The Statistical Mechanics Approach to Protein Sequence Data: Beyond Contact Prediction

Christoph Feinauer

A thesis submitted for the degree of
Doctor of Philosophy

PhD Program in Physics, XXVIII Cycle
Advisor: Prof. Andrea Pagnani

Contents

1 Introduction: Proteins, Multiple Sequence Alignments and Co-Evolution

1.1 Proteins and Structure

This Section is based on the introductory chapters on proteins of [1] and [69].

Proteins make up the largest part of the dry mass of the cell and are involved in virtually every organized event happening in the cell [1]. They are used as structural elements, enzymes, parts of molecular motors, hormones (and generally signaling molecules), carriers, and more. Proteins are coded in genes and their expression and maturation is a central part of the (auto-)regulation of cells.

There is a huge number of different proteins. In the human body alone there are around 20000 protein coding genes which produce a much higher number of different proteins, taking post-transcriptional modification into account. Proteins show an enormous variety in structure and function and the picture becomes even more complicated when protein complexes, consisting of many proteins binding to each other transiently or permanently, are taken into account.

Given their importance and complexity, it is not surprising that many diseases like Alzheimer's or Cancer are linked to malfunctioning or missing proteins and a better understanding of how proteins work might lead to better treatments for such sicknesses.

The structure of a protein is closely related to its function, a claim that is known as the *structure-function paradigm* [78]. Therefore it is often very advantageous to know the structure of a protein if one wants to understand what the protein does and how it does it.

A protein can be seen as a chain of amino acids, linked to each other by a covalent peptide bond. Each amino acid carries one of twenty different side chains which all have different chemical properties. The sequence of the amino acids along the chain is called the primary structure of the protein. The chain can bend and twist and certain amino acid pairs can form covalent or non-covalent bonds, so that after folding of the chain a stable structure emerges. This structure is usually divided into 4 parts, which are

- *Primary Structure*: The amino acid sequence of the protein along its peptide-bond backbone
- *Secondary Structure*: Motifs based on relatively local hydrogen bonds between the amine hydrogen and carboxyl oxygens, building often found elements such as α-helices or β-sheets
- *Tertiary Structure*: The 3D conformation of the protein. This is determined by how the chain, already locally folded into secondary structure, makes contacts with itself (for example by hydrogen bonds or disulfide bridges).

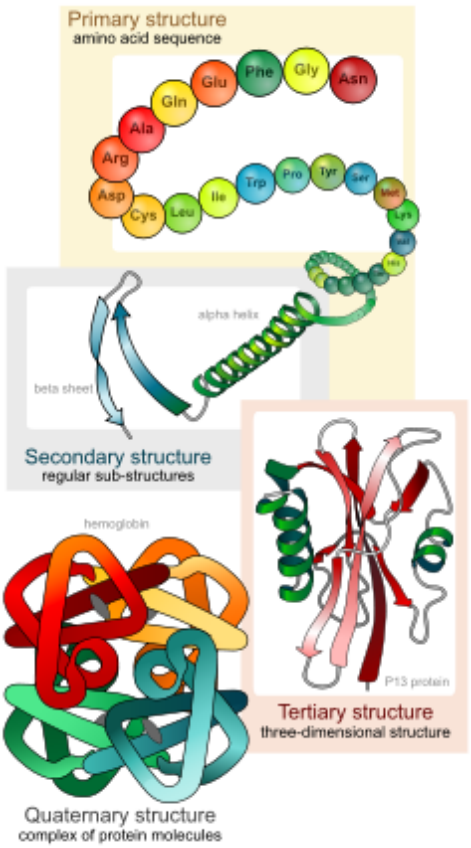


Figure 1.1: Different levels of protein structure. Figure taken from `https://en.wikipedia.org/wiki/Protein_structure`

- *Quaternary Structure*: Several protein chains bound to each other, building a complex.

Even though the sequence of a protein largely determines its structure (something known as *Anfinsen's Dogma* [2]), this mapping from sequence to structure is not trivial even with modern computational techniques. In fact, *computational protein folding* is one of the most worked on topics in biophysics and bioinformatics [11].

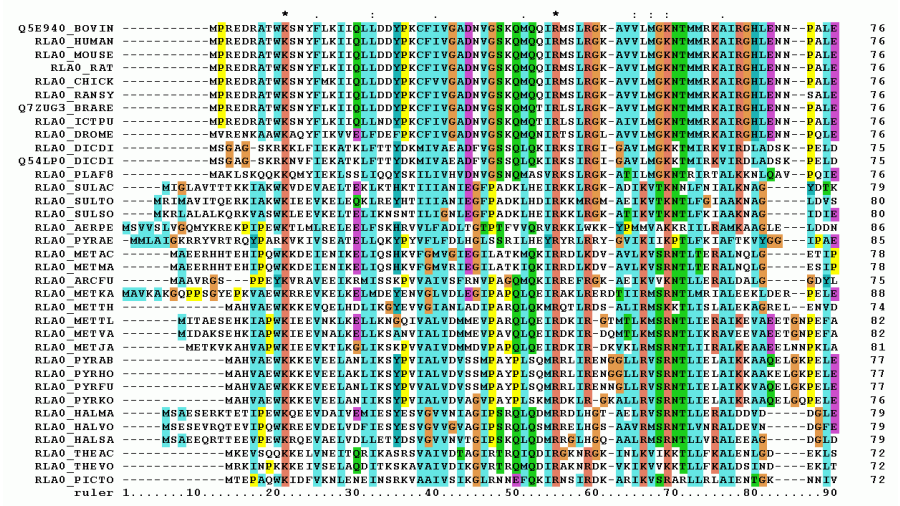


Figure 1.2: Multiple Sequence Alignment, Rows corresponds to proteins (leftmost entry is the protein name), Columns correspond to consensus positions (residues); Figure taken from `https://en.wikipedia.org/wiki/Multiple_sequence_alignment`

1.2 Multiple Sequence Alignments

1.2.1 Protein Families

Proteins evolve and mutations in their sequences occur. This comprises amino acids substitutions, inserts and deletes. Even though the structure of a protein is very intimately connected to its function, many different sequences lead to the same structure and leave the functionality of the protein unimpaired (or only slightly impaired or even improved).

It is natural to group *homologous* proteins, which have similar structure and function due to their phylogenetic relationship, and treat them effectively as different versions of the same protein. The set of such sequences makes a *protein family* [59].

In order to make the data more amenable to statistical analysis it is favorable to define consensus residues of the family and map the amino acids of the single proteins onto them. This leads to a data matrix that contains for every sequence m an amino acid a_i^m belonging to the consensus site i (or a gap symbol if no such amino acid can be found). This data matrix is called a multiple sequence alignment (MSA) and online databases like PFAM [77] contain large amounts of protein families (more than 16000 at the conception of this thesis) with up to several hundred thousand sequences.

The creation of quality MSAs is largely a task for bioinformatics and most of the work presented here considers the MSA as given. Nonetheless, it is favorable to have

some understanding of the main algorithms used in this field in order to understand the data and its idiosyncrasies better (Section 3 for example treats the problem of gap-stretches, something that is fundamentally an artefact of the alignment generation procedure). We therefore give a short discussion of the central concept of profile Hidden Markov Models (HMMs), which underly all of the alignments used in this thesis via bioinformatical algorithms and programs such as HMMER [35] and HHBlits [79]. This exposition follows closely the corresponding chapters of [24].

1.2.2 Profile Hidden Markov Models

The creation of a large MSA usually needs a smaller MSA of a few sequences as input (HHBlits can also start from a single query sequence). These small alignments should be high quality and contain only sequences that are with high confidence members of the protein family one wants to model. The Pfam database (`http://pfam.xfam.org/`) makes curated seed alignments available, together with the final alignment created with HMMER. The creation of a seed alignment from unaligned sequences is still another topic. This can be done manually or using programs such as MAFFT [53].

Given a seed alignment, one usually wants to extract more sequences of this family from a large database of protein sequences like Uniprot [16] (which contains about $56 \cdot 10^6$ sequences at the conception of this thesis). The way this is usually done is to train a probabilistic model on the seed alignment and then search for sequences in the database that have a high probability given this model. This is not trivial since the sequences in the database are not aligned to the seed alignment and insertions and deletions have to be taken into account. Arguably the most popular models for this task are Profile Hidden Markov Models (HMM) [24].

A profile hidden markov model defines a probability for a sequence of *states* of variable length. This probability distribution has the structure of a markov chain, which means that the probability of a state s_i at the i^{th} position in the chain is conditionally independent of the states $s_1, \ldots, s_{i-2}$ given s_{i-1}:

$$P(s_i \mid s_{i-1}, s_{i-2}, \ldots, s_1) = P(s_i \mid s_{i-1}) \tag{1.1}$$

The symbols can have one of three different types, insertion states I_j, deletion states D_j and match states M_j. These states are indexed with the consensus residue j of the protein they correspond to. Note that above we used i to index the state of the HMM, which is *not* the same as a consensus residue in the protein. In fact, several consecutive states of the markov chain can belong to the same consensus residue. An easy example is an insertion of an amino acid after a consensus residue j of the family, where this insertion is specific to this one protein sequence. The HMM models this insertion as coming from a state I_j, and it belongs to the same consensus residue as the preceding amino acid which belongs to the match state M_j.

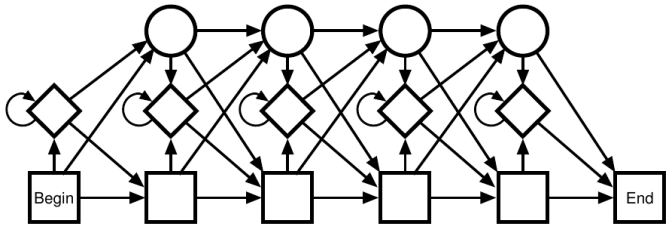


Figure 1.3: A profile HMM. Squares correspond to match states, diamonds to insert states and circles to delete states, Figure taken from [24]

Insertions and deletion states emit amino acid with a probability depending on the amino acids and the position on the chain, while deletion states emit no residue (a gap symbol can be inserted instead). Therefore protein sequences generated from such a model can have different lengths.

The transition probabilities to go from one state to the other define the probability distribution, e.g.

$$P(s_5 = M_3 \mid s_4 = D_2), \tag{1.2}$$

which is the probability to go to a match state for consensus residue 3 if the last state was a delete state corresponding to consensus residue 2. In fact, files defining HMMs as for example the pre-calculated HMMs that Pfam [36] makes available for their protein families are not much more than a table with probabilities to jump between different states and the amino acid emission probabilities for the different states (dependent on the consensus residue). We summarize from now on all these probabilities defining the HMM by the capital letter H.

An pictorial representation of a profile HMM, found in [24] can be found in Figure 1.3. Here, squares correspond to match states, diamonds to insert states and circles to delete states.

The procedure to arrive at a MSA is to estimate the transition probabilities from the seed alignment and then to search for sequences in the database that have a large probability given the parameters H. Emission probabilities for amino acids and transition probabilities for the states can be estimated directly from the multiple sequence alignment when the state sequence is known. Some care has to be taken to avoid overfitting, like not assigning a zero probability to amino acids never seen at a residue, especially when dealing with seed alignments that consist of only a few sequences.

Given a new and unaligned sequence we do not know its possible state sequence, but only the emitted symbols. Technically, one would like to calculate the probability of the sequence given the model H,

$$P(e \mid H) = \sum_{s} P(e|s, H)P(s, H) \tag{1.3}$$

where e is the observed (emitted) sequence and the sum runs over all sequences s with their model probability $P(s)$. According the Markov property,

$$P(s) = P(s_1) \prod_{i=2}^{N_s} P(s_i \mid s_{i-1}). \tag{1.4}$$

The quantity $P(e \mid H)$ can be calculated by a dynamic programming algorithm called the forward algorithm [24]. Alternatively one can extract the s that gives the maximal contribution of the sum in Equation 1.3 and calculate

$$P(e, s^* \mid H) = P(e \mid s^*, H)P(s^* \mid H) \tag{1.5}$$

where $s^* = \underset{s}{\text{argmax}}\ P(e, s \mid H)$. This can be calculated by the Viterbi algorithm [24].

Whether to include the sequence in the multiple sequence alignment or not can then be decided by looking whether a score derived from these probabilities exceeds some threshold. HMMER [35] for example looks at the ratio of the probability of the sequence given H and the probability of a sequence in a random model obeying only the background amino acid frequencies (the log of this ratio is called the log-odds score). In addition, one can introduce E-values, which measure how likely it is that a random sequence achieves a higher log-odds ratio than the sequence under investigation. This gives an estimate of false positives in the alignment and can be used to control specificity.

1.3 Statistical Analysis of Protein Sequence Data and Co-Evolution

Having generated a MSA, statistical analysis can be conducted. An example is to look for conserved sites, having a low entropy [60]

$$H_i = -\sum_{a} f_i(a) \log f_i(a), \tag{1.6}$$

where $f_i(a)$ is the frequency of occurence of amino acids a in residue (MSA column) i and the sum runs over all possible amino acids. Such conservation might be indicative of evolutionary conservation and evolutionary conservation might be indicative of importance for structure and/or function [89].

