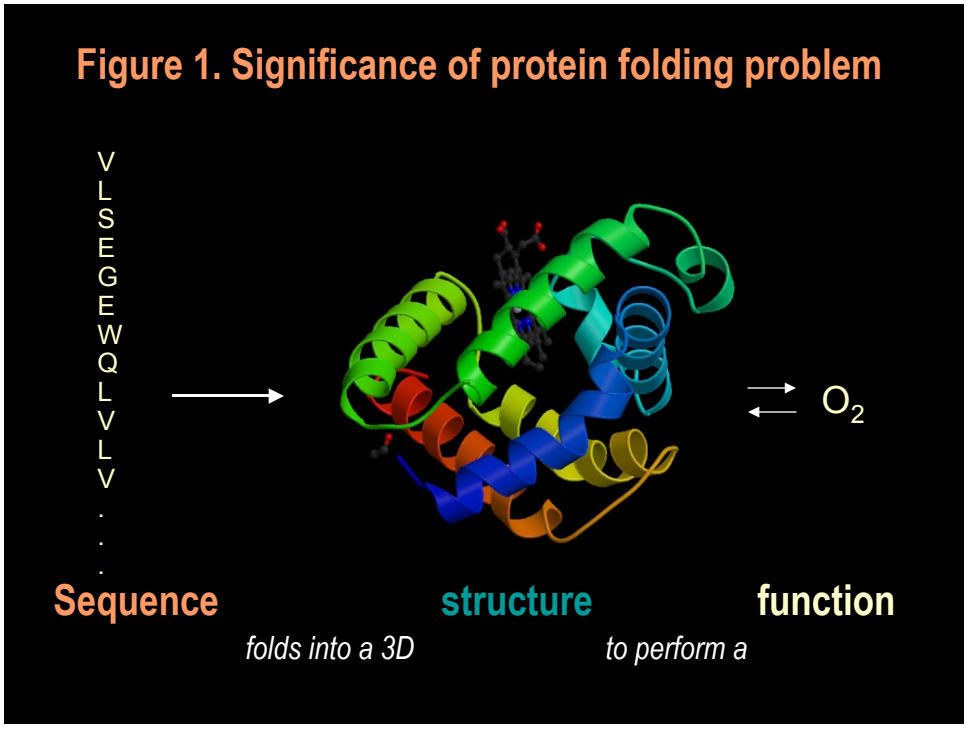
A. Porollo, R. Adamczak, M. Wagner and **J. Meller**; *Maximum Feasibility Approach for Consensus Classifiers: Applications to Protein Structure Prediction*, Proceedings of The Second International Conference on Computational Intelligence, Robotics and Autonomous Systems, CIRAS 2003

Part IV contains several appendices, including abbreviations and definitions regarding basic notions discussed in this dissertation, list of other publications by the author of this thesis, selected materials highlighting the context and importance of some of the findings presented here and a brief description of the LOOPP software package.

I.2. The Protein Folding Problem

The recent unveiling of the human genome marked the transition in the biological sciences towards the post-genomic era, in which the understanding of protein structure and function becomes a crucial extension of the sequencing efforts. Despite recent progress in high throughput techniques, the experimental determination of protein structure by using X-ray crystallography or NMR spectroscopy [Branden and Tooze, 1991; van Holde et. el., 1998] remains a bottleneck in structural genomics. This poses a challenge and an opportunity for computational approaches to complement and facilitate experimental methods.

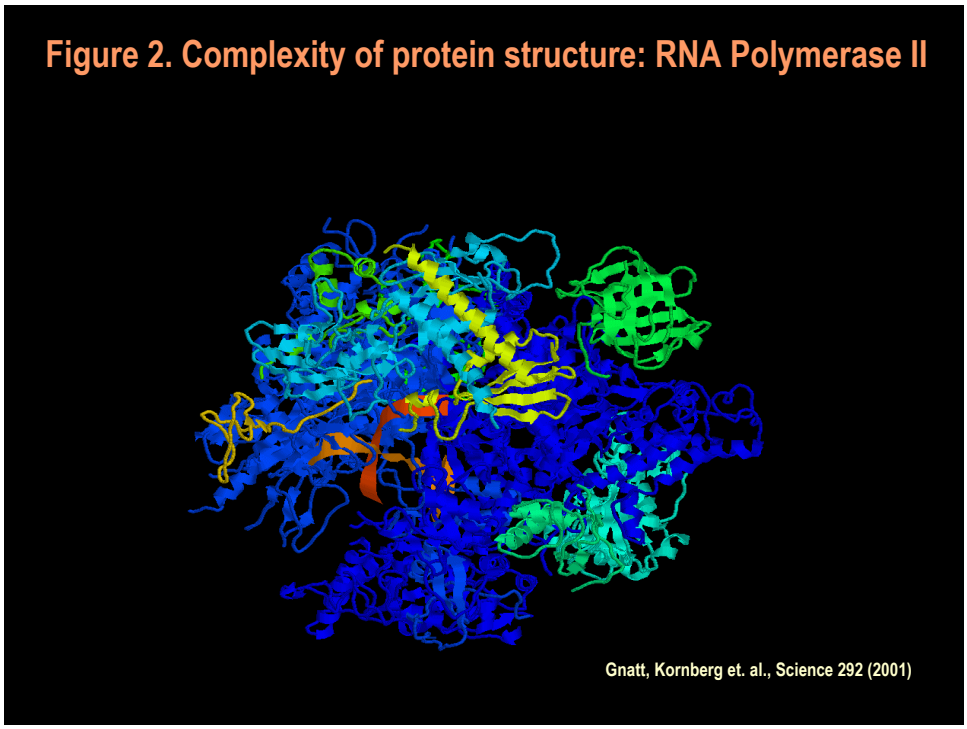


Proteins are linear polymers composed of a sequence of amino acid residues that are connected by peptide bonds (creating the protein "backbone"). Without accounting for several rare amino acids and numerous chemical modifications of the basic blocks, there are 20 different amino acids that are characterized by chemically unique side chains (containing

from one to approximately 20 atoms) that hang off the backbone chain. Protein molecules consist of several tens to several thousands of amino acids and thus between a few hundred and tens of thousands of atoms [Branden and Tooze, 1991].

Proteins typically adopt a "unique" three-dimensional structure, meaning that under physiological conditions proteins with identical or nearly identical sequences would adopt similar backbone conformations that form a well-defined cluster (called protein family), which is different from "structures" (clusters) of other families [Murzin et. al., 1995; Bateman et. al., 2002]. On the other hand, however, there are many examples of inherently unstructured or unstable proteins that may adopt very different conformations, for example depending on specific interactions with other proteins.

The above paragraph contains several somewhat fuzzy qualifiers – something that quantitatively trained readers may find difficult to accept. We will try to explain some of the above statements in the remaining part of this section. Yet, it is important to realize that the nature of biological objects and processes we are dealing with is such that approximate and inherently arbitrary distinctions need to be made when designing mathematical models representing the underlying biology. Here, for the sake of simplicity we will only consider proteins that under specific conditions fold into well-defined stable structures.



The *protein folding* problem consists of predicting the three-dimensional structure of a protein from its amino acid sequence (see Figure 1). The methodology and modeling aspects of protein folding have been vastly discussed in the literature. For excellent and up-to-date surveys of methods as well as their limitations, the reader is referred to [Schonbrun et. al., 2002; Banavar et. al., 2001; Sternberg et. al., 1999]. In what follows, we briefly discuss several central concepts and ideas that underlie developments in the field.

The overall three-dimensional structure (conformation) of a protein may be hierarchically described first in terms of the backbone conformation, with locally ordered elements of secondary structure, such as alpha helices and beta strands, and then in terms of side chain conformations given the relatively rigid backbone conformation. Protein structures can be further classified according to their secondary structure content and the relative packing of the secondary structure elements into distinct structural classes called folds. At present, there are well over 20,000 known protein structures, which are deposited in the Protein Data Bank (PDB) [Berman et. al., 2000]. Depending on the classification criteria these structures are divided into several hundred to about one thousand distinct folds. A number of families can be distinguished for each fold, with a total number of about 6,000 distinct families according to the Protein Families (PFAM) database [Bateman et. al., 2002].

The computational protein structure prediction is a challenging problem. In order to appreciate the difficulty of the problem at hands it is useful to consider a brute force approach based on exhaustive enumerations of all possible conformations that may be adopted by a chain of amino acids. Each residue adds to the backbone two single bonds, which are free to rotate around their axis. However, due to steric constraints (clashes between backbone and side chains atoms), only up to three states (torsional angles) can be adopted around each single bond. Therefore, the number of possible backbone conformations is of the order of 9^N, where N is the number of amino acids in the chain [Branden and Tooze, 1991; van Holde et. el., 1998].

While this estimate is an upper bound, the conformational space to be explored becomes huge even for relatively short proteins, making a straightforward approach of exhaustive search impractical. Of course, even if we could perform an exhaustive search we would still face the problem of finding an appropriate scoring function capable of scoring the native-like structures higher than all the alternative conformations, which is far from trivial as discussed in the subsequent sections.

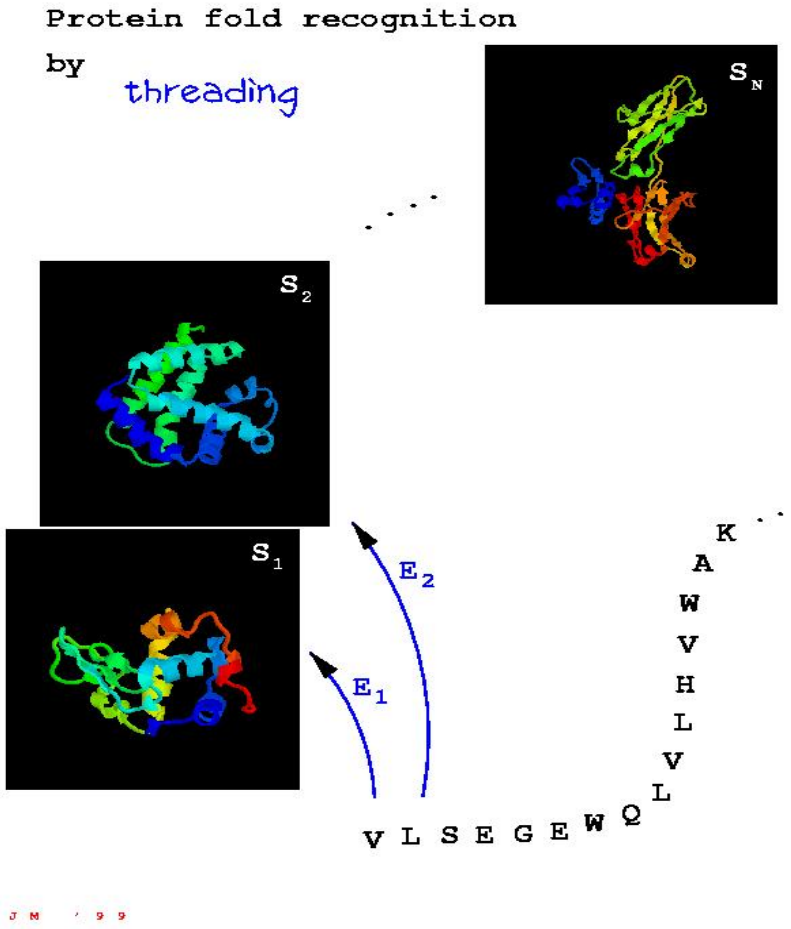
Except for extremely slow folders, proteins fold under physiological conditions on the time scales of milliseconds to seconds. Thus, the exponential scaling in the size of the conformational space remains in stark contrast with the observed folding rates, an observation known as the Levinthal's paradox [van Holde et. el., 1998]. Clearly, nature does not use a combinatorial approach in order to fold proteins. Consequently, using nature as a guideline may help in designing successful modeling and simulation protocols. In general, the existing computational approaches to protein folding problem may be roughly divided into two classes, based on the underlying principles and the extent of incorporating the physical characteristics of the protein folding process into computational protocols.

The *ab-initio* or *de novo* protein folding simulations attempt to reproduce (or at least to use as a guideline) the actual physical folding process. Such folding simulations are based on the thermodynamical hypothesis, first introduced by Anfinsen [Anfinsen, 1973], in which the unique three-dimensional structure of a protein is postulated to correspond to a global minimum of the free energy function. In folding simulations, the energy function to be minimized is usually postulated in the form of a conveniently chosen atomistic force field (or folding potential), with parameters fitted to reproduce experimental data, whereas the search for the native conformation entails the solution of a global optimization problem. Some methodological aspects of atomistic models of proteins and computer simulations using

Molecular Dynamics, Monte Carlo and other global optimization techniques are discussed in [Meller, 2003].

One might argue that knowing the complete atomistic description of the environment and the underlying physical interaction laws, we should be able to find the structure of a protein given the environment and interaction partners in particular. However, the problem is far from being trivial due to mentioned before size of the conformational space and the resulting sampling problem as well as inherent inaccuracy of atomistic force fields. Therefore, protocols that are in fact effective combinations of the *de novo* and knowledge-based approaches (see below), such as the *Rosetta* method by D. Baker and colleagues [Simons et. al. 1997], are more successful in practice.

Figure 3. Fold recognition and sequence-to-structure matching (threading)



The alternative *protein* (or *fold) recognition* approach [Bowie et. al., 1991; Jones et. al., 1992; Sippl et. al., 1992] relies on the fact that a significant fraction of protein structures (folds) have already been determined. The search for the overall structure is reduced in fold recognition methods to tests of sequence fitness into known and limited number of known folds (thus it cannot be applied to novel folds). In other words, the search for the native conformation is restricted to the set of known structures, as opposed to computationally

expensive search in the space of all possible conformations in case of *ab initio* folding simulations.

Since proteins of similar sequences usually fold into similar structures, sequence alignment (discussed in the next section) is the basic tool for assigning an unknown protein to a family of structurally and functionally characterized proteins. In many cases, sequence identity between 20 to 30% is sufficient to confidently assign a new protein to its family by using family profile based methods for sequence alignment, such as Position Specific Iterative Basic Local Alignment Tool (Psi-BLAST) algorithm [Altschul et. al., 1997] or profile Hidden Markov Models (HMMs) [Durbin et. al., 1998]. High degree of sequence similarity (also called sequence homology) allows one to obtain reliable alignments and effectively overlap new sequences with backbones of known structures. Furthermore, final three-dimensional models may be built by subsequent refinement of the alignment-based initial structure with atomistic force fields and global optimization, an approach known as *homology modeling*.

On the other hand, experiments found a limited set of folds compared to a large diversity of sequences. In other words, while sequence similarity usually implies significant structural similarity, the reverse is not true i.e. structural similarity does not necessarily imply sequence similarity. Because divergent or unrelated sequences may fold into similar structures, it suggests the use of structures to find remote similarities between proteins.

Threading is a fold recognition technique to directly match a sequence with a protein structure and a plausible function [Bowie et. al., 1991]. Protein recognition by sequence-to-structure matching or threading, allows one to find distant homologs that share the same fold without detectable sequence similarity (see for example [Meller and Elber 2001]). Given an appropriate scoring function, which can be thought of as a simplified folding potential, these methods find the "best" template from the library of known folds by evaluating directly sequence-to-structure fitness [Mirny and Shaknovich, 1998].

The scoring functions for threading (threading potentials) may incorporate different measures of sequence to structure fitness, such as compatibility between predicted and observed secondary structures or optimality of the effective inter-residue interactions imposed by overlaying a query sequence with a template structure. Such scoring functions should have a functional form that facilitates efficient computing of optimal alignments (with gaps) of a sequence into known protein structures, as discussed in the next two sections.

I.3. Sequence Alignment and Dynamic Programming

There is an enormous level of redundancy in biological systems [Gibson and Muse, 2002]. For instance, identical or very similar molecules and involving them processes are being used across different cells, tissues and species. On the other hand, it is important to recognize the limits to similarity (for example between analogous protein pathways in human and yeast) in order to identify the most significant (i.e. conserved) features. For these reasons, analogy and comparison between molecular objects and processes is an extremely powerful tool in biology.

Proteins and other important bio-molecules such as nucleic acids and poly-saccharides are linear (with some exceptions) polymers that can be represented as strings or

sequences in mathematical terms. For that reason (and in light of the remarks from the previous paragraph) string matching and sequence alignment algorithms play central role in bioinformatics as crucial tools of sequence analysis and comparison. For example, as discussed before, high degree of sequence similarity typically implies similar structure and function and, therefore, new proteins can be assigned to known protein families using sequence alignment tools.

In order to assess the level of similarity between two sequences one may utilize their optimal alignment. The problem of finding the optimal alignment of two sequences with gaps results in a global optimization problem that may be solved efficiently by the *Dynamic Programming* (DP) algorithm. DP is a classical computer science technique to solve combinatorial optimization problems [Gusfield, 1999], and plays an important role in computational biology [Durbin et. al., 1998].

A typical DP problem spawns a search space of potential solutions in a recursive fashion, from which the final answer is selected according to some criterion of optimality. If an optimal solution can be derived recursively from optimal solutions of subproblems, DP can evaluate a search space of exponential size in polynomial time and space as a function of the length of the sequences to be aligned, provided that a ("local") scoring function leading to piecewise decomposable problem is used [Durbin et. al., 1998]. In the following we will show how DP can be applied to the sequence and sequence-to-structure alignment problem, highlighting these aspects of DP that play an important role in designing effective threading potentials for sequence-to-structure matching.

Formally, the relatedness of two strings or sequences may be defined in terms of the *edit* distance defined as the minimal number of basic edit operations, including substitution, insertion and deletion, that are needed to transform one string into another [Gusfield, 1999]. Edit distance may be further generalized, allowing for character dependent weights (scores) of different substitutions. Alternatively, a notion of similarity between two sequences in terms of the score of their optimal alignment (which corresponds to their minimum weighted edit distance) may be introduced.

Let us consider two strings (or sequences in the dual formalism), $S_1 = a_1 a_2 \ldots a_n$ and $S_2 = b_1 b_2 \ldots b_n$, over certain alphabet A (for example consisting of twenty letters representing different amino acids, $A = \{\alpha_i\}_{i=1}^{20}$), $a_k, b_l \in A \quad \forall k,l$. We also consider an extended alphabet that contains the "space" or "dash" symbol, $\overline{A} = A \cup \{-\}$, representing "gaps" i.e. insertions of unknown ("missing" with respect to other sequences in the family) characters to one of the sequences or equivalently deletions of characters from the other sequence.

A *global alignment* of sequences S_1 and S_2, denoted here as $\Lambda(S_1, S_2)$, is obtained as a result of intercalating the two sequences such that a new sequence of length $n+m$ is obtained and the order of characters in each sequence is preserved. Such intercalated sequence may be conveniently displayed with one of the original sequences above the other so that every character or gap in either string is placed against a unique character or gap in the other sequence (with gap against gap alignments excluded).

As an example of conversion between the two representations of global alignments let us consider an intercalated sequence $a_1 b_1 a_2 a_3 b_2 b_3 a_4 b_4$, which corresponds to an alignment $\Lambda = a_1 b_1 a_2 - a_3 b_2 - b_3 a_4 b_4$ that can be represented in the alternative notation as:

$$\begin{matrix} a_1 a_2 a_3 - a_4 \\ b_1 - b_2 b_3 b_4 \end{matrix} . \qquad (1)$$

We would like to comment that *local alignments*, which are more appropriate when partial similarity (e.g. similarity between protein domains) is considered, are in fact displayed in Figure 3. As opposed to global alignments, only the subsequences that maximize the similarity in terms of the alignment score (defined in equation (2) below) are considered in case of local alignments.

We define a *scoring function* (also referred to as a scoring matrix), $f_s : \overline{A} \times \overline{A} \to R$, that assigns to each pair of characters a score for replacing (substituting) one character by the other, e.g. a score for amino acid substitution, $f_s(\alpha_i, \alpha_j)$. The total score of an alignment $\Lambda(S_1, S_2)$ of length l is defined as the sum of scores for pairs of characters that are aligned against each other, $f_s(x_i, x_{i+1})$:

$$f_{\text{tot}}(\Lambda(S_1, S_2)) = \sum_{i=1}^{l/2} f_s(x_i, x_{i+1}) \;\; ; \quad x_i, x_{i+1} \in \overline{A} \quad \forall i \, . \tag{2}$$

We assume here that the scores of individual pairs (substitutions) are not explicitly dependent on the alignment. In other words, the scores are *local* and do not change depending on what characters are aligned at other positions.

There is an extensive literature regarding the design of scoring matrices for sequence alignment (see for example [Henikoff and Henikoff, 1989; Durbin et. al., 1998]). Biologically meaningful alignments can only be obtained when suitable scoring schemes are used and different tasks may require different scoring matrices. One approach is to choose the scores based on the observed frequencies of amino acid substitutions between carefully selected representatives of known protein families.

An example of such derived scores are the BLOSUM scoring matrices, with the number indicating the level of evolutionary relatedness between the representatives included in the training set (for example BLOSUM50 denotes the scoring matrix derived from sequences sharing at least 50% of sequence identity). In addition to BLOSUM scoring matrices for 20 amino acids, one also needs to assign gap penalties. Here, for simplicity gap penalties are assumed to be proportional to the number of spaces that are inserted. More realistic models of gap penalties usually assume different cost of opening and extending a gapped region [Durbin et. al., 1998].

Figure 4. Sequence alignments reveal biological relatedness

```
i..d......1.......i..2.........3.....i.i...iii.....i.....5.   531 - 582
-FKLELVEKLFAEDTEAK-NPFSTQDTDLDLEMLAPY-I-PMD---DDLQL-RSFDQLS   Hif-1a
SFE-ETVEILFEAGASAELDDCRGVSENVILGQMAPIGTGAFDVMIDEESLVKYMPEQK   1i50_A (Rpb1)
... ......1.........2.........3.........4.........5........   1400 - 1458
```

```
i..d......1.......i..2.........3.........4.........5.   531 - 582
-FKLELVEKLFAEDTEAK-NPFSTQDTDLDLEMLAPYIPMDDDLQLRSFDQLS   Hif-1a
SFE-ETVDVLMEAAAHGESDPMKGVSENIMLGQLAPAGTGCFDLLLDAEKCKY   Rpb1 (Human)
... ......1.........2.........3.........4.........5..   1400 - 1451
```

The size of the search space in the problem of finding the optimal alignment with gaps scales exponentially with the length of the sequences considered. Indeed, the total number of non-redundant global alignments (two alignments are redundant if they result in the same score, f_{tot}) for two sequences of length n and m is given by $(n+m)!/[m!n!]$. This is a

simple consequence of the one-to-one correspondence between alignments and intercalated sequences stated in our definition, and it may be easily verified as follows. The order of each of the sequences is preserved when intercalating them, and therefore, we have in fact $n+m$ positions to place m elements of the second sequence (once this is done the position of each of the elements of the first sequence is fixed unambiguously). Hence, the number of intercalated sequences is simply the number of m-element combinations of $n+m$ elements.

Gaps allow one to take into account important evolutionary events that lead to insertions or deletions of stretches of nucleotides (and consequently amino acids) of various length, leading to proteins of similar core structures and functions, but of different lengths. It is the introduction of gaps, however, which makes the problem scaling exponentially. In light of the huge and ever growing size of the biological sequences databases, the importance of efficient solutions to this problem can hardly be overstated. This is exactly why DP is so important in bioinformatics - using DP the problem may be solved in the order of $O(n \times m)$ steps, i.e. the optimal alignment may be found in polynomial time [Durbin et. al., 1998].

This dramatically less expensive solution is achieved by breaking the problem into subproblems. Only best partial alignments up to a given point are considered and then another pair of characters is added to the alignment, depending on what is the optimal extension of a given partial alignment. For the problem of the global alignment and the linear gap penalty considered here (with a score for aligning a residue with a gap defined as $f_s(-,\alpha_i) = f_s(\alpha_i,-) = -d\,;\ d>0$), the particular DP solution is known as the Needleman-Wunsch algorithm [Needleman and Wunsch, 1990], which consists of two steps: the construction of the so-called DP representing possible alignments table and the trace back procedure to identify the optimal alignment.

Figure 5. Dynamic Programming table for global alignment

		H	E	A	G	A	W	G	H	E
	0	<-8	<-16	<-24	<-32	<-40	<-48	<-56	<-64	<-72
P	^-8	*-2	*-9	*-17	<-25	*-33	<-41	<-49	<-57	*-65
A	^-16	^-10	*-3	*-4	<-12	*-20	<-28	<-36	<-44	<-52
W	^-24	^-18	^-11	*-6	*-7	*-15	*-5	<-13	<-21	<-29
H	^-32	*-14	*-18	*-13	*-8	*-9	^-13	*-7	*-3	<-11
E	^-40	^-22	*-8	<-16	^-16	*-9	*-12	^-15	*-7	*3

HEAGAWGHE

--P-AW-HE

The DP table represents all the possible alignments. However, starting from the first pair of characters, only these partial alignments are traced, which proceed through locally

optimal extensions of partial alignments up to a given point, defined using the following recursive rules:

$$f_{\text{tot}}(0,0) = 0; \quad f_{\text{tot}}(k,0) = f_{\text{tot}}(0,k) = -kd$$

$$f_{\text{tot}}(i,j) = \max\{ f_{\text{tot}}(i-1,j-1) + f_s(a_i, b_j), f_{\text{tot}}(i-1,j) - d, f_{\text{tot}}(i,j-1) - d \}. \quad (5)$$

Therefore, the optimal alignment can be then traced back, starting from the lower right corner of the DP table, as shown in the example included in Figure 5.

The BLOSUM50 scoring matrix (for instance $f_s(P,H) = -2$) and the gap penalty $d = 8$ were used in this example. Note that symbols *, ^ and < are used to indicate which of the three possible extensions of the alignment was optimal, corresponding to the alignment of an amino acids in the first sequence with an amino acid in the second sequence, a gap in the first sequence with an amino acid in the second sequence, or an amino acid in the first sequence with a gap in the second sequence, respectively.

The above symbols represent in fact pointers that allow one to efficiently trace back the optimal alignment. Since the number of the cells in the DP table is $(n+1)\times(m+1)$ and a fixed number of operations per cell are required, it is easy to see that the overall complexity of the Needleman-Wunsch algorithm is indeed polynomial (quadratic in n assuming for simplicity that $n = m$) in time and space. Further discussion of this algorithm may be found in [Durbin et. al., 1998].

There are many extensions and modifications of this basic scheme, such as the Smith-Waterman [Smith and Waterman, 1981] algorithm for local alignments. Dynamic Programming is truly ubiquitous in sequence analysis [Gibson and Muse, 2002; Pevzner, 2000, Gusfield, 1999]. On the other hand, however, DP with its quadratic polynomial complexity may be computationally too expensive for large-scale applications. Therefore, many heuristic schemes, such as BLAST [Altschul et. al., 1997], which are more efficient but are not guaranteed to find the optimal solution, were devised.

The sequence-to-structure matching may be perceived as a generalized sequence matching, with one of the sequences consisting of amino acids and the other of structural sites characterized in terms of their structural environment (e.g. the number of neighbors to a site). Therefore, DP techniques may be directly applied to solve efficiently the problem of finding optimal sequence-to-structure alignments. In light of considerations included in this section, however, scoring functions for efficient sequence-to-structure matching should enable piecewise approach and decomposition of the problem into "local" subproblems. This observation is the starting point for the developments summarized in the next section.

I.4. Contact Potentials for Protein Recognition

Protein structure is often represented in terms of simplified, reduced models that speed up computation. For example, the commonly used contact model represents each amino acid by just one point, which defines the approximate location (site) of an amino acid. The overall shape of a protein may be characterized in terms of contacts between closely packed amino acid residues, or in other words in terms of effective interactions between the structural sites representing amino acid residues.

Such contact models allow one to capture the packing of hydrophobic residues that are buried in the core of the protein and contribute to the stability of the structure.

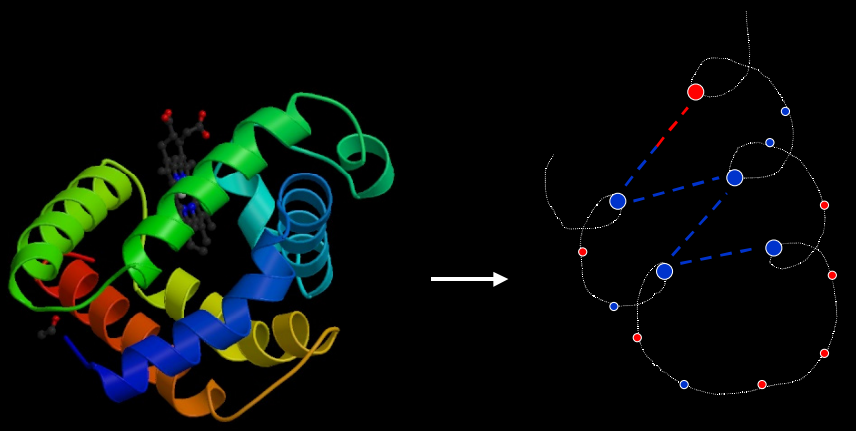
Hydrophobic residues are represented as blue circles in Figure 6, as opposed to hydrophilic residues that are marked in red and are predominantly found on the surface of globular proteins [Branden and Tooze, 1991].

Let us consider widely used inter-residue folding potentials. In contact pairwise models [Sippl et. al., 1992; Bryant et. al., 1993; Godzik et. al., 1992] the energy of a protein with sequence S and structure $\mathbf{X}$ is a sum of pair energies from all pairs of interacting amino acids:

$$E(S,\mathbf{X};\mathbf{z}) = \sum_{i<j} z_{\alpha_i\beta_j} = \sum_{\gamma} z_\gamma n_\gamma(S,\mathbf{X}) . \quad (6)$$

The summation index, $\gamma \equiv \alpha\beta$, runs over 210 different contact types, where α and β denote types of amino acids ($\alpha,\beta \in \{1,2,...,20\}$) at certain sites i and j in contact, and $n_\gamma(S,\mathbf{X})$ denotes the number of contacts of a specific type found in the structure $\mathbf{X}$. Thus, given the effective "pair energies", $z_\gamma \equiv z_{\alpha\beta}$ (also denoted as $\varepsilon_{\alpha\beta}$ throughout papers included in this dissertation), computing the overall energy of a structure reduces to counting of different types of contacts. Sites i and j are said to be in contact, if their distance, r_{ij}, is sufficiently small. In this work we consider the model that was used before to optimize threading potentials [Tobi et. al., 2000], with geometric side chain centers as interaction sites. Two sites are assumed to be in contact if their distance satisfies $1.0 < r_{ij} < 6.4$ Å, which implies that only neighbors from the first contact shell are taken into account. Furthermore, $|i-j| \geq 4$, i.e. pairs of residues that are separated by fewer than four virtual bonds are excluded.

Figure 6 Reduced representations of protein structure



Each amino acid represented by a point in the 3D space; simple c ontact model – two amino acids in contact if their distance smaller than a cut off.

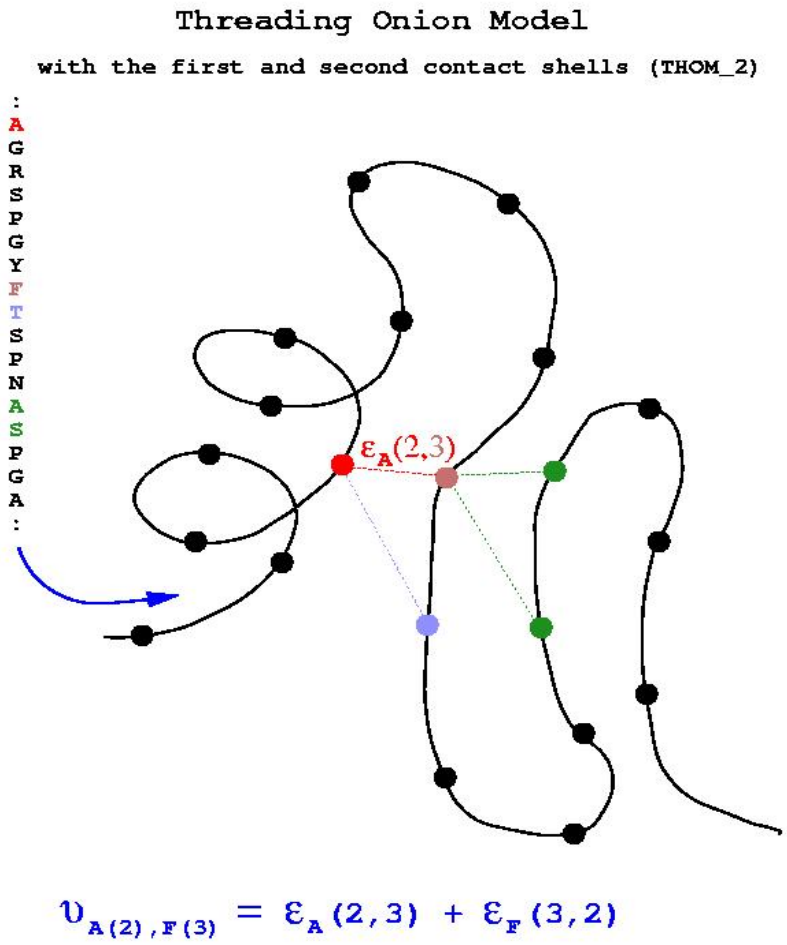
The effective pair energies for inter-residue interactions can be derived from the analysis of contacts in known structures, with z_γ defined by the frequency of observing contacts of type γ normalized by the so-called background frequencies [Sippl et. al., 1992]:

$$z_{\alpha\beta} = -C \ln[\frac{p_{\alpha\beta}}{p_\alpha p_\beta}] . \quad (7)$$

Here, C is a positive constant that defines the energy scale, $p_{\alpha\beta}$ denotes the probability of observing (in a set of native structures) amino acids of types α and β in contact, whereas p_α and p_β denote probabilities of observing these individual amino acids (again in a set of native structures). Such knowledge-based, pairwise potentials are widely used in fold recognition [Jones et. al., 1992; Bryant et. al., 1993; Mirny and Shaknovich 1998], *ab-initio* folding [Sternberg et. al., 1999; Liwo et. al., 1997; Xia et. al., 2000] and sequence design [Babajide et. al., 1997, 1999]. Alternative strategies to find the effective pair energies (parameters of folding potentials in general) are discussed below.

It is important to realize that such simplified models incorporate the interactions with the solvent in terms of the effective pair energies. Proteins adopt their three-dimensional conformations in specific environments. Soluble proteins fold in an aqueous environment, whereas membrane proteins fold in a lipid environment. Thus, effective pair energies must be derived separately for different environments in order to account for the observed (in a given environment) structure.

Figure 7. A novel contact model for protein recognition



As an alternative to pairwise contact models, one may consider the so-called "profile" models [Bowie et. al., 1991; Elofsson et. el., 1998], in which the overall effective energy of a

protein takes the form of a sum of individual site contributions, depending on the structural environment of a site. For example, the solvation or burial state or the secondary structure can be used to characterize different local environments.

The advantage of profile models is the simplicity of finding optimal alignments with gaps (deletions and insertions into the aligned sequence) that allow the identification of homologous proteins of different length. As discussed in the previous section, using DP algorithm one may compute optimal alignments with gaps in polynomial time, as compared to the exponential number of all possible alignments, if a "local" scoring function is used.

In contrast to profile models, the potentials based on pair energies do not lead to exact alignments with dynamic programming. The reason for that may be explained by considering how a score for aligning an amino acid residue with a structural site is computed when using pairwise potentials. Namely, all contacts to a site need to be considered, each contributing an effective pair energy that is dependent on the identity of the "other" amino acid in contact. However, the placement of gaps (i.e. the alignment) may change the identity of the "other" residues and the problem becomes non-local (NP-complete in fact [Lathrop, 1994]).

A number of heuristic algorithms, providing approximate alignments, have been proposed, e.g. [Lathrop and Smith, 1996]. However, they cannot guarantee an optimal solution with less than exponential number of operations. We introduced a novel energy function that employs reduced, contact models of protein structure and blends the contact energies with profile models to achieve computational efficiency and higher accuracy in recognition of native-like structures [Meller and Elber, 2002]. The new model is called THreading Onion Model 2 (THOM2) since it uses information about the first and the second contact shells of an amino acid residue and it incorporates some cooperativity effects that are not included in standard pairwise folding potentials.

In THOM2 one defines the effective energy $z_{\alpha_i}(n_i, n_j)$ (also denoted as $\varepsilon_{\alpha_i}(n_i, n_j)$ in some of the figures and papers included here, see Figure 7 for example) of a contact between structural sites *i* and *j*, where n_i is the number of neighbors to site *i* and n_j is the number of neighbors to site *j*. The type of amino acid at site i is α_i. Only one of the amino acids in contact is known. The total contribution to the energy of site *i* is a sum over all contacts to this site $\phi_{i,\mathrm{THOM2}}(\alpha_i, \mathbf{X}) = \sum_j{}' z_{\alpha_i}(n_i, n_j)$. The prime indicates that we sum only over sites *j* that are in contact with *i*, where contact is defined as previously for pairwise models. The total energy is finally given by a double sum over *i* and *j*,

$$E_{THOM2} = \sum_i \sum_j{}' z_{\alpha_i}(n_i, n_j) \ . \tag{8}$$

As was the case for pairwise models defined before, computing the overall energy of a structure reduces to the counting of different types of contacts, $n_\gamma(S, \mathbf{X})$, which are however defined in terms of the number of neighbors to sites involved in contact and identity of the amino acid occupying the "primary" site. Therefore, we may express the overall energy as linear combination with respect to the parameters z_γ:

$$E_{THOM2}(S, \mathbf{X}; \mathbf{z}) = \sum_\gamma z_\gamma n_\gamma(S, \mathbf{X}), \tag{9}$$

where the summation index is defined now in terms of the amino acid type occupying the primary site, its number of neighbors and the number of neighbors to the other site involved in contact, $\gamma \equiv \gamma(\alpha_i, n_i, n_j)$. We use a coarse-grained model leading to a reduced set of

structural environments (types of contacts) by merging residues with similar number of neighbors into several classes. Therefore, the number of parameters, which might be very large in principle (assuming up to ten neighbors to a site we would obtain $20 \times 10 \times 10 = 2000$ parameters), is reduced to a number comparable with 210 parameters of the pairwise model (see Paper 1 for details).

Since each contact contributes twice to the overall energy, it is possible to define an effective pair energy using THOM2 as well (see also Figure 7):

$$V_{ij}^{eff} = z_{\alpha_i}(n_i, n_j) + z_{\alpha_j}(n_j, n_i) \ . \tag{10}$$

Hence, one can formally express the THOM2 energy as a sum of pair energies,

$$E_{THOM2} = \sum_{i<j} V_{ij}^{eff} \ . \tag{11}$$

The effective energy mimics the formalism of pairwise interactions. However, in contrast to the usual pair potential, the optimal alignments with gaps can be computed efficiently with THOM2, since structural features alone determine the "identity" of the neighbor.

The energy terms (parameters of the potentials), $z_{\alpha_i}(n_i, n_j)$, could be computed using statistical approach for example, in analogy to knowledge based pairwise potentials defined in equation (7). However, such statistical potentials "learn" from the native structures ("good" examples) only. In order to increase their power to distinguish misfolded states (the "bad" examples) from native states, more sophisticated protocols incorporate data from decoy structures as well. One approach to designing potentials that improves upon statistical potentials is the so-called Z-score optimization, discussed in Paper 3.

Here, in order to achieve better discrimination of native structures with respect to misfolded decoys, we explicitly demand that the folding potentials mimic the postulate that the native states have the lowest energy. Such formulation leads to a problem of solving linear system of inequalities, which we chose to solve using Linear Programming techniques (for an overview of LP and other techniques and algorithms for solving linear systems of inequalities the reader is referred to [Vanderbei, 1996]).

We used LP methods to design and evaluate several scoring functions (including THOM2) for threading and to optimize their parameters. For example, the site energies $z_{\alpha_i}(n_i, n_j)$ are optimized using the LP protocol to find a solution of a large set of linear inequalities derived from a large set of native and misfolded structures as described in the next section. LP is also used to determine optimal gap penalties. The new model provided an efficient threading approach for annotations of remote homologs that share structural similarity without significant sequence similarity. Applications of this approach are presented in papers included in Part III of this dissertation.

I.5. LP Approach to the Design of Folding Potentials

In both ab-initio folding and protein recognition we are faced with the problem of finding (designing) an appropriate expression for the free energy or scoring function (also called here folding potentials), respectively. The basic requirement for protein folding potentials is their ability to distinguish native-like from non-native structures. This can be achieved by an appropriate choice of the functional form and parameters of the energy function (in the

following we will use the "physical" convention according to which well folded structures are expected to yield low energies, as opposed to high scores when using scoring functions).

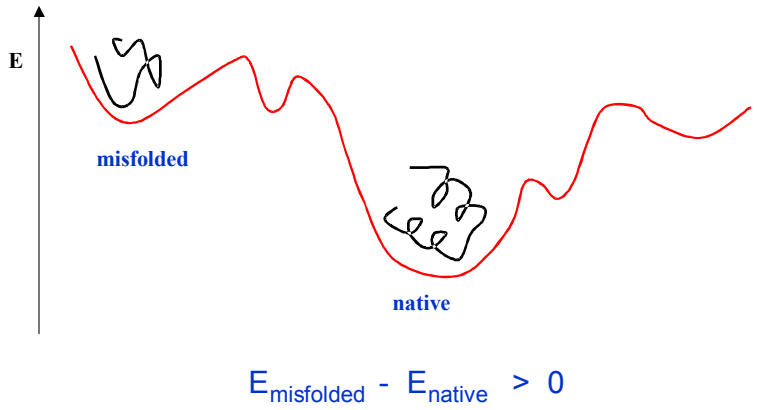
Assuming that folding potentials are expected to have the lowest energy for the native fold, one may impose that for each pair of native and misfolded structures that are considered the following constraints are satisfied:

$$\Delta E_{\text{mis, nat}} \equiv E_{\text{misfolded}} - E_{\text{native}} \geq \varepsilon \quad . \tag{12}$$

Here, $E_{\text{native}} \equiv E(S, \mathbf{X}_{nat}; \mathbf{z})$ is the energy of the native structure $\mathbf{X}_{nat}$, $\mathbf{z}$ is the vector of parameters to be optimized, $E_{\text{misfolded}} \equiv E(S, \mathbf{X}_{mis}; \mathbf{z})$ represents the energies of the misfolded (non-native) structures $\mathbf{X}_{mis}$ and ε is a positive constant. In other words, we require that the energies of native structures are lower than the energies of misfolded structures.

It should be noted that casting the problem of designing folding potentials in terms of optimization of the parameters $\mathbf{z}$, such that correct recognition (classification) of a set of examples (pairs of native and misfolded structures) is imposed in the training, implies that the problem is in fact formulated within the framework of the supervised classification approach. Obviously, as with any other supervised classification protocol, the choice of training set of examples and further validation of the results on independent control sets is critical for the successful optimization of folding potentials. Discussion of different issues involved in making these critical choices is included in Paper 2.

Figure 8. Recognition of native structures by folding potentials



For energy models considered here, such as the contact potentials defined in (6) and (9), one may in general expand the energy as linear combination in terms of their parameters:

$$E(S, \mathbf{X}; \mathbf{z}) = \sum_{\gamma} z_{\gamma} \alpha_{\gamma}(S, \mathbf{X}), \tag{13}$$

with the coefficients of the linear combination, $\alpha_{\gamma}(S, \mathbf{X})$, taking a model specific (structure and sequence dependent) form. In such case, the set of inequalities in equation (12) that