologous by human experts.

- **Fold type:** The next level of the classification is fold type. Fold types are defined as clusters of *structural neighbors in fold space* with average pairwise Z-scores (by Dali) above 2. The threshold has been chosen empirically and groups together structures which have topological similarity. Higher Z-scores corresponds to structures which agree more closely in architectural detail.

- **A fold space attractor region:** The highest level of the fold classification corresponds to secondary structure composition and super-secondary structural motifs. Five attractor regions in fold space have been identified. The fold space is partitioned so that each domain is assigned to one of attractors 1-5, which are represented by archetype structures, using a shortest-path criterion.

Density distribution of domains in fold space and attractors:

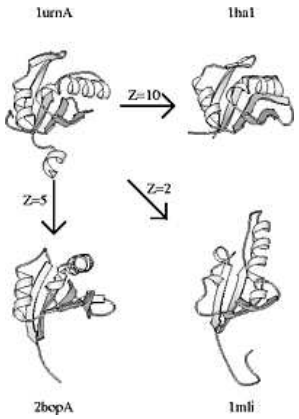


Five attractor regions in fold space have been identified, represented by archetype structures:

I	II	III	IV	V
α/β	all-β	all-α	antiparallel β-barrels	α-β meander

(Both figures: http://www.ebi.ac.uk/dali/domain/3.1beta/Help.html)

The fold types are defined via Z-scores. In the following example, some structural neighbors of 1urnA (top left) are displayed. Note that 1mli (bottom right) has the same fold type as 1urnA, even though there are shifts in the relative orientation of secondary structure elements.



(http://www.ebi.ac.uk/dali/domain/3.1beta/Help.html)

Additional to the five attractor regions in fold space, two further classes are defined. Domains which are not clearly closer to one attractor than another, are assigned to the mixed class 6. Currently, class 6 comprises about one sixth of the representative domain set. In the future, some of these may be assigned to emerging new attractors. Structures, which are disconnected from other structures (no connected path to any attractor), are assigned to class 7.

References:

- L. Holm, L. and C. Sander. *Dictionary of recurrent domains in protein structures*, Proteins, 33:88-96 (1998).
- L. Holm and C. Sander. *Mapping the protein universe*, Science, 273:595-603 (1996).

8.26 Summary of discussed classifications

SCOP	CATH	Dali dictionary
family common evolutionary origin	**sequence family** common evolutionary origin	**sequence family** common evolutionary origin
super family common evolutionary origin	**hom. super family** common evolutionary origin	**functional family** plausible common evolutionary origin
common fold same major secondary structure arrangement	**topology** same shape and connectivity of secondary structure	**fold type** structural neighbors in fold space
	architecture overall shape of domain structure	
class 11 classes of folds	**class** 3 classes of secondary structure composition and packing	**attractor region** 5 attractors in fold space, plus 2 additional classes

8.27 Structure comparison

Given two three-dimensional protein structures. There are a number of reasons why we would like to compare them, e.g.:

- for classification purposes, i.e. to determine whether they probably have a common evolutionary origin and thus belong in the same family etc.,
- to predict the function of one protein by determining structurally similar proteins of known function, or
- if one is the true structure and one is a predicted one, to measure how good the prediction is, e.g. in the CASP contests.

A number of different approaches exist, e.g. for alignments without indels, one can use rigid body superposition and distance map similarity, whereas in the general case, e.g. one can use double dynamic programming and contact map overlap.

8.28 Rigid body superposition

Given two structures x and y, each represented by a linear list $x = (x_1, x_2, \ldots, x_m)$ and $y = (y_1, y_2, \ldots, y_m)$, respectively, of three-dimensional coordinates, with the understanding that x_i is matched to y_i, for all i.

Naively, one might consider the following measure:

$$\sqrt{\frac{1}{m}\sum_{i=1}^{m}(x_i - y_i)^2}.$$

In practice, this is not useful, because it depends on the absolute locations of the two structures.

Let T denote any orientation- and distance preserving motion. The *root squared mean distance* (RSMD) between x and y is defined as:

$$RSMD(x, y) = \sqrt{\frac{1}{m}\min_T \sum_{i=1}^{m}(x_i - Ty_i)^2}.$$

For $RSMD$, we need to find the transformation T that minimizes the expression above. This is usually done in two steps: first, translate the center of mass of x and y to the origin. Then find the best rotation, e.g. by doing a step-wise search of rotational space.

8.29 Distance map comparison

Alternatively, if we only compare the *distances between pairs* of positions in x and y, then the resulting value will be independent of the absolute coordinates of the two structures:

$$RSMD_d(x, y) = \frac{1}{m}\sqrt{\min_T \sum_{i=1}^{m}\sum_{j=1}^{m}(d(x_i, x_j) - d(y_i, y_j))^2},$$

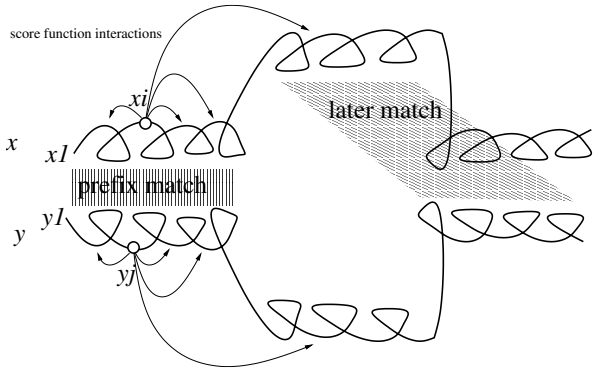
where $d(x_i, x_j)$ denotes the distance between the two points x_i and x_j.

There is a close relationship between $RSMD$ and $RSMD_d$, however $RSMD_d$ cannot distinguish between mirror images.

8.30 Structure alignment

Two *sequences* x and y can be aligned using dynamic programming, because the optimality of the alignment of two prefixes of x and y *does not depend* on how later positions of the two sequences are aligned.

This kind of independence generally does not hold for the alignment of structures, because the employed scoring functions usually take the matching of neighboring positions into account:



8.31 Double dynamic programming

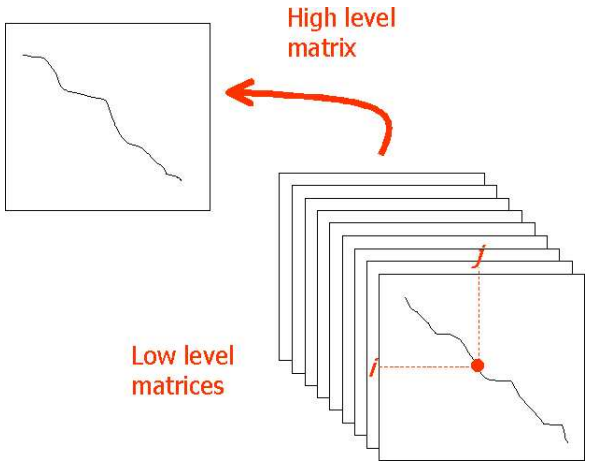
Double dynamic programming is a heuristic that attempts to extend the application of dynamic programming to the problem of aligning protein structures. It is the basis of the SSAP program, see C.O. Orengo and W.R. Taylor (1996) for details.

The following description of DDP is based on:
`http://www.ii.uib.no/~inge/talks/ismb-tutorial/`.

Given two folded sequences $x = (x_1, \ldots, x_m)$ and $y = (y_1, \ldots, y_n)$. The main idea in double dynamic programming is to perform *two levels* of dynamic programming:

- Ordinary dynamic programming is performed on a *high level* scoring matrix R to obtain the final alignment.
- Each cell (i, j) of R should contain a value that expresses how likely it is that the pair (x_i, y_j) will be contained in an optimal alignment.
- For each cell (i, j), this value is determined by performing a *low level* dynamic program on a matrix ${}^{ij}R$, under the constraint that (x_i, y_j) must be part of the low-level alignment.
- The scores along each of these low-level alignments are accumulated in the high level scoring matrix R.

The high-level DP matrix R is obtained by computing a low-level matrix ${}^{ij}R$ for each cell (i, j) in R and then accumulating the values in R:



(http://www.ii.uib.no/~inge/talks/ismb-tutorial/)

8.32 The low-level matrix

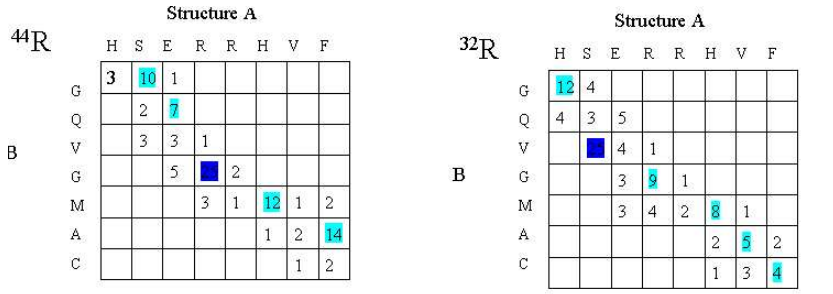
For each cell (i, j) in the high-level matrix R, we compute a matrix ${}^{ij}R$ such that ${}^{ij}R_{kl}$ is a score obtained for aligning x_k to y_l, under the constraint that x_i is aligned to y_j, using dynamic programming.

Then, given such a low-level matrix ${}^{ij}R$: For each cell (k, l) on the optimal path in ${}^{ij}R$ through (i, j), we increase the value of R_{kl} by ${}^{ij}R_{kl}$, if ${}^{ij}R_{kl}$ exceeds a pregiven threshold.

The overall aim is to give high scores to cells in R that occur in optimal paths for many of the low-level alignments.

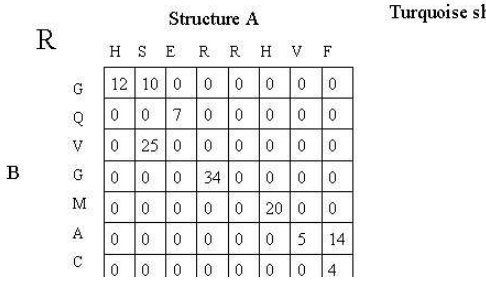
Running a dynamic program using the values in the high-level matrix R produces the final alignment.

Given two folded sequences A and B. Here we see how the optimal values for ${}^{44}R$ and ${}^{32}R$



Turquoise shows the best alignments

are added to the R matrix:



(http://www.ii.uib.no/~inge/talks/ismb-tutorial/)

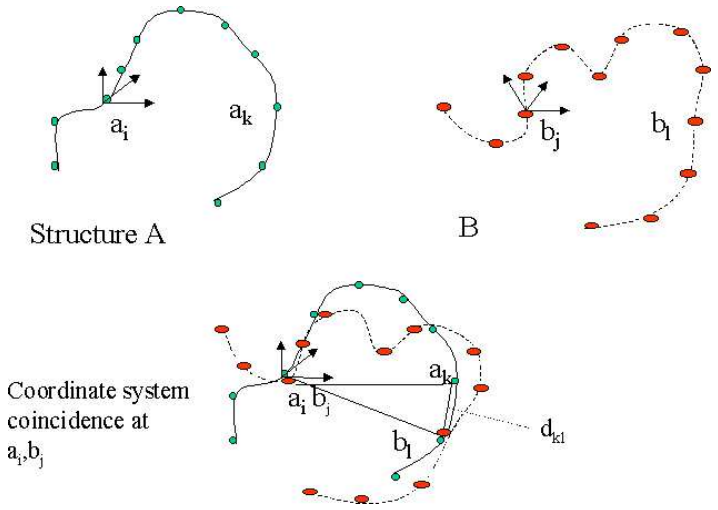
8.33 Scoring the low-level matrix

The value ${}^{ij}R_{kl}$ should capture how well x_k and y_l match up in three-dimensional space, under the assumption that x_i and y_j are superimposed upon each other.

To measure this,

- define two local reference systems for x_i and y_j, e.g. by using the C_α atoms of x_{i-1}, x_i, x_{i+1} and of y_{j-1}, y_j, y_{j+1}, respectively,
- transform the coordinates of the residues of x and y into the respective coordinate systems, and then
- compute a score for x_k and y_l based on the transformed coordinates. For example, simply use the distance between the two residues.

For sequences A and B, the simplest scoring scheme for scoring the low-level matrix ${}^{ij}R$ is obtained by superimposing residue a_i onto b_j and then setting ${}^{ij}R_{kl}$ using the distance between residues a_k and b_l:



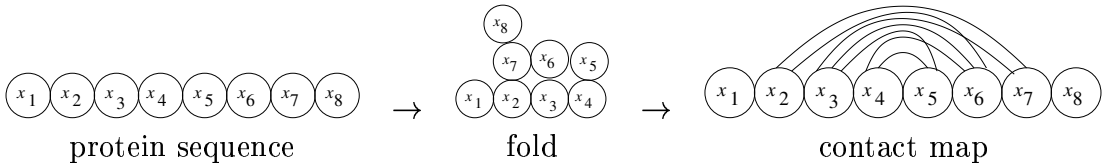
(http://www.ii.uib.no/~inge/talks/ismb-tutorial/)

8.34 Contact map overlap

Given a protein sequence $x = (x_1, \ldots, x_m)$ and assume we are given coordinates for each residue (e.g., for their C_β atoms).

We say that two residues x_i and x_j are *neighbors*, or are *in contact*, if the distance between the two corresponding C_β atoms is less than some given threshold, e.g. 7 Angstrom.

Here is an illustration:



The contact map of a folded protein is represented as a $0-1$, symmetric $m \times m$ matrix M, with $M_{ij} = 1 \Leftrightarrow x_i$ and x_j are neighbors.

It is convenient to think of M as the adjacency matrix of a graph $G = (V, E)$ with node set $V = \{1, 2, \ldots, m\}$.

8.35 The return of the alignment graph...

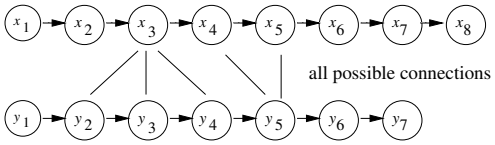
Given two folded sequences x and y. Loosely speaking, the goal is to find an alignment of the two sequences that preserves as much of the contact map as possible. The idea here is that similar sites in a structure will make similar contacts.

Recall the definition of an *alignment graph* (Chapter 3):

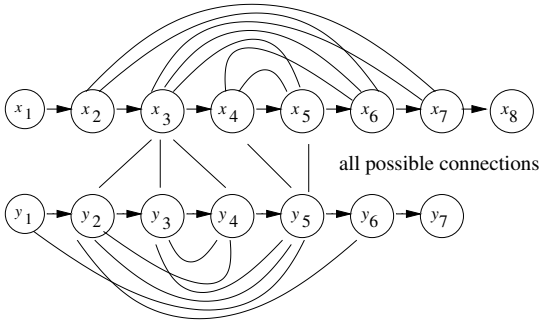
Given two sequences $x = (x_1, \ldots, x_m)$ and $y = (y_1, \ldots, y_n)$. The *alignment graph* $G = (V, E)$ is the complete bipartite graph on nodes $V = \{v_1, \ldots, v_m\} \cup \{w_1, \ldots, w_n\}$ such that v_i

represents residue x_i and w_j represents residue y_j. Moreover, every node in the first subset is connected to every node in the second subset, and vice versa. We obtain the *extended* alignment graph by adding a set H of directed edges (v_i, v_{i+1}) and (w_j, w_{j+1}) for all $i = 1, \ldots, m-1$ and $j = 1, \ldots, n-1$.

Here is the extended alignment graph $G = (V, E, H)$ for two sequences x and y:



Then we add the set I of given contact edges for both sequences to obtain a *structural alignment* graph $G = (V, E, H, I)$:



8.36 Structural trace

Note that this graph is very similar to the one defined for the problem of RNA secondary structure alignment, where interactions were defined via base-pairing. In that problem, we formulated an integer linear program (ILP) for computing an optimal weight alignment subject to the constraint that any one node can participate in at most one interaction.

Given the structural alignment graph $G = (V, E, H, I)$ for two folded sequences x and y. Let $T \subseteq H$ be a trace, i.e. a non-crossing subset of H. Two contact edges p and q (that do not belong to the same sequence) are called *matched*, if the two alignment edges e_l and e_r joining the two left respectively right nodes of p and q are contained in the trace T.

Let B be the set of all matched contacted edges. We call (T, B) a *structural trace*.

8.37 Scoring a structural trace

Let (T, B) be a structural trace. As for conventional traces, each edge $e \in T$ is given a weight $\omega(e)$ which is simply the score for aligning the two corresponding symbols.

Any contact match $\{i_p, i_q, e_l, e_r\}$ is completely specified by the two edges e_l and e_r and so we can denote it by m_{lr} and assign a weight $\omega(l, r)$ to it. Let M denote the set of all interaction

matches:

$$M = \{m_{lr} \mid m_{lr} = \{i_p, i_q, e_l, e_r\} \text{ is an interaction match in } G\}.$$

We define the *score of a structural trace* as:

$$S((T,B)) = \sum_{e \in T} \omega(e) + \sum_{\substack{i_p, i_q \in B, e_l, e_r \in T \\ \{i_p, i_q, e_l, e_r\} \in M}} \omega(l,r).$$

We can find the maximal scoring trace by solving an ILP.

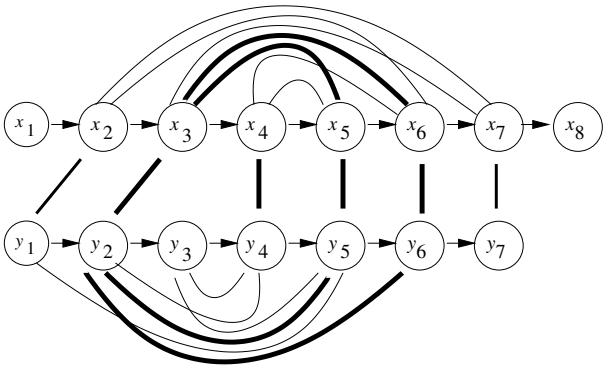
8.38 Maximal scoring structural trace

Given a structural alignment graph $G = (V, E, H, I)$, the score ω_i for realizing an edge $e_i \in E$ and the score ω_{lr} for realizing a contact match m_{lr}.

By dropping the constraint that any given node can interact with at most one other node, from the ILP formulated for RNA structure alignment we obtain an ILP for maximizing the score:

$$\begin{array}{lll} \text{Maximize} & \sum_{e_i \in E} \omega_i x_i + \sum_{m_{ij} \in M} \omega_{ij} x_{ij}, & \\ \text{subject to} & \sum_{e_i \in C \cap E} x_i \leq |C \cap E| - 1, & \text{for all critical} \\ & & \text{mixed cycles } C, \\ & x_{ij} \leq x_i \text{ and } x_{ij} \leq x_j, & \text{for all variables, and} \\ & x_i, x_{ij} \in \{0,1\} & \text{for all variables.} \end{array}$$

In this example, two matches are realized:



By appropriate application of the "Lagrangian relaxation method", such structural ILPs can be solved very efficiently. For details, see:

- Giuseppe Lancia, *Mathematical programming approachs for computational biology problems*, to appear in: Scuola CIRO.