An alternative hallmark for structural constraints influencing the amino acid distributions in an MSA is residue co-evolution. Put forward more than 20 years ago [40],

this reasoning has given rise to its own branch of research [20]. Co-evolution can arise for example when two residues are in contact with each other and the fitness of the protein is impaired when a mutation occurs which makes this contact unstable. This should lead to correlated amino acid substitutions at contacting residues and, if correlated residues are found in the MSA, can vice versa be used to search for evidence of a contact in the MSA.

Several problems make this *protein contact inference* not trivial:

- Co-evolution must not necessarily originate in a contact but may be due to functional constraints
- Correlation might not be a good indicator for a *direct* co-evolutionary signal; residue i might be correlated to a residue j without being in contact with it, because both are in contact with a third residue k
- Strongly conserved residues (which are a priori more likely to contact other residues, see Figure 2.3) may show little to no variation, in which case only a low correlation signal may be detected
- MSAs might have only a few sequences and the correlation signal may be noisy
- The sequences in the MSA might have diverged only recently, which introduces a phylogenetic bias that distorts the correlation signal
- Experiments are biased to sample from organisms that are of academic or medical interest, which introduces a further bias in the correlation signal
- In homo-dimers intra-protein co-evolution is not distinguishable from co-evolution due to the inter-protein contacts

Nonetheless, the field of statistical analysis of protein sequence data and especially the extraction of co-evolutionary signals between residues is active and thriving. In the next section, we will review some methods for protein contact prediction that address the problems presented above, especially the disentangling of direct and indirect co-evolution between protein residues.

2 Methods: Existing Approaches for Predicting Residue Contacts

All methods used in this thesis are part of the DCA approach to contact prediction, based on the Generalized Potts Model and brought to fame in [68]. Nonetheless, it is necessary to point out that the field of contact prediction is more than 20 years old [40] and it seems therefore necessary to discuss also some other approaches. We therefore present in this Section a relatively old but basic approach, based on *Mutual Information*, and two newer ones, PSICOV and the application of Bayesian Trees. The latter two perform similarly to DCA for contact prediction. We also mention that non DCA approaches, such as PSICOV for example, are important inputs for meta-methods, which arrive at the best overall performance for contact prediction to date [51, 84].

2.1 Some Non-DCA Approaches

2.1.1 Mutual Information

Mutual Information (MI) is one of the preferred measures of correlation between two discrete stochastic variables X and Y in information theory [66] and defined as

$$I_{XY} = I_{XY} = \sum_{A,B} P_{XY}(A,B) \log\left(\frac{P_{XY}(A,B)}{P_X(A)P_Y(B)}\right), \tag{2.1}$$

where P_{XY} is the joint probability distribution of X and Y and the sum runs over all values A and B that the random variables can take. In terms of the entropy H of the stochastic variables X and Y this can be rewritten as:

$$I_{XY} = H_X - H_{X|Y} = H_Y - H_{Y|X} \tag{2.2}$$

Intuitively, this can be read as the amount of information of the variable X that is left after the information that Y contains about X has been subtracted.

One can apply this measure to protein sequence data by treating the columns of a MSA as realizations of random variables and calculate the (empirical) mutual information I_{ij} between all column pairs i and j using the data distribution $f_{ij}(a,b)$,

$$I_{ij} = I_{ji} = \sum_{A,B} f_{ij}(A,B) \log\left(\frac{f_{ij}(A,B)}{f_i(A)f_j(B)}\right), \tag{2.3}$$

where $f_{ij}(a,b)$ is the frequency of co-occurrence of amino acid a at residue i and amino acid b at residue j.

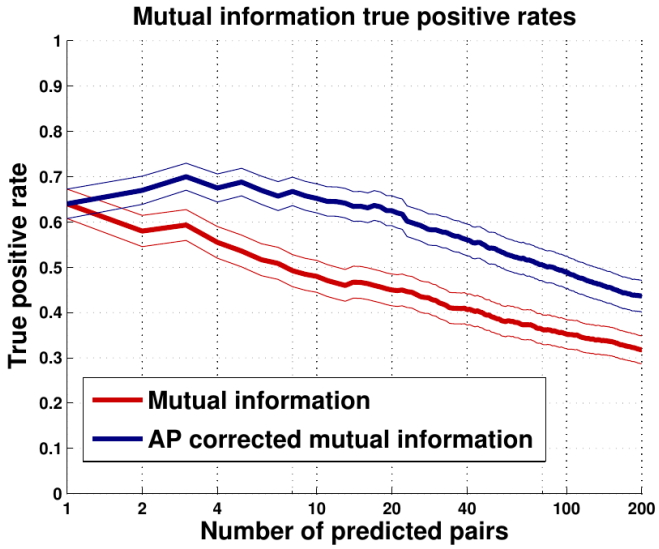


Figure 2.1: True positive rate (true positives divided by number of predictions) in the n top-ranking predictions, where n is indicated by the y-axis. The red curve corresponds to mutual information, the blue curve to mutual information with an average product correction as presented in [23]. The test-set used here are the 53 proteins analyzed in [68].

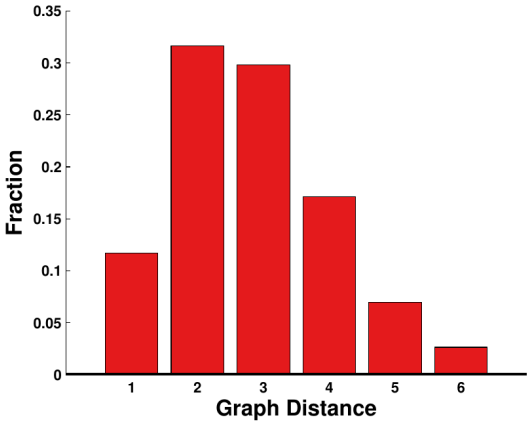


Figure 2.2: Histogram of distances between nodes based on the contact maps derived from the 53 pdb files corresponding to the data set analyzed in [68].

Figure 2.1 shows that using the mutual information of a residue pair as a score for this pair being a contact does not work very well. Even the highest scoring prediction is a true positive in only about 60% of all cases. Several reasons might be responsible for this:

Indirect couplings: First, note that mutual information as any measure of correlation is also a measure of how well knowledge of the variable X can be used to predict the outcome of a measurement of the variable Y. This is connected to, but not identical to, the quantity that we assume to be an indication for a 3D contact: A strong *direct* influence of the variable X on Y. For predicting the outcome of one variable given the other, on the other hand, the kind of connection between the variable is not important. A prototypical example of a case where this makes mutual information a bad proxy for direct coupling is when two residues which are not in contact in a protein show correlated amino acid substitutions because they are both in contact with a third residue. An indication of the possible extent of this problem is the pronounced interconnectedness of the protein residue contact network, as shown by the graph distances (the number of edges of the shortest path between two nodes in the contact network) in Figure 2.2. A more sophisticated analysis of the problem can be found in [13], where it is shown that highly correlated but non-contacting residues often have chains of co-evolving contacting residues between them.

Site entropy: Another problem consists in the different entropy (variability) of the variables. Mutual information can be shown to be positive semidefinite, so Equation 2.2 sets the upper bound as $\min(H_X, H_Y)$ and a prediction based on mutual information will be biased to predict more contacts at higher entropy of the two corresponding positions. Intuitively, the relation should be vice versa: Taking part in a contact *induces* a evolutionary pressure for conservation in a position, *lowering* the entropy of

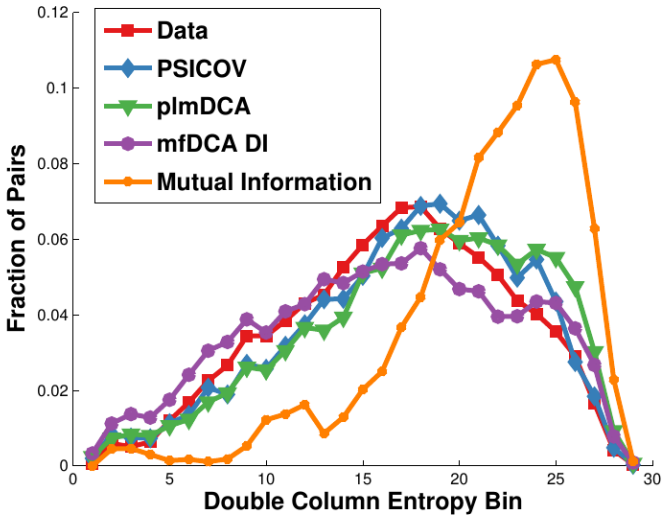


Figure 2.3: Mutual information is biased to predict contacts between pairs with a high sum of entropies. The bins have sizes 0.2. 'Data' shows the fraction of all contacts measured in the PDB files of the 53 analyzed proteins in [68] falling into a bin depending on the sum of the entropies of the two sites into the bins. 'mfDCA DI','plmDCA','PSICOV' and 'Mutual Information' show the fraction of the first 200 predicted pairs per bin.

the stochastic variable connected to that position [13].

Figure 2.3 shows the dependence of the probability of receiving a high score for a pair on the sum of the entropies of the residues of the pair for several methods. It can be seen that MI indeed tends to assign a high score when the sum of the column entropies is higher, while the more successful methods presented later on follow rather closely the distribution as measured in PDB files.

Phylogenetic background The multiple sequence alignment consists of sequences that (ideally) have a relatively recent common ancestor. It is natural to assume that two sequences look more similar if they are closer in the phylogenetic tree originating from this common ancestor. This violates the assumption that the measured sequences are independent and identically distributed (i.i.d.). It might for example happen that two positions with no mutual influence get fixed and conserved in their value in all branches following generation G (but possibly with different values in the individual branches). The two positions then show a high degree of correlation (and mutual information) resulting from phylogeny alone.

A partial remedy for these shortcomings can be found in an interesting approach called Average Product Correction (**APC**) [23]. This approach aims to minimize

any background influences like phylogeny and site entropy. We follow a sketchy but instructive derivation found in [13] and assume that the mutual information I_{ij} of positions i and j is made of two parts; I^r_{ij} due to a real mutual influence (be it indirect or direct) and B_iB_j, a product of single-site characteristics that do not depend on the partner:

$$I_{ij} = I^r_{ij} + B_iB_j \tag{2.4}$$

These single site characteristics might stem for example from a higher or lower entropy of the residue with respect to the mean. By estimating this contribution and subtracting it from the score one hopes now to correct for such a background. The special form in which the single site characteristics B_i and B_j enter in Equation 2.4 lead to the name of Average *Product* Correction. While other forms like the Average *Sum* Correction have been studied, the one presented here leads to the largest increase in prediction quality [23].

It is now assumed that $I^r_{ij} \ll B_iB_j$. Then the one- and two-site averages of I_{ij} will also be dominated by the single site contributions (average denoted by $\bullet$):

$$I_{i\bullet} \approx B_iB_\bullet \tag{2.5a}$$
$$I_{\bullet\bullet} \approx (B_\bullet)^2 \tag{2.5b}$$

and therefore

$$I^r_{ij} \approx I_{ij} - \frac{I_{i\bullet}I_{j\bullet}}{I_{\bullet\bullet}}. \tag{2.6}$$

Mutual information thus corrected has a significantly better prediction quality (although still not satisfactory in the absolute value) [23].

Notice that in the derivation there was no explicit reference to the characteristics of mutual information. This ansatz can therefore be used as well to correct other quantities for a dominating background. Indeed, APC has been shown to enhance the prediction quality also for other scores like the PSICOV score (see Section2.1.2), the posterior probabilities of the Bayesian network approach (see Section 2.1.3) and the Frobenius norms of plmDCA (see Section 2.2.3). The only score that seems to be unaffected by an average product correction term is the Direct Information as described in Section 2.2.2. This is interesting because it is not a priori clear what background influences are successfully corrected by APC. That Direct Information and Frobenius norm react differently to APC but use the same reweighting-technique to correct for phylogenetic bias might be a hint that APC is correcting mainly for an entropic bias and that this is a minor problem in the DI framework. In fact, plotting in Figure 2.4 the same type of graph as Figure 2.3 for the Frobenius norm and the average product

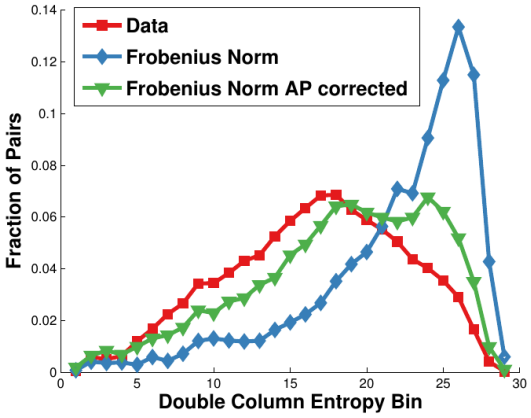


Figure 2.4: APC corrects the Frobenius Norm for entropic bias The bins have sizes 0.2. 'Data' shows the fraction of all contacts measured in the PDB files of the 53 analyzed proteins in [68] falling into a bin depending on the sum of the entropies of the two sites into the bins. 'Frobenius Norm' and 'Frobenius AP corrected' show the fraction of the first 200 predicted pairs per bin, with couplings inferred by plmDCA (see Section 2.2.3).

corrected Frobenius norm, we see that the distribution of the double column entropies is much closer to the real one after the correction.

2.1.2 Sparse Inverse Covariance Estimation (PSICOV)

A classic approach to disentangle direct from indirect contributions to correlations is the calculation of partial correlation coefficients. A successful application to protein sequences termed PSICOV (***P**rotein **S**parse **I**nverse **Cov**ariance Estimation*) was presented by Jones et al. in 2012 [50].

Partial correlation coefficients are a method to subtract from the correlation signal of two random variables within an intercorrelated system of random variables contributions that arise by the influence of the rest of the system on these two variables. Hence, this quantity should be thought of as the correlation between two variables, *after the influence of the other variables has been removed* [56].

To this end the authors transform the MSA into a numerical representation using binary variables x_{ia}^m, which are 1 if in the m^{th} sequence of the MSA one finds amino acid a at residue i, and 0 else (these binary variables are more formally introduced in the later Section 2.2.1, but the information given should be enough to follow this section). Denoting by $\overline{x}_{ia}$ average of these variables over the whole data set and by

$$C_{ia,jb} = \mathbb{E}\left[(x_{ia}^m - \overline{x}_{ia})(x_{jb}^m - \overline{x}_{jb})\right] \tag{2.7}$$

their covariance matrix, where the index ia corresponds to the integer $(i-1)\cdot q + a$ with q being the number of amino acids symbols in the alignment.

The partial correlation coefficients connected to the two variables x_{ia} and x_{jb} are defined as:

$$\rho_{ia,jb} = -\frac{\Theta_{ia,jb}}{\sqrt{\Theta_{ia,ia}\Theta_{jb,jb}}}, \tag{2.8}$$

where the matrix Θ is defined as

$$\Theta_{ia,jb} = (C_{ia,jb})^{-1}. \tag{2.9}$$

The idea to invert the covariance matrix to arrive at interaction scores will be revisited in different contexts in Sections 2.2.2 and 3. The problem encountered is also the same, namely that the correlation matrix C is not invertible.

Whereas mfDCA uses a pseudocount to cure the invertibility of the matrix (see Section 2.2.2), the authors of [50] apply methods of *sparse inverse covariance estimation* and search for an approximate solution to Equation 2.9, adding a sparsity prior. Notice that this sparsity corresponds to the assumption that most positions show no directly coupled mutations. Regarding contacts, this sparsity is empirically well-founded: as the number of contacts is on the order of N, while the number of *possible* contacts on the order of $\binom{N}{2}$. Sparsity priors (or at least priors favoring small couplings) are also used in DCA, for example by plmDCA (see Section 2.2.3).

In order to find an approximation to the inverse of a singular matrix with the constraint of sparsity, the authors in [50] use the graphical Lasso method. The final objective function can be seen as the negative log-likelihood in a Gaussian approximation (treating the variables as real variables) with the l1-norm of the matrix Θ as a regularizer:

$$\sum_{ia,jb} C_{ia,jb}\Theta_{ia,jb} - \log\det\Theta + \tau\sum_{ia,jb}|\Theta_{ia,jb}| \tag{2.10}$$

The first two terms have for an invertible C a minimum at $\Theta = C^{-1}$. The second part is the regularizer, favoring sparse solutions and ensuring convexity. Its strength is controlled by τ. In terms of Bayesian Inference this is nothing else than a exponential prior on the parameters multiplied with the posterior:

$$P(D,\Theta) = P(D|\Theta)P(\Theta) \tag{2.11}$$

with

$$P(D|\Theta) \propto \frac{e^{-M \sum_{ia,jb} C_{ia,jb}\Theta_{ia,jb}}}{(\det\Theta)^M} \tag{2.12}$$

$$P(\Theta) = e^{-\sum_{ia,jb} |\Theta_{ia,jb}|} \tag{2.13}$$

In order to speed up convergence, the sample covariance matrix C is **shrinked** towards a structured, unbiased estimator:

$$\hat{C}_{ia,jb} = \lambda\overline{C}\delta_{ia,jb} + (1-\lambda)C_{ia,jb}, \tag{2.14}$$

where $\overline{C}$ is the mean of the diagonal entries of C. For $\lambda \to 1$ this new matrix becomes non-singular. The strategy of the authors in [50] is to increase λ gradually until the resulting matrix is non-singular. The minimization of Equation 2.10 is then executed with the new matrix $\hat{C}$.

The score for the residue pair i and j is based on the l1-norm of the $(q-1)\times(q-1)$ submatrix of Θ that contains the partial correlation coefficients corresponding to the position pair i and j *without* the contribution of the gaps:

$$F_{ij} = \sum_{a,b}^{q-1,q-1} |\Theta_{ij}(a,b)|, \tag{2.15}$$

were we have used tensorial notation for Θ to lighten notation.

In order to arrive at the final score, the authors in [50] apply an average product correction (see Section 2.1.1 for a description). To further reduce the phylogenetic bias, a reweighting scheme equal to the one described in 2.2.2 is applied.

The method performs very well for contact prediction (see Figure 3.1) and is an important ingredient for meta-methods [84].

2.1.3 Bayesian Trees

An interesting method to disentangle direct from indirect couplings that conceptually rather different from the ones presented until now and framed completely in the language of Bayesian inference was presented 2010 by Burger and van Nimwegen [13]. The idea of the method is to assume that the random variables (i.e. the amino acids) have a statistical interdependence that can be modeled by a Bayesian tree [73]. In a Bayesian tree every node representing a variable is assigned set of parent nodes. For the considerations here we also assume that every node has a *single* parent. These parent-children relationships express conditional independencies: A random variable is independent of his ancestors *given the value of his parent.* Identifying the nodes

with columns in a MSA, the probability of a sequence $\underline{s} = (s_1, s_2, ..., s_N)$ can therefore be written as:

$$P(\underline{s}|\pi) = \prod_i P(s_i|\{s_{\pi(i)}\}), \tag{2.16}$$

where $\{s_{\pi(i)}\}$ is the set of parents of node s_i, and the probability of an empirical MSA D given a tree is simply

$$P(D|\pi) = \prod_m^M P(\underline{s}^m|\pi). \tag{2.17}$$

The tree π is of course not known a priori. To arrive at a probability for the MSA, a sum over all trees using a prior should be calculated:

$$P(D) = \sum_\pi P(D|\pi)P(\pi). \tag{2.18}$$

An unnormalized posterior probability for a given edge (ij) representing a direct influence of position i on position j can be constructed using Equation 2.18. This gives a probability of the data $P_{ij}(D)$, *given that the edge i-j exists*. Then, a score for this edge can be defined as:

$$S_{ij} = \frac{P_{ij}(D)}{P(D)} \tag{2.19}$$

The two problems left is to find explicit expressions for the probabilities in Equation 2.18 and, what seems to be more complicated, to evaluate the sum over all possible trees.

The first problem is easily solvable by substituting the relations in Equations 2.16. We adapt the notation of [13] and write $P(D_{ij})$ for the marginal probability to find column i and j together in the MSA, and $P(D_i)$ analogously for only one of them. We further label the root node of the tree with r. This node r does not have a parent, which will be made explicit in the following equations:

$$P(D|\pi) = P(D_r) \prod_{i \neq r} P(D_i|D_{\pi(i)}) \tag{2.20}$$

$$= \left[\prod_i P(D_i)\right] \left[\prod_{i \neq r} \frac{P(D_{i,\pi(i)})}{P(D_i)P(D_{\pi(i)})}\right]. \tag{2.21}$$

Given a tree and the single- and double-site probabilities, we can thus calculate the posterior. These single- and double-site probabilities should follow a multinomial distribution as the sequences are assumed to be drawn identically and independently from the same sequence-distribution:

$$P(D_i|\{w_i(a)\}) = \prod_a w_i(a)^{Mf_i(a)}, \tag{2.22}$$

$$P(D_{ij}|\{w_{ij}(a,b)) = \prod_{a,b} w_{ij}(a,b)^{Mf_{ij}(a,b)} \tag{2.23}$$

where, to avoid confusion and in accordance with [13], the symbol $w_i(a)$ denotes the probability to find amino acid a at position i and the symbol $w_{ij}(a,b)$ the corresponding two-site probability. The authors in [13] use a Dirichlet prior (the conjugate prior to the multinomial distribution) for the probabilities $\{w_i(a)\}$ and $\{w_{ij}(a,b)\}$ with hyperparameters λ and λ' to arrive at explicit expressions for the column probabilities. Given the single and double site frequencies $f_i(a)$ and $f_{ij}(a,b)$ measured in the MSA the following relations can be obtained:

$$P(D_i) = \frac{\Gamma(q\lambda)}{\Gamma(M+q\lambda)} \prod_a \frac{\Gamma\left(Mf_i(a)+\lambda\right)}{\Gamma(\lambda)} \tag{2.24}$$

$$P(D_{ij}) = \frac{\Gamma(q^2\lambda')}{\Gamma(M+q^2\lambda')} \prod_{ab} \frac{\Gamma(Mf_{ij}(a,b)+\lambda')}{\Gamma(\lambda')}, \tag{2.25}$$

where consistency demands $\lambda = q\lambda'$. Together with Equation 2.21 this defines $P(D|\pi)$.

The problem left is to construct a prior over the trees and evaluate the sum in Equation 2.18. For a class of priors called decomposable priors, a method based on a generalized version of *Kirchhoff's Matrix Tree Theorem* [42] developed by Meila and Jaakkola [65] can be applied. In such a *decomposable* prior the prior probability of a tree can be written as the product of probabilities W_{ij} over its edges (i,j):

$$P(\pi) = \prod_{i \neq r} W_{i,\pi(i)} \tag{2.26}$$

Plugging Equations 2.26 and 2.21 in Equation 2.18 we arrive at the final expression for $P(D)$:

$$P(D) = \left[\prod_i P(D_i)\right] \left[\sum_\pi \prod_{i \neq r} R_{i,\pi(i)} W_{i\pi(i)}\right] \tag{2.27}$$

where

$$R_{i\pi(i)} = \frac{P(D_{i,\pi(i)})}{P(D_i)P(D_{\pi(i)})} \tag{2.28}$$

and $W_{i\pi(i)}$ enables one to easily incorporate any prior information one believes to have.

The sum over all trees in Equation 2.27 can now be written as the determinant of any minor of the Laplacian of the matrix M_{ij} [65] with

$$M_{ij} = R_{ij}W_{ij}. \tag{2.29}$$

The computation of the score in Equation 2.19 can therefore be done with the time complexity of calculating the determinant of a $(N-1) \times (N-1)$ matrix, i.e. cubic in N.

The authors in [13] apply as a last ingredient an average product correction to the score (see Section 2.1.1).

The performance of the method in protein contact inference is better than the simple mutual information approach, but the worst one of the more advanced techniques. One of the reasons might be that the underlying model is bad: There is no good reason to assume a tree is a good model for conditional independencies between residues of a protein. An indication that it is indeed a bad one is the fact that using just the maximum-likelihood tree instead of a full Bayesian approach produces results only marginally better than using mutual information [13].

2.2 DCA Approaches

Approaches for the statistical modeling of protein sequence data based on the so-called *Generalized Potts Model* (see Eq. 2.30 below) are collected under the umbrella-term *Direct Coupling Analysis* (DCA). Examples are mean-field DCA (mfDCA) [68], pseudo-likelihood based DCA (plmDCA) or DCA based on Boltzmann Machine learning [87]. This model assigns a statistical weight $P(\underline{s}|\theta)$ to any possible protein sequence $\underline{s}$, dependent on the parameters θ. The general approach is to learn the parameters θ given the data and then use the parameters to calculate an interaction scores between protein residues. This interaction score is then used as a representation of the confidence that two residues are in contact.

The approaches differ in the way they preprocess the data, in the way they calculate the parameters given the data and also in the way they calculate the interaction score. In the following we will first present some general features of the Generalized Potts model and then describe some approaches for protein contact reference based on it in more detail.

2.2.1 The Generalized Potts Model and Maximum Entropy

The Generalized Potts Model We define the *Generalized Potts Model* (GPM) as the discrete exponential family that assigns to a sequence $\underline{s} = (s_1, \ldots, s_N)$ of length N a probability of the form

$$P(\underline{s} \mid J, h) = \frac{1}{Z(J,h)} \exp\left(\sum_{i=1}^{N}\sum_{j=i+1}^{N} J_{ij}(s_i, s_j) + \sum_{i=1}^{N} h_i(s_i)\right), \tag{2.30}$$

where the s_i can take on any value from an alphabet of size q and the $J_{ij}(a, b)$ and $h_i(a)$ are real numbers indexed by the positions i and j and the symbols (amino acids) a and b. Note that this use of the term *Potts Model* deviates from the terminology used traditionally [99]. The normalization constant Z is defined as

$$Z = \sum_{\underline{s}} \exp\left(\sum_{i=1}^{N}\sum_{j=i+1}^{N} J_{ij}(s_i, s_j) + \sum_{i=1}^{N} h_i(s_i)\right). \tag{2.31}$$

Notice that this sum contains q^N terms. With $q = 21$ (20 amino acids and one gap symbol, see Section 1.2) and $N \approx 30$ (for a small protein) these are around 10^{19} terms. This means that an exact and direct calculation of Z is impossible even for small proteins.

Inspired by statistical physics, the exponent of an object like Equation 2.30 is often called (with reversed signs) *Hamiltonian*, or simply *the model*:

$$-H(\underline{s}) = \left(\sum_{i=1}^{N}\sum_{j=i+1}^{N} J_{ij}(s_i, s_j) + \sum_{i=1}^{N} h_i(s_i)\right) \tag{2.32}$$

It defines the probability distribution.

It is often convenient to write Equation 2.30 using Kronecker deltas $\delta_i^a(\underline{s})$ (see e.g. [4]), which are defined to be 1 if $s_i = a$ in $\underline{s}$ and 0 otherwise. Any sequence can then be represented as a binary vector of length $N \cdot q$,

$$\underline{s} = \begin{pmatrix} \delta_1^a(\underline{s}) \\ \delta_1^b(\underline{s}) \\ \cdots \\ \delta_N^q(\underline{s}) \end{pmatrix} \tag{2.33}$$

and Equation 2.30 can be transformed into

$$P(\underline{s} \mid J,h) = \frac{1}{Z(J,h)} \exp\left(\sum_{i=1}^{N}\sum_{j=i+1}^{N}\sum_{a,b} J_{ij}(a,b)\cdot\delta_i^a(\underline{s})\delta_j^b(\underline{s}) + \sum_{i=1}^{N}\sum_{a} h_i(a)\delta_i^a(\underline{s})\right). \tag{2.34}$$

We hope the reader will bear with us for yet another representation that is more general and more convenient to do calculations in, especially in the later Section 3.2 when the model of Equation 2.32 is extended beyond pairwise terms.

With a simple index α for every term, real parameters ξ_α, absorbed signs and arbitrary functions $\phi_\alpha(\underline{s})$ (called potentials) we can rewrite the probability in Equation 2.30 as

$$P(\underline{s} \mid \xi) = \frac{1}{Z(\xi)} \exp\left(\sum_{\alpha=1}^{R} \xi_\alpha \phi_\alpha(\underline{s})\right) \tag{2.35}$$

Obviously, one can go back to Equation 2.34 by assigning one α to every term in Equation 2.34 and substituting the ξ for the J and h, and the Kronecker deltas (or products of them) for the ϕ. The number of parameters is $R = \binom{N}{2}q^2 + Nq$.

Maximum Entropy Derivation The GPM can be derived in the context of the *Maximum Entropy Principle* (MaxEnt) [48]. This principle answers the question

'Which probabilistic model should I choose for my data?'

with

'The model that has the maximal entropy of all models that are coherent with the constraints derived from the data.'.

This can be seen as a generalization of the *Principle of Indifference* [49]. Its core idea is, colloquially speaking, that in the absence of any good reason to do otherwise, one should assign the probabilities in a probability space as even as possible among all possible outcomes.

Mathematically this means to find the probability vector p (where p has q^N entries that assign probabilities for the q^N possible sequences) that maximizes the equation $C(p) + S(p)$, where the term $C(p)$ enforces the constraints and $S(p)$ is the Shannon Entropy.

In order to arrive by this reasoning at the GPM of Equation 2.30 one chooses as constraints the equality of the single-site and double-site frequencies of the model and data.

We define

$$
\begin{aligned}
p_i(a) &= \mathbf{E}\left[\delta_i^a(\underline{s})\right]^p \\
p_{ij}(a,b) &= \mathbf{E}\left[\delta_i^a(\underline{s}) \cdot \delta_j^b(\underline{s})\right]^p \\
f_i(a) &= \mathbf{E}\left[\delta_i^a(\underline{s})\right]^f \\
f_{ij}(a,b) &= \mathbf{E}\left[\delta_i^a(\underline{s}) \cdot \delta_j^b(\underline{s})\right]^f
\end{aligned}
\tag{2.36}
$$

with $E\left[f(\underline{s})\right]^p$ the expectation of function f in the model distribution and $E\left[f(\underline{s})\right]^f$ in the data distribution.

The function determining p can then be written as

$$
\begin{aligned}
P^* = \underset{p}{\operatorname{argmax}}\Bigg[&- \sum_{\underline{s}} P(\underline{s}) \log P(\underline{s}) + \sum_{i<j} \sum_{a,b} \lambda_{ij}(a,b)\left(f_{ij}(a,b) - P_{ij}(a,b)\right) \\
&+ \sum_i \sum_a \lambda_i(a)\left(f_i(a) - P_{ij}(a)\right) + \omega\left(\sum_{\underline{s}} P(\underline{s}) - 1\right)\Bigg],
\end{aligned}
\tag{2.37}
$$

where the first term in the bracket is the Shannon Entropy and the second and third term enforce the equalities of the marginals using the Lagrange multipliers λ that are indexed in the same way as the marginals. The last term enforces the normalization of p by the Lagrange multiplier ω.

After calculating the derivative with respect to the probability $P(\underline{\hat{a}})$ of some fixed but arbitrary state $\underline{\hat{a}}$ it follows immediately that P^* must be of the form of Equation 2.30 with $\lambda_{ij}(a,b) = J_{ij}(a,b)$ and $\lambda_i(a) = h_i(a)$ for all i,j,a and b.

Notice that this tells us only the form of the distribution, but gives no direct equation for the parameters J and h. These must be chosen such that the constraints are satisfied. This means that the model has still be to inferred on the data (see below).

In the case presented here the practical value of this line of reasoning is doubtable. There is no a priori rule on how to choose the constraints, but with a corresponding choice of constraints any model of the form

$$
-H(\underline{s}) = \sum_i h_i(s_i) + \sum_{i<j} J_{ij}(s_i, s_j) + \sum_{i<j<k} K_{ijk}(s_i, s_j, s_k) + \ldots \tag{2.38}
$$

can be derived, with arbitrary interactions of all orders. One needs additional considerations, like the fact that inference for more than 2-body interactions becomes unfeasible because of the large number of parameters and the problem of estimating them from few samples.

Alternatively, one can refer to the general form of the Hamiltonian in Equation 2.38 and see the Model in Equation 2.32 as a hopeful truncation thereof [29].

Inference A common way to find suitable parameters of the Potts Model of Equation 2.30 given the data D is to maximize the posterior [60],

$$P(J, h \mid D) \propto P(D \mid J, h) \cdot P(J, h), \tag{2.39}$$

where the first term on the right hand side is the *likelihood* (read as a function of the parameters) and the second term a *prior*. Setting the prior to a constant and under the assumption that the data consists of independent samples this method is known as *maximum likelihood estimation*. In this case, the likelihood function $\mathcal{L}$ can be written as

$$\mathcal{L} = \prod_{m=1}^{M} P(\underline{s}^m \mid J, h) \tag{2.40}$$

where m is indexing the M sequences in the data.

It is often more convenient to maximize the logarithm of this function. Inserting Equation 2.30 one gets

$$1/M \cdot \log \mathcal{L}(J, h) = \sum_{i<j} \sum_{a,b} J_{ij}(a, b) f_{ij}(a, b) + \sum_{i} \sum_{a} h_i(a) f_i(a) - \log Z, \tag{2.41}$$

and the goal is to find the J and h for which this function is maximal.

Another way to arrive at the same equation is to minimize the Kullback-Leibler distance [60] between the model distribution and the data distribution f,

$$D\left(f \mid\mid p\right) = \sum_{\underline{s}} f(\underline{s}) \log \left(\frac{f(\underline{s})}{P(\underline{s})} \right) = \sum_{\underline{s}} f(\underline{s}) \log f(\underline{s}) - \sum_{\underline{s}} f(\underline{s}) \log P(\underline{s}), \tag{2.42}$$

which leads to Equation 2.41 after inserting Equation 2.30 (up to a term not depending on J and h). Given the aspect of the last term the (negative) log-likelihood is also called *cross entropy* in this context [15].

Notice also that Equation 2.41 still contains the partition function Z and its calculation is therefore not feasible for arbitrary J and h.

Concavity and Gauge Freedom It is easy to show that the log-likelihood is concave since its Hessian (the negative of which is also called the *observed information* [88]) is proportional to the correlation matrix between potentials (notation of Equation 2.35):

$$-\frac{\partial^2 \log(\mathcal{L})}{\partial \xi_\alpha \partial \xi_\beta} \propto E[\phi_\alpha \phi_\beta] - E[\phi_\alpha]E[\phi_\beta] \tag{2.43}$$

Since this is negatively semi-definite, the function is concave.

This does not mean, however, that the maximum of 2.41 is unique. In fact, it is easy to see that any two Hamiltonians H_1 and H_2 for which

$$H_1(\underline{s}) = H_2(\underline{s}) + C \tag{2.44}$$

holds for some constant C will lead to the same probability distribution 2.30 and to the same likelihood 2.41. An easy transformation of the couplings that leaves the probability distribution unaltered is for example

$$J_{ij}(a,b) \to J_{ij}(a,b) + K_{ij} \tag{2.45}$$

with K_{ij} arbitrary.

2.2.2 Mean Field DCA

Mean-Field Direct Coupling Analysis (**mfDCA**), put forward in [68], was the first fast and efficient method to infer the couplings $J_{ij}(a,b)$ in Equation 2.30, given a MSA. The basic idea is a Taylor-expansion around zero couplings similar to the high-temperature expansion (around $\beta = 0$) for the Ising model described in [39].

The starting point is a Legendre-transformation $G(\alpha)$ of the free energy (the logarithm of the partition function defined by Equation 2.31) in combination with the introduction of a perturbation parameter α controlling the strength of the interaction term in the Hamiltonian (notation taken from Section 2.2.1):

$$\mathcal{G}(\alpha) \coloneqq -\ln Z(\alpha) - \sum_{i}^{q}\sum_{a}^{N} P_i(a)h_i(a), \tag{2.46}$$

with

$$\mathcal{Z}(\alpha) = \sum_{a_1,a_2,..,a_N} \exp\left(\alpha \sum_{i=1}^{N}\sum_{j=i+1}^{N} J_{ij}(a_i,a_j) + \sum_{i=1}^{N} h_i(a_i)\right) \tag{2.47}$$

Notice that $\alpha = 0$ corresponds to a system with no interaction and $\alpha = 1$ to the original system with full interaction. By virtue of the Legendre-Transformation the following relations hold:

$$\begin{aligned} h_i(a) &= \frac{\partial \mathcal{G}(\alpha)}{\partial P_i(a)} \\ (C^{-1})_{ij}(a,b) &= \frac{\partial^2 \mathcal{G}(\alpha)}{\partial P_i(a) \partial P_j(b)}, \end{aligned} \tag{2.48}$$

where $C_{ij}(a,b) = P_{ij}(a,b) - P_i(a)P_j(b)$ is the connected correlation matrix of size $Nq \times Nq$ and the last equality can be derived very non-rigorously by

$$\delta_{ia,jb} = \frac{\partial P_{ia}}{\partial P_{ib}} = \sum_{kc} \frac{\partial P_{ia}}{\partial h_{kc}} \cdot \frac{\partial h_{kc}}{\partial P_{jb}} \tag{2.49}$$

and noting that for the Potts Model $\frac{\partial P_{ia}}{\partial h_{jb}} = C_{ij}(a,b)$. Since from deriving the first Equation in 2.48 with respect to P_{jb} one gets $\frac{\partial h_{kc}}{\partial P_{jb}} = \frac{\partial^2 \mathcal{G}(\alpha)}{\partial P_k(c) \partial P_j(b)}$ one can reinterpret Equation 2.49 as a matrix inversion and rewrite it as the second part of Equation 2.48.

As for all matrices we will use the tensor notation $C_{ij}(a,b)$ and matrix notation $C_{ia,jb}$ quite interchangeably. The reason is that with the former a sum over a and b is easy to write, while when dealing with the inversion of the matrix the latter form is preferable.

The mean field solution now consists in expanding the relation for $\mathcal{G}$ in powers of α in first order:

$$\mathcal{G}(\alpha = 1) \approx \mathcal{G}(0) + \left. \frac{\partial \mathcal{G}(\alpha)}{\partial \alpha} \right|_{\alpha=0} \tag{2.50}$$

Together with Equation 2.48 this leads to an explicit expression for the couplings:

$$(C^{-1})_{ia,jb} = -J_{ia,jb} \text{ for } i \neq j \tag{2.51}$$

Plugging into this equation an empirical version of the matrix $\hat{C}_{ij}(a,b) = f_{ij}(a,b) - f(a)f(b)$, derived from the MSA, one can infer the couplings $J_{ij}(a,b)$ by virtue of one single matrix inversion.

The major problem with this solution is that the matrix $C_{ia,jb}$ is surely rank deficient as

$$\sum_b C_{ia,jb} = \sum_b (f_{ij}(a,b) - f_i(a)f_j(b)) = f_i(a) - f_i(a) = 0, \tag{2.52}$$

and the problem as presented by Equation 2.51 therefore ill-defined. A remedy for this problem is to note that the gauge transformation (see Section 2.2.1)

$$J_{ij}(a,b) \to J_{ij}(a,b) - J_{ij}(a,q) - J_{ij}(q,b) + J_{ij}(q,q) \tag{2.53}$$

leaves the probability distribution unchanged. Because in this gauge all entries of the couplings $J_{ia,jb}$ involving the amino acid q are a priori zero, the representation of the matrices C and J can be cut down to a reduced alphabet in which the amino acid indices are $1...q-1$. In this $N(q-1) \times N(q-1)$ representation the trivial rank deficiency in Equation 2.52 vanishes and the matrix at least *could* be invertible.

Unfortunately, in most cases the matrix is still singular. We take as an example the Pfam PF00014 alignment ($N = 53$) [77]. The matrix C has $N(q-1) = 1060$ rows but MATLAB© reports a rank of only 902. A quick computational check reveals that in fact 153 rows and columns have only zeros as entries, corresponding to the fact that certain amino acids are never observed at certain positions (for example Tyrosine at the 53^{th} position).

One method to solve the problem of missing observations is the adding of a pseudocount. The intuition is to extract artificial data from an even distribution over all amino acids at all sequence-positions and blend the resulting correlation matrix into the one resulting from empirical measurements. Therefore, a new correlation matrix $C^{\mathrm{PS}}_{ia,jb}$ is set up using changed frequencies:

$$\begin{aligned} \hat{f}_i(a) &= (1-\lambda) f_i(a) + \frac{\lambda}{q} \\ \hat{f}_{ij}(a,b) &= (1-\lambda) f_{ij}(a,b) + \frac{\lambda}{q^2} \text{ if } i \neq j \\ \hat{f}_{ij}(a,b) &= (1-\lambda) f_{ij}(a,b) + \frac{\lambda \delta_{a,b}}{q} \text{ if } i = j \end{aligned} \tag{2.54}$$

For a λ sufficiently large (but obeying $0 < \lambda < 1$) the resulting matrix $C^{\mathrm{PS}}_{ia,jb}$ is invertible and the original program of inferring the couplings can be executed. Experience shows that a value between 0.3 and 0.5 gives good results. Notice that this is fairly large as $\lambda = 0.5$ means that we give the pseudocount the exact same weight as the actual data.

The task left is how to combine the couplings connected to the positions i and j to a score. The authors in [68] use for this purpose a quantity termed **direct information** (DI_{ij}), the mutual information between site i and j based on a two-site probability model

$$P^d_{ij}(a,b) = \frac{1}{\mathcal{Z}_{ij}} \exp\left(J_{ij}(a,b) + \hat{h}_i(a) + \hat{h}_j(b)\right), \tag{2.55}$$

where $\mathcal{Z}_{ij}$ is calculated in an analogous manner to Equation 2.47 but restricted to positions i and j and the new fields $\hat{h}$ are inferred such that the empiricial single- and double-site frequencies are recovered.

The direct information between sites i and j is then the mutual information calculated in this restricted model:

$$DI_{ij} = \sum_{a,b} P_{ij}^d(a,b) \ln \frac{P_{ij}^d(a,b)}{P_i(a), P_j(b)}. \tag{2.56}$$

Taking DI_{ij} as a score has the advantage that it is independent of a gauge transformation and therefore deserves the appellation of *observable*; the independence of the gauge transformation is desirable in order to ensure that the same probability distributions, which we assume to contain all information extractable, produce the same contact predictions.

Even though disfavored in the original publication [68] there exist rivaling, more intuition-based scores for a pair (i,j). An example is the Frobenius $J_{ij}(a,b)$ for fixed i and j, which is presented in Section 2.2.3. Such measures have the disadvantage that the resulting score is dependent on the gauge, so one has to choose one that seems suitable (one usually chooses the one that minimizes the absolute values of the couplings, in order to explain away as much as possible of the distribution with fields). The upside is that in most numerical experiments they lead to a better prediction quality (a fact that lacks explanation so far). It also seems that these matrix norms get a strong boost in prediction quality by the application of an average product correction term in contrast to the DI score, which is virtually unaffected by it.

The method presented until now addresses the problem of indirect interactions but not the problem of phylogenetic bias. In [68] (and subsequently in many other DCA implementations) a reweighting scheme is introduced: For every sequence S in the alignment the number of similar sequences is determined: The number of sequences with a Hamming distance less than a parameter Θ. The inverse of this number, w_m for the m^{th} sequence in the MSA, is then used as a weight for this sequence in the calculation of the *reweighted* frequencies $\hat{f}$:

$$\hat{f}(\underline{s}) = \frac{1}{M_{eff}} \sum_{m=1}^{M} w_m I\left[\ \underline{s} = \underline{s}^m\ \right], \tag{2.57}$$

where $M_{eff} = \sum_{m=1}^{M} w_m$ in order to ensure normalization of $\hat{f}$.

The method performs significantly better than mutual information, but is outperformed when more precise inference methods are used for the Potts Model, like pseudo-likelihoods (see for example [27] for a comparison with plmDCA).

2.2.3 Pseudolikelihoods

This section presents some technical aspects of the work in [27] and [26]. We also notice that some concepts and ideas were already described in [3]. The context of this section is the same as Section 2.2.2. The general approach is still to infer the GPM of Equation 2.30 on protein sequence data and then extract a score for the existence of a contact between residue i and j from the parameters J_{ij}. The differences between this Section and the foregoing one are found in the method of inference (pseudo-likelihoods vs. mean-field approximation) and the scoring (Direct Information vs. Frobenius Norm). We will therefore focus on these two aspects in this Section. The later Sections 4 and 5 will build on the method presented here.

The Objective Functions The idea of pseudo-likelihoods is to use as an alternative to the full likelihood the likelihood of one variable *given the others* [6].

The probability of the i^{th} position given the other ones in a sequence $\underline{s}$, written in the representation of Equation 2.30, is

$$P(s_i \mid s_{/i}, J, h) = \frac{1}{Z_i(s_{/i})} \exp\left(\sum_{j \neq i} J_{ij}(s_i, s_j) + h_i(s_i)\right) \tag{2.58}$$

with

$$Z_i(s_{/i}) = \sum_a \exp\left(\sum_{j \neq i} J_{ij}(a, s_j) + h_i(a)\right). \tag{2.59}$$

Notice that the conditional probability corresponding to site i depends only on parameters which are connected to site i, or, in the language of factor graphs, on factor nodes that are adjacent to the variable node i.

The pseudo log-likelihood function corresponding to the i^{th} position reads

$$\log \mathcal{PL}_i = \sum_{m=1}^{M} w_m \left[\sum_{j \neq i} J_{ij}(s_i^m, s_j^m) + h_i(s_i^m) - \log Z_i(s_{/i}^m)\right] \tag{2.60}$$

This form is convenient since it can be implemented straight away. The w_m are sequence weights that can be used to give individual sequences more or less weight in the inference process, introduced in the context of pseudolikelihoods for protein contact inference in [27]. This can be used to correct for experimental and phylogenetic biases. They are calculated in the same way as the weights for the reweighted frequencies for mfDCA of Equation 2.57 of Section 2.2.2.

The conditional probability can be written in the more general form of Equation 2.35. We add this here since it makes the calculation of the gradient easy also when higher order terms are introduced, which will be done in 3.2.

$$P(s_i \mid s_{/i}, J, h) = \frac{1}{Z_i(s_{/i})} \exp\left(\sum_{\alpha \in \partial i} \xi_\alpha \phi_\alpha(s_i, s_{/i})\right), \tag{2.61}$$

where we have one summand in the exponent for every summand in the exponent of Equation 2.58.

The pseudo log-likelihood can then be written as

$$\log\left(\mathcal{PL}_i\right) = M \cdot \left(\sum_\alpha \xi_\alpha E\left[\phi_\alpha(s_i, s_{/i})\right]^{\hat{f}} - E\left[\log Z_i(s_{/i})\right]^{\hat{f}}\right) \tag{2.62}$$

and its gradient with respect to a specific ξ_β as

$$\frac{\partial \log\left(\mathcal{PL}_i\right)}{\partial \xi_\beta} = M \cdot \left(E\left[\phi_\beta(s_i, s_{/i})\right]^{\hat{f}} - E\left[\phi_\beta(s_i, s_{/i})\right]^{t}\right) \tag{2.63}$$

$$= \sum_{m=1}^{M} w_m \cdot \left(\phi_\beta(s_i^m, s_{/i}^m) - \sum_{s_i} P(s_i \mid s_{/i}^m, J, h)\phi_\beta(s_i, s_{/i}^m)\right), \tag{2.64}$$

where in the first line t is the distribution defined by $t(\underline{s}) = P(s_i|s_{/i}, J, h)\hat{f}(s_{/i})$ and the second line is reported because it is convenient to be implemented directly.

Written explicitly in terms of the parameters h and J this leads to

$$\frac{\partial \log\left(\mathcal{PL}_i\right)}{\partial h_i(a)} = \sum_{m=1}^{M} w_m \cdot \left(\delta_i^a(\underline{s}^m) - P(a \mid s_{/i}^m, J, h)\right) \tag{2.65}$$

$$\frac{\partial \log\left(\mathcal{PL}_i\right)}{\partial J_{ij}(a,b)} = \sum_{m=1}^{M} w_m \cdot \delta_j^b(\underline{s}^m) \cdot \left(\delta_i^a(\underline{s}^m) - P(a \mid s_{/i}^m, J, h)\right), \tag{2.66}$$

where the definition of the δ can be found in Section 2.2.1.

In [26] the authors introduce a $l2$-regularizer to the objective functions, which is the sum of all squares of all parameters of the model. If this is subtracted from the pseudo log-likelihood, one forces the inference process to make a trade-off between optimizing the bare pseudo log-likelihood and using small absolute parameter values:

$$l2_i(J,h) = \lambda_J \sum_{j \neq i} \sum_{a,b} J_{ij}(a,b)^2 + \lambda_h \sum_a h_i(a)^2 \tag{2.67}$$

The λ_J and λ_h control the regularization strength for the couplings and the fields. That a prior is necessary can be seen from an inspection of example of 2.65. If any of the $E\left[\phi_\beta(s_i, s_{/i})\right]^f$ is 0 (for example if an amino acid is never present at residue i) some parameters diverge in the inference process since the corresponding gradient in Equation 2.63 is never 0 for finite parameter values (since Equation 2.58 cannot assign a zero probability to any amino acid for finite parameter values). Another good reason for the introduction of the prior is that if the structure of the coupling matrix J reflects the contact map, one would assume most elements to be vanishing since the number of contacts in a protein scales roughly with N [91], while the number of parameters scales with N^2.

The final functions $g_i(J,h)$ to be maximized is therefore

$$g_i(J,h) = \mathcal{PL}_i(J,h) - l2_i(J,h) \tag{2.68}$$

These functions are maximized independently in [26]. This poses the problem that one obtains several estimates for the couplings since couplings like $J_{ij}(a,b)$ appear both in g_i and g_j. The easy solution that performs well for protein contact prediction is to take the mean of both values.

The alternative strategy of maximizing

$$G(J,h) = \sum_{i=1}^{N} g_i(J,h) \tag{2.69}$$

is also feasible, but is slower and leads to a virtually identical performance for contact prediction [27]. We therefore use only the former approach in this work.

Scoring and Contact Prediction Instead of using the Direct Information presented in Section 2.2.2, the authors in [26, 27] decide to use the simpler Frobenius norm of the matrix J_{ij} to arrive at a contact score for residues i and j.

$$S_{ij} = \sqrt{\sum_{a=1,b=1}^{q,q} J_{ij}(a,b)^2} \tag{2.70}$$

This may appear conceptually less appealing for two reasons: First, there is to our knowledge no derivation that explains why the Frobenius Norm should be a good measure for how much two residues are co-evolving (even though it is very intuitive

that it should be at least some measure for this). Secondly, the Frobenius Norm is not gauge invariant. Consider the simple gauge transformation.

$$J_{ij}(a,b) \to J_{ij}(a,b) + C_{ij} \tag{2.71}$$

for arbitrary C_{ij}. This leaves the probability distribution itself unaltered, but an arbitrary order of the S_{ij} can be imposed by varying the C_{ij}.

Nonetheless, the use of this score (with an AP correction, see below) has been shown to lead to a better performance in terms of contact prediction be it for parameters inferred with the pseudo-likelihood method [26], be it for parameters inferred within the mean-field approximation [4].

As a last addition, the score is *average product corrected* (APC). This consists in the transformation,

$$S_{ij}^{APC} = S_{ij} - \frac{S_{i\cdot}S_{j\cdot}}{S_{\cdot\cdot}} \tag{2.72}$$

where the dot indicated to take the average over the corresponding index, i.e.

$$S_{i\cdot} = \frac{1}{N-1}\sum_{j\neq i} S_{ij}. \tag{2.73}$$

This is the correction described in Section 2.1.1 and was introduced in [23]as an entropy correction factor for mutual information. It is not clear why this improves prediction when applied to the Frobenius Norm, but it does [27].

2.3 The Application to Protein Structure Prediction

A major goal of bioinformatics is the prediction of protein structure given a sequence. That the amino acid sequence of a protein usually defines its structure is something known as Anfinsen's Principle [2], but the physico-chemical process leading from the sequence to the final structure *in vivo* is generally unknown and probably complicated [28].

Setting aside the question how the protein folds *in vivo*, methods for predicting the final protein structure (or the arrangement of the individual proteins in a complex, the inference of which is called *docking*) from the sequence data are the central field of application for DCA [62,63,71,83]. The inferred residue contacts are used as constraints for the final structure, limiting the space of conformations drastically *a priori*.

Protein structure prediction and docking algorithms usually use a coarse-grained model of the protein, for example a representation of every amino acid as a bead on a string,

several beads on a string or a bead for every heavy atom [96]. For this model of the protein a Hamiltonian is defined and interactions between the parts ideally guide the protein to the native state.

Many such algorithms have been designed, differing for example in the definition of the force fields (for example Amber [92]) or the way the solvent is included.

The Hamiltonian can also be used to include prior information. Between residues that are inferred to be in contact by the analysis of multiple sequence alignments one can define a potential that pushes them together, facilitating the dynamics to find the correct conformation.

An example is the work found in [83], a docking simulation using DCA inferred contacts between proteins as inputs. Here, the authors analyze the bacterial two-component signal transduction system (TCS) consisting of the protein pair Spo0B/Spo0F. Such systems generally work by translating an external stimulus in a phosphorylation of a response regulator by a histidine kinase [86]. The need for cooperation leads to co-evolution in such systems, in the form of correlated residue mutations at the interface between the two proteins.

The large amount of available sequence data for this specific pair of proteins make them especially amenable to protein-protein residue interaction prediction based on DCA (see Section 4 for a detailed description of the concept of protein-protein interaction prediction). After the inference of residue contacts at the interface has been done using the mean-field approximation (see Section 2.2.2), 6 of the inferred contacts are used to constrain the following docking simulations. Given that a crystal structure for the complex is available, the resulting structure can be compared to the inferred one, assessing its quality. The quality can be measured by the *Root Mean Square Deviation* (RMSD), which determines how much the inferred structure deviates on the mean for the experimentally determined one. The authors of [83] conclude that an accuracy similar to experiment is achievable by the docking simulations using DCA inferred contacts as input. The authors used the same method for predicting the complex of another TCS pair, which had no resolved structure at the time but was analyzed experimentally later and with an (excellent) RMSD of 3.3 Å between the predicted and the experimental structure.

Similar works with the same positive evaluation of the utility of DCA or related methods for the structure prediction of proteins are found in [44] (structure prediction for membrane proteins), [62] (structure prediction using an algorithm originating in the field of solving structures with constraints from NMR studies) and [71] (a combination of inferred residue contacts with the popular structure prediction software Rosetta).

It should also be noted that the importance of predicted residue contacts for structure prediction can be seen in the fact that many of the top-performing groups in the biannual tournament *Critical Assessment of Techniques for Protein Structure Prediction* (CASP) now use inferred protein contacts as input.

3 Results: Improving Residue Contact Prediction

3.1 Faster Inference by Gaussian Modeling

This section describes an approach of modeling protein sequence data by using a Gaussian approximation. This work has been published by the authors Baldassi C., Zamparo M., Procaccini A., Zecchina R., Feinauer C., Weigt M. and Pagnani, A. in *PLoS ONE* [4]. Parts of this work will be re-used verbatim in this Section. The journal's copyright policy permits this explicitly.

The notation was adapted to the one used in Section 2.2.1 and the rest of the thesis. Some points that have already been treated in the foregoing Sections are repeated in order to make this section more accessible.

The work in [4] is not only concerned with the prediction of contacts *within* one protein but also with the prediction of contacts *between* proteins. Since this will be the main topic of Section 4, we will not discuss this part of the work here.

Overview Similar to [50], we considered a multivariate Gaussian model in which each variable represents one of the q possible amino-acids at a given site, and aimed in principle at maximizing the likelihood of the resulting probability distribution given the empirically observed data (in particular, given the observed mean and correlation values, computed according to a reweighting procedure presented Section 2.2.2 devised to compensate for the sampling bias). Doing so would yield the parameters for the most probable model which produced the observed data, which in turn would provide a synthetic description of the underlying statistical properties of the protein family under investigation. Unfortunately, however, this is typically infeasible, due to undersampling of the sequence space. A possible approach to overcome this problem, used e.g. in Section 2.2.3 or in [50], is to introduce a sparsity constraint, in order to reduce the number of degrees of freedom of the model. Here, instead, we propose a Bayesian approach, in which a suitable prior is introduced, and the parameter estimation was then performed over the posterior distribution.

A convenient choice for the prior is the normal-inverse-Wishart (NIW), which, being the conjugate prior of the multivariate Gaussian distribution, provides a NIW posterior. Thus, within this choice, the posterior simply is a data-dependent reparametrization of the prior: as a result, the problem is analytically tractable, and the computation of relevant quantities can be implemented efficiently. Furthermore, by choosing the parameters for the prior to be as uninformative as possible (i.e. corresponding to uniformly distributed samples), we obtained an expression for the posterior which, interestingly, can be reconciled with the pseudo-count correction described for the mean-field approach in Section 2.2.2. In the Gaussian framework, the pseudo-count parameter has a natural interpretation as the weight attributed to the prior.

We then estimated the parameters of the model as averages on the posterior distri-

bution, which have a simple analytical expression and can be computed efficiently (in practical terms, the computation amounts to the inversion of a $L(q-1) \times L(q-1)$ matrix), where L is the protein length and $q = 21$ (we slightly deviate from notation in the rest of thesis, were N is reserved for protein length for reasons that will become clear below). This yields an estimate of the strengths of direct interactions between the residues of the alignments, which can be used to predict protein contacts.

Contact prediction between residues relies on the model's inferred interaction strengths (i.e. couplings), which are represented by $q \times q$ matrices; in order to rank all possible interactions, we computed a single score out of each such matrix. As mentioned above, these matrices are numerically identical to those obtained in the mean-field approximation of the discrete (Potts) DCA model. We tested two scoring methods: the so-called direct information (DI), introduced in Section 2.2.2, and the Frobenius norm (FN) as computed in Section 2.2.3. The DI is a measure of the mutual information induced only by the direct couplings, and its expression is model-dependent: in the Gaussian framework it can be computed analytically and yields slightly different results with respect to the Potts model (but with a comparable prediction power, see below). The FN, on the other hand, does not depend on the model, and therefore some of the results which we report here for the contact prediction problem are applicable in the context of the Potts model as well. In our tests, the FN score yielded better results; however, the DI score is gauge-invariant and has a well-defined physical interpretation, and is therefore relevant as a way to assess the predictive power of the model itself.

Data and Methods Input data is given as multiple sequence alignments of protein domains. We directly use MSAs downloaded from the Pfam database version 27.0 [34, 77], which are generated by aligning successively sequences to profile hidden Markov models (HMMs) [25] generated from curated seed alignments. We selected 50 domain families, which were chosen according to the following criteria: *(i)* each family contains at least 2,000 sequences, to provide sufficient statistics for statistical inference; *(ii)* each family has at least one member sequence with an experimentally resolved high-resolution crystal structure available from the Protein Data Bank (PDB) [5], for assessing *a posteriori* the predictive quality of the purely sequence-based inference. The average sequence length of these 50 MSAs is $\langle L \rangle \simeq 173$ residues, the longest sequences are those of family PF00012 whose profile HMM contains 602 residues. The list of included protein domains, together with their PDB structure, is provided in Table 3.1.

Pfam ID	Description	PDB
PF00001	7 transmembrane receptor (rhodopsin family)	1f88, 2rh1
PF00004	ATPase family associated with various cellular activities (AAA)	2p65, 1d2n
PF00006	ATP synthase alpha/beta family, nucleotide-binding domain	2r9v
PF00009	Elongation factor Tu GTP binding domain	1skq, 1xb2
PF00011	Hsp20/alpha crystallin family	2bol

Table 3.1 – continues on next page

Table 3.1 – continued from previous page

Pfam ID	Description	PDB
PF00012	Hsp70 protein	2qxl
PF00013	KH domain	1wvn
PF00014	Kunitz/Bovine pancreatic trypsin inhibitor domain	5pti
PF00016	Ribulose bisphosphate carboxylase large chain, catalytic domain	1svd
PF00017	SH2 domain	1o47
PF00018	SH3 domain	2hda, 1shg
PF00025	ADP-ribosylation factor family	1fzq
PF00026	Eukaryotic aspartyl protease	3er5
PF00027	Cyclic nucleotide-binding domain	3fhi
PF00028	Cadherin domain	2o72
PF00032	Cytochrome b(C-terminal)/b6/petD	1zrt
PF00035	Double-stranded RNA binding motif	1o0w
PF00041	Fibronectin type III domain	1bqu
PF00042	Globin	1cp0
PF00043	Glutathione S-transferase, C-terminal domain	6gsu
PF00044	Glyceraldehyde 3-phosphate dehydrogenase, NAD binding domain	1crw
PF00046	Homeobox domain	2vi6
PF00056	Lactate/malate dehydrogenase, NAD binding domain	1a5z
PF00059	Lectin C-type domain	1lit
PF00064	Neuraminidase	1a4g
PF00069	Protein kinase domain	3fz1
PF00071	Ras family	5p21
PF00072	Response regulator receiver domain	1nxw
PF00073	Picornavirus capsid protein	2r06
PF00075	RNase H	1f21
PF00077	Retroviral aspartyl protease	1a94
PF00078	Reverse transcriptase (RNA-dependent DNA polymerase)	1dlo
PF00079	Serpin (serine protease inhibitor)	1lj5
PF00081	Iron/manganese superoxide dismutases, alpha-hairpin domain	3bfr
PF00082	Subtilase family	1p7v
PF00084	Sushi domain (SCR repeat)	1elv
PF00085	Thioredoxin	3gnj
PF00089	Trypsin	3tgi
PF00091	Tubulin/FtsZ family, GTPase domain	2r75
PF00092	Von Willebrand factor type A domain	1atz
PF00102	Protein-tyrosine phosphatase	1pty
PF00104	Ligand-binding domain of nuclear hormone receptor	1a28
PF00105	Zinc finger, C4 type (two domains)	1gdc
PF00106	Short chain dehydrogenase	1a27
PF00107	Zinc-binding dehydrogenase	1a71

Table 3.1 – continues on next page

Table 3.1 – continued from previous page

Pfam ID	Description	PDB
PF00108	Thiolase, N-terminal domain	3goa
PF00109	Beta-ketoacyl synthase, N-terminal domain	1ox0
PF00111	2Fe-2S iron-sulfur cluster binding domain	1a70
PF00112	Papain family cysteine protease	1o0e
PF00113	Enolase, C-terminal TIM barrel domain	2al2

Table 3.1: 50 Pfam families used in the benchmarks, together with their associated PDB entries. Table taken from [4]

Following [50], we discarded the sequences in which the fraction of gaps was larger then 0.9. However, in [50], an additional pre-processing stage was applied, in which a target sequence is chosen as the one for which prediction of contacts is desired, and all residue positions in the alignment (i.e. columns in the alignment matrix) where the target sequence alignment has gaps are removed. We did not find this pre-processing step to improve the prediction, for either PSICOV or our model, and therefore all results presented in this section do not include this additional filtering.

As written above, the input data were MSAs. An MSA provides a $M \times L$-dimensional array $D = (s_l^m)_{l=1,\dots,L}^{m=1,\dots,M}$: each row contains one of the M aligned homologous protein sequences of length L. Sequence alignments are formed by the $q = 21$ different symbols, which contain 20 amino acids and one gap symbol, see Section 1.2.

Here we consider a modified representation, similar to that used in [50] and similar to the representation found in Equation 2.33 in Section 2.2.1. This turns out to be more practical for the multivariate modeling we are going to propose. The MSA is transformed into a $M \times (Q \cdot L)$-dimensional array $X = (x_i^m)_{i=1,\dots,QL}^{m=1,\dots,M}$ over a binary alphabet $\{0, 1\}$, with $Q = q - 1$. More precisely, each residue position in the original alignment is mapped to Q binary variables, each one associated with one standard amino-acid, taking value one if the amino-acid is present in the alignment, and zero if it is absent; the gap is represented by Q zeros (i.e. no amino-acid is present). Consequently, at most one of the Q variables can be one for a given residue position. For each sequence, the new variables are collected in one row vector, i.e. $x^m_{(l-1)Q+a} = \delta^a_l(\underline{s}^m)$ in the notation of Equation 2.33.

Denoting the row length of X as $N = QL$, we introduce its empirical mean $\overline{x} = (\overline{x}_i)_{i=1,\dots,N}$ and the empirical covariance matrix $C(X, \mu) = \left(C(X, \mu)_{ij}\right)_{i,j=1,\dots,N}$ for given mean $\mu = (\mu_i)_{i=1,\dots,N}$:

$$\overline{x}_i = \frac{1}{M} \sum_{m=1}^{M} x_i^m \,, \tag{3.1}$$

$$C_{ij}(X, \mu) = \frac{1}{M} \sum_{m=1}^{M} (x_i^m - \mu_i)\left(x_j^m - \mu_j\right) \tag{3.2}$$

The empirical covariance is thus $\overline{C} = C(X, \overline{x})$. Note that the entry $\overline{x}_i$, with $i = (l-1)Q + a$, measures the fraction of proteins having amino-acid $a \in \{1, \ldots, Q\}$ at position $l \in \{1, \ldots, L\}$. Similarly, the entry $C_{ij}(X, 0)$ of the correlation matrix, with $i = (k-1)Q + a$ and $j = (l-1)Q + b$, is the fraction of proteins which show simultaneously amino-acid a in position k and b in position l.

Gaussian Modeling We develop our multivariate Gaussian approach by approximating the binary variables as real-valued variables. Even though the former are highly structured, due to the fact that at most one amino-acid is present in each position of each sequence, we will not enforce these constraints on the model. Instead, we shall rely on the fact that the constraint is present by construction in the input data, and that as a consequence we have, for any residue position l and any two states a and b with $a \neq b$:

$$C_{(l-1)Q+a,(l-1)Q+b} = -\overline{x}_{(l-1)Q+a}\overline{x}_{(l-1)Q+b} < 0 \tag{3.3}$$

i.e. two different amino-acids at the same site are anti-correlated. Therefore, we shall let the parameter inference machinery work out suitable couplings between different amino-acid values at the same site, which generate these observed anti-correlations.

The multivariate Gaussian model and the Bayesian inference of its parameters are well-studied subjects in statistics, thus here we only briefly review the main ideas behind our approach, referring to [38] for details. The multivariate Gaussian distribution is parametrized by a mean vector $\mu = (\mu_i)_{i=1,\ldots,N}$ and a covariance matrix $\Sigma = (\Sigma_{ij})_{i,j=1,\ldots,N}$. Its probability density is

$$P(x|\mu, \Sigma) = (2\pi)^{-\frac{N}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right], \tag{3.4}$$

$|\Sigma|$ being the determinant of Σ, and it turns out that the $Q \times Q$ block

$$e_{kl}(a, b) = -\left(\Sigma^{-1}\right)_{(k-1)Q+a,(l-1)Q+b} \tag{3.5}$$

(with $k, l \in \{1, \ldots, L\}$ and $a, b \in \{1, \ldots, Q\}$)

plays the role of the J in the GPM described in Section 2.2.1. Assuming for the moment statistical independence of the M different protein sequences in the MSA, the probability of the data X under the model (i.e. the likelihood) reads

$$P(X|\mu, \Sigma) = \prod_{m=1}^{M} P(x^m|\mu, \Sigma) = (2\pi)^{-\frac{NM}{2}} |\Sigma|^{-\frac{M}{2}} \exp\left[-\frac{M}{2}\mathrm{tr}\left(\Sigma^{-1} C(X, \mu)\right)\right], \tag{3.6}$$

with $C(X, \mu)$ given by Eq. 3.2.

When the empirical covariance $\overline{C}$ is full rank, the likelihood attains its maximum at $\mu = \overline{x}$ and $\Sigma = \overline{C}$, which constitute the parameter estimates within the maximum likelihood approach. However, due to the under-sampling of the sequence space, $\overline{C}$ is typically

rank deficient and this inference method is unfeasible. To estimate proper parameters, we make use of a Bayesian inference method, which needs the introduction of a prior distribution over μ and Σ. The required estimate is then computed as the mean of the resulting posterior, which is the parameter distribution conditioned to the data. As we have already mentioned, a convenient prior is the conjugate prior, which gives a posterior with the same structure as the prior but identified by different parameters accounting for the data contribution. The conjugate prior of the multivariate Gaussian distribution is the normal-inverse-Wishart (NIW) distribution. A NIW prior has the form $p(\mu, \Sigma) = p(\mu|\Sigma)\, p(\Sigma)$, where

$$p(\mu|\Sigma) = (2\pi)^{-\frac{N}{2}} \kappa^{\frac{N}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left[-\frac{\kappa}{2}(\mu - \eta)^T \Sigma^{-1} (\mu - \eta)\right] \tag{3.7}$$

is a multivariate Gaussian distribution on μ with covariance matrix Σ/κ and prior mean $\eta = (\eta_i)_{i=1,\dots,N}$. The parameter κ has the meaning of number of prior measurements. The prior on Σ is the inverse-Wishart distribution

$$p(\Sigma) = \frac{1}{Z} |\Sigma|^{-\frac{\nu+N+1}{2}} \exp\left[-\frac{1}{2}\mathrm{tr}\left(\Lambda\Sigma^{-1}\right)\right], \tag{3.8}$$

where Z is a normalizing constant:

$$Z = 2^{\frac{\nu N}{2}} \pi^{\frac{N(N-1)}{4}} |\Lambda|^{-\frac{\nu}{2}} \prod_{n=1}^{N} \Gamma\left(\frac{\nu + 1 - n}{2}\right). \tag{3.9}$$

The parameters ν and $\Lambda = (\Lambda_{ij})_{i,j=1,\dots,N}$ are the degree of freedom and the scale matrix, respectively, shaping the inverse-Wishart distribution. The condition for this distribution to be integrable is $\nu > N - 1$. The posterior $p(\mu, \Sigma|X)$, proportional to $P(X|\mu, \Sigma) \cdot p(\mu, \Sigma)$, is still a NIW distribution, as one can easily verify starting from Eqs. 3.6, 3.7 and 3.8. The posterior distribution $p(\mu, \Sigma|X)$ is characterized by parameters κ', η', ν', and Λ' given by the formulas

$$\begin{cases} \kappa' = \kappa + M, \\ \eta' = \dfrac{\kappa}{\kappa + M}\eta + \dfrac{M}{\kappa + M}\overline{x}, \\ \nu' = \nu + M, \\ \Lambda' = \Lambda + M\overline{C} + \dfrac{\kappa M}{\kappa + M}(\overline{x} - \eta)(\overline{x} - \eta)^T. \end{cases} \tag{3.10}$$

The mean values of μ and Σ under the NIW prior are η and $\Lambda/(\nu - N - 1)$, and, similarly, their expected values under the NIW posterior are η' and $\Lambda'/(\nu' - N - 1)$, respectively. Our estimations of the mean vector and the covariance matrix, that with a slight abuse of notation we shall still denote by μ and Σ for the sake of simplicity, are thus

$$\mu = \eta' = \frac{\kappa}{\kappa + M}\eta + \frac{M}{\kappa + M}\overline{x} \tag{3.11}$$

and

$$\Sigma = \frac{\Lambda'}{\nu' - N - 1} = \frac{\Lambda + M\overline{C} + \frac{kM}{k+M}\left(\overline{x} - \eta\right)^T\left(\overline{x} - \eta\right)}{\nu + M - N - 1}. \tag{3.12}$$

The NIW posterior is maximum at $\mu = \eta'$ and $\Sigma = \Lambda'/(\nu' + N + 1)$, with the consequence that the *maximum a posteriori* estimation would provide the same estimate of μ and an estimate of Σ that only differs from the previous one by a scale factor.

As a first attempt of protein contact prediction by means of the present model, we choose η and Λ to be as uninformative as possible. In particular, since $U = \Lambda/\left(\nu - N - 1\right)$ is the prior estimate of Σ, it is natural to set $\eta = \left(\eta_i\right)_{i=1,\ldots,N}$ and $U = \left(U_{ij}\right)_{i,j=1,\ldots,N}$ to the mean and the covariance matrix of uniformly distributed samples, which is easily obtained from Eqs. 3.1 and 3.2: therefore, we set $\eta_i = 1/\left(Q+1\right)$ for any i, and U to a block-matrix composed of $L \times L$ blocks of size $Q \times Q$ each, where the out-of-diagonal blocks are uniformly 0:

$$U_{(k-1)Q+a,(l-1)Q+b} = \frac{\delta\left(k,l\right)}{Q+1}\left(\delta\left(a,b\right) - \frac{1}{Q+1}\right), \tag{3.13}$$

where $k, l \in \{1, \ldots, L\}$ and $a, b \in \{1, \ldots, Q\}$, and δ is the Kronecker's symbol. Moreover, we choose $\nu = N + \kappa + 1$ in order to reconcile Eq. 3.12 with the pseudo-count-corrected covariance matrix of [68] with pseudo-count parameter λ. Indeed, identifying λ with $\kappa/\left(\kappa + M\right)$, this instance allows us to recast the estimate of Σ as

$$\Sigma = \lambda U + \left(1 - \lambda\right)\overline{C} + \lambda\left(1 - \lambda\right)\left(\overline{x} - \eta\right)^T\left(\overline{x} - \eta\right) \tag{3.14}$$

and $J = \Sigma^{-1}$ becomes the same as in the mean-field Potts model. Manifestly from here, the effect of the prior is enhanced by values of λ close to 1 while it is negligible when λ approaches 0. Interestingly, the Gaussian framework provides an interpretation of the pseudo-count correction as introduced in Section 2.2.2 in terms of a prior distribution, which may allow improving the inference issue by exploiting more informative prior choices.

Reweighting and Scoring In order to remove phylogenetic and experimental bias, the reweighting scheme described in Section 2.2.2 was used.

For scoring we tested both Direct Information and the Frobenius Norm described in Section 2.2.2 resp. 2.2.3, with Σ^{-1} playing the role of the couplings matrix in the GPM model described in Section 2.2.1. It was found that the Frobenius Norm performed better in predicting protein contacts and that for both scoring methods it was advantageous to use the Average Product Correction described in Section 2.2.3.

The comparison of several competing methods and the two scoring schemes can be found in Figure 3.1. We can summarize the information there that the method outperforms the original mfDCA implementation [68] as well as PSICOV [50]. After around 10 predictions, however, the performance of plmDCA [27] becomes best of all

tested method. This is not surprising given the coarse approximations that have been made in the Gaussian Modeling.

Another important aspect when interpreting the results is the running time. Table 3.2 lists the running time for several methods on proteins of varying length and sequence counts. It can be seen that GaussDCA is several orders of magnitude faster than plmDCA, at comparable (albeit slightly worse) predictive performance.

	PF00014	PF00025	PF00026	PF00078
N	53	175	317	214
M	4915	5460	4762	172360
Gaussian DCA (parallel)	0.7	5.3	16.3	534.8
Gaussian DCA (non-parallel)	1.7	12.7	52.1	3583.4
PSICOV	11.7	1141.9	5442.7	10965.1
plmDCA	433.2	6980.7	37364.8	303331.0

Table 3.2: Running times in seconds for a representative sample of proteins with varying length (N) and sequences in alignment (M), using different algorithms. Since the Gaussian DCA code is parallelized, we show two series of results, one in which we used 8 cores and one in which we forced the code to run on a single core, for the sake of comparing with the non-parallel code of PSICOV and plmDCA. These benchmarks were taken on a 48-core cluster of 2100.130 MHz AMD Opteron™6172 processors running Linux 3.5.0; PSICOV version 1.11 was used, compiled with gcc 4.7.2 at `-O3` optimization level; plmDCA was run with MATLAB®version r2011b. Gaussian DCA timings shown are taken using the Julia version of the code, using Julia version 0.2. Table taken from [4]

Summary of the residue contact prediction steps

To summarize the previous sections, here we list the steps which are taken in order to get from a MSA to the contact prediction:

- clean the MSA by removing inserts and keeping only matched amino acids and deletions;
- remove the sequences for which 90% or more of the entries are gaps;
- assign a weight to each sequence, and compute the reweighted frequency counts $\overline{C}$ and $\overline{x}$ (see Eqs. 3.1 and 3.2, and Supplementary Materials);
- estimate the correlation matrix Σ by means of Eq. 3.14;
- compute Σ^{-1}, and divide it in $Q \times Q$ blocks e_{kl} (see Eq. 3.5);
- for each pair $1 \leq k, l \leq L$, compute a score (DI or FN) from e_{kl}, thus obtaining an $L \times L$ symmetric matrix S (with zero diagonal);

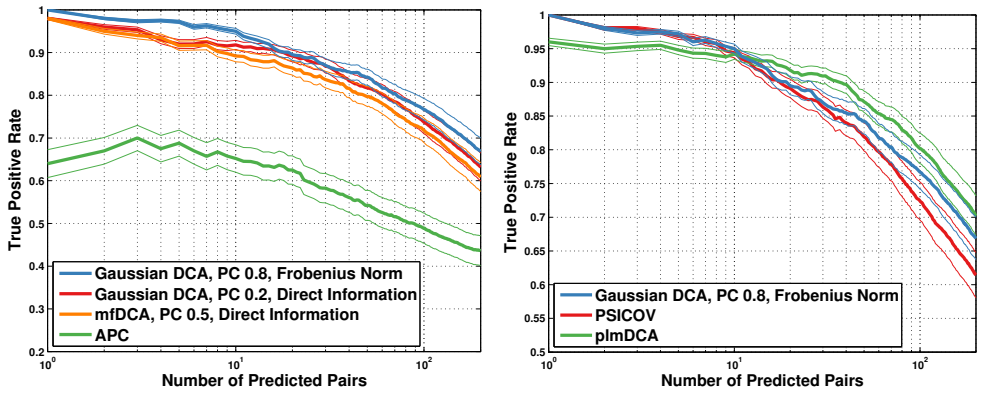


Figure 3.1: True positive rates in the top n predictions, were n is indicated by the x-axis. PC is the pseudo-count parameter. The blue curves are the best performance we were able to generate within the Gaussian Approximation. APC is Mutual Information with the Average Product Correction as described in [23] and PSICOV is the method described in Section 2.1.2. Figure and caption taken from [4]

- apply APC to the score matrix (i.e. subtract to each entry S_{kl} the product of the average score over k and the average score over l, divided by the overall score average – the averages are computed excluding the diagonal), and obtain an adjusted score matrix S_{kl}^{APC};
- rank all pairs $1 \leq k < l \leq L$, with $l - k > 4$, in descending order according to S_{kl}^{APC}.

3.2 Improving Contact Prediction by Modeling Gap Stretches

This section is a synopsis of the work on the improvement of contact prediction found in Reference [29]. Here, the authors describe three aspects of contact prediction and ways to improve them, namely the *data*, the *model* and the *inference method*. The latter two points are the most interesting for us here since they deal with modeling and inference while the first point is more a problem of pure bioinformatics.

Gaps in MSAs are the result of not finding a corresponding amino acid for an alignment column, see Section 1.2. Their average distribution along the positions of an MSA and the probability to find several of them consecutively is markedly different from other amino acids. In Figure 3.2 the upper half shows the distribution of gap and non-gap symbols in the PFAM PF00014 alignment. One notices that gaps are more frequent at the borders of alignments. The lower half shows the distribution of stretches of repeated symbols of a given length. Gaps are more likely to appear in longer stretches

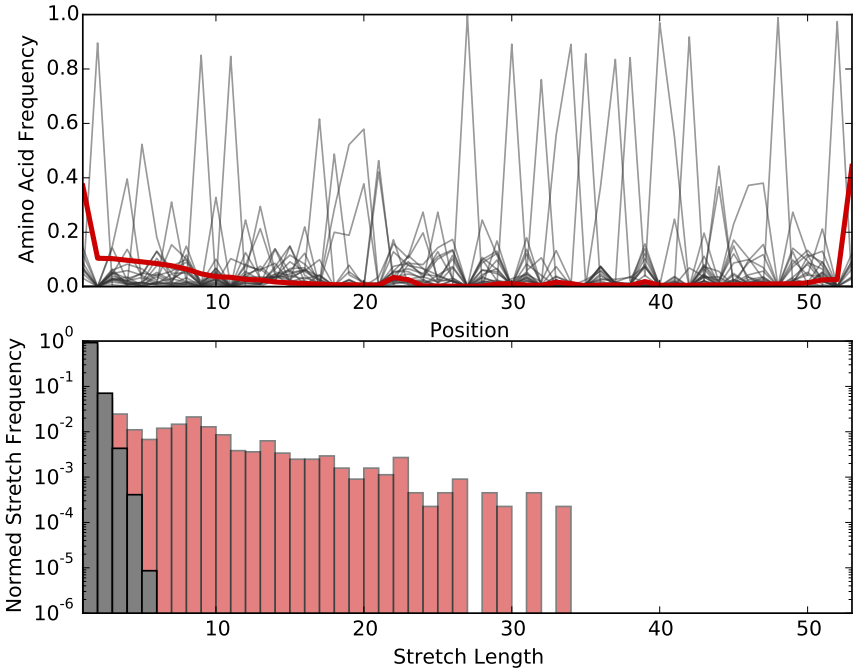


Figure 3.2: **Upper half:** Distribution of gap (red) and non-gap (grey) symbols along the PFAM PF00014-alignment **Lower half:** The distribution of stretches of the same symbol for gap (red) and non-gap (grey) symbols. The two histograms have been normalized to have (both) an area of 1

and for example the probability to find a stretch of length 3 is not markedly different from the probability to find a stretch of length 10.

The different statistics for gaps are a potential problem for DCA approaches using the pairwise model in Equation 2.30. Long stretches of symbols would be rather uncharacteristic for such a model to produce and it seems that higher order terms are necessary to reproduce this specific behavior. This is interesting since one would expect that adding appropriate higher order terms to the model might also lead to a better inference of the pairwise terms and therefore possibly to a higher performance in the inference of protein contacts. Evidence that gaps might indeed be a reason for high ranking false positives can be found in Figure 3.3. Here the upper-left parts of the panels show true and false positives in the first few predictions for two different proteins. Grey dots represent true contacts, green dots true positives and red dots false positives. One notices in both proteins an accumulation of false positives at the C-terminus and N-terminus, where gaps in the alignment are especially frequent.

This might for example be due to strong couplings representing spurious interactions between gaps.

We therefore decided to add parameters for gap stretches to the model [29]. To this end, we introduce indicator functions $I_i^l(\underline{s})$ that are defined to be 1 if in sequence $\underline{s}$ a gap stretch of length l begins at residue i. Notice that these indicator functions are the same objects as the Kronecker deltas defined for Equation 2.33. We recall the original Potts-Hamiltonian of Equation 2.32,

$$-H_{Potts}(\underline{s}) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} J_{ij}(s_i, s_j) + \sum_{i=1}^{N} h_i(s_i) \qquad \text{(2.32 revisited)}$$

and add a new term H_{Gap} taking into account gap stretches

$$H_{Gap}(\underline{s}) = -\sum_{l=1}^{L} \sum_{i=1}^{N-l+1} \xi_i^l \cdot I_i^l[\underline{s}], \tag{3.15}$$

where N is the length of the protein and L the longest gap stretch we would like to model. Generally, one can set L to the longest gap stretch found in the alignment. We notice that the final model consisting of the pairwise terms and the gap terms,

$$H = H_{Potts} + H_{Gap}, \tag{3.16}$$

is not expected to have much more parameters than H_{Potts} only, since the additional term scales roughly like NL, while the number of parameters in H_{Potts} scales like $\frac{1}{2}q^2N^2$ in leading order.

In [29] the inference procedure of choice is the pseudo-likelihood method presented in Section 2.2.3. This is a convenient choice since the method has already been shown to perform excellently in the context of protein contact prediction [27]. Also, it has the major advantage that the inclusion of the term 3.15 is not difficult with respect to the inference procedure, while for example in the mean-field approach presented in Section 2.2.2 it is not trivial to include such a term.

The full pseudo log-likelihood function to be minimized, including the $l2$-regularization (see Section 2.2.3, reads:

$$\begin{aligned}\log \mathcal{PL}_i = \sum_{m=1}^{M} w_m \left[\sum_{j \neq i} J_{ij}(s_i^m, s_j^m) + h_i(s_i^m) + \sum_{l=1}^{L} \sum_{j} \xi_j^l \cdot I_j^l[\underline{s}^m] - \log Z_i(s_{/i}^m) \right] \\ - \lambda_J \sum_{j \neq i} \sum_{a,b} J_{ij}(a,b)^2 - \lambda_h \sum_{a} h_i(a)^2 - \lambda_\xi \sum_{l=1}^{L} \sum_{j} \left(\xi_j^l\right)^2 \end{aligned} \tag{3.17}$$

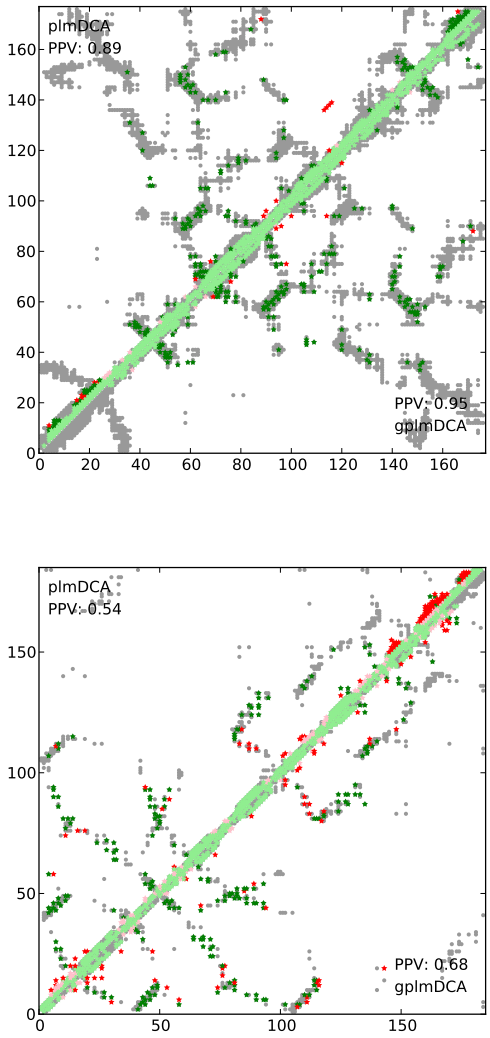


Figure 3.3: Influence of gap parameters on the performance of predicting contacts in 1JFU (left) and 1ATZ (right). **Grey**: True contacts according to the PDB structures. **Pale green**: Contacts predicted with distance less than 5 along the chain. **Dark green**: Contacts predicted with distance larger/equal than 5 on the chain. **Upper Triangles**: Predictions without gap parameters. **Lower Triangles**: Prediction with triangles. **Figure taken from [29]**

The summation over the residue-index j of the gap-parameters ξ was not written explicitly in order to lighten notation. Since we maximize the log pseudo-likelihood independently for every position i, we want to have a dependence only on parameters ξ_j^l such that $i-l+1 \leq j \leq i$, i.e. parameters that depend on position i. Excluding $j < 1$ and $j+l-1 > L$ we therefore have to sum from $max(1, i-l+1)$ to $min(N-l+1, i)$.

The parameter L is the length of the longest gap stretch that one would like to model. This can be set to the longest gap-stretch that is found in the alignment. The parameters λ_J,λ_h and λ_ξ are free parameters. The first two were fixed on the same value as in [26], while λ_ξ was set to 0.001 after some preliminary tests.

The derivatives with respect to the gap parameters are easily done using the general relation 2.63.

$$\frac{\partial \log\left(\mathcal{PL}_i\right)}{\partial \xi_j^l} = \sum_{m=1}^{M} w_m \cdot \left(I_j^l[s_i^m, s_{/i}^m] - \sum_{s_i} P(s_i \mid s_{/i}^m, J, h) I_j^l[s_i, s_{/i}^m] \right). \tag{3.18}$$

The gradient descent can be done in what way preferred. For this work, the same algorithms as in [26] were used. The resulting ξ can be discarded since their only purpose is to improve the inference of the couplings.

The computation of residue interaction-scores was done the same way as described in Section 2.2.3 and the lower triangles of Figure 3.3 shows some results for the same two proteins for which the negative effects of gaps was discussed above. The introduction of the gap parameters reduces strongly the number of errors made at the end of the proteins (compare the upper triangles, which are the predictions without gap parameters). Since this is were gaps are found mostly in the alignments, we can speculate that the effect is in fact due to the removal of the influence of these gaps on the couplings.

In order to get a more quantitative measure for the improvement in contact prediction when using gap parameters, we run the algorithm on a test set of 729 proteins [29]. The results can be seen in 3.4. The performance gain using gap parameters is especially significant when using alignments created with HHBlits [79]. It is not surprising that the effects of changes in the model depend on the way the input-data was created, since this will influence the statistics of gaps and other amino acids strongly. It is in fact a major result and of [29] that the way the input is generated will have a large effect on predictive performance and should be optimized. These results and the performance in PFAM alignments, used e.g. in [68] and [27], are summarized in Figure 3.5.

Also, as is discussed in the supporting information in [29], the measured performance depends on the way a protein contact is defined. The gap parameters seem to have an especially pronounced effect if the distance between two protein residues is defined as the distance between their Cβ atoms and a contact is defined as two residue with a distance < 8Å. The advantage of this more stringent criterion is that one would expect less meaningless contacts in the resulting contact map, which will certainly

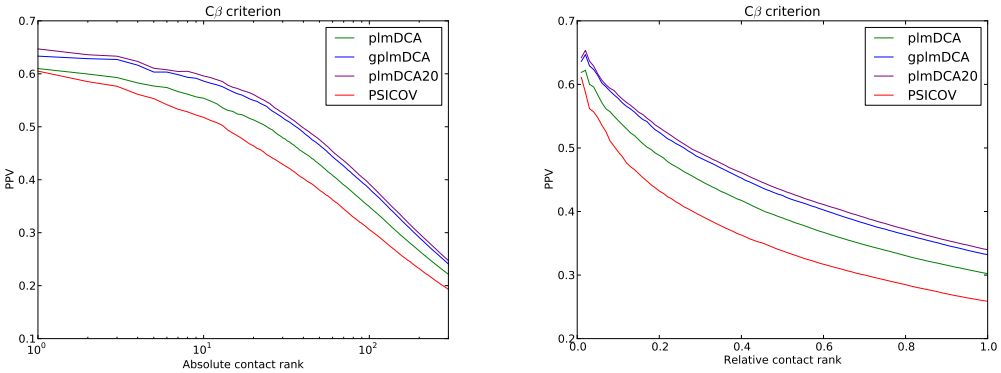


Figure 3.4: Predictive Performance on the *Main Test Set* described in [29] (a set 729 alignments used in prior studies). **Left Panel:** Absolute contact rank. The y-axis shows the mean fraction of true contacts in the first n predictions, where n is indicated by the x-axis. **Right panel:** Relative contact rank. The y-axis shows the mean fraction of true positives in the first n predictions, where n depends on the protein length N and $n = N \cdot f$ where f is indicated by the x-axis. Figure taken from [29]. The methods are as described in the text, PSICOV is described in Section 2.1.2.

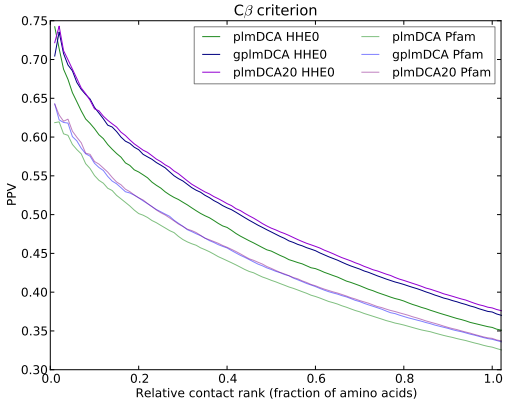


Figure 3.5: Predictive Performance on the *Main Test Set* described in [29] (a set 729 alignments used in prior studies). Relative contact rank. The y-axis shows the mean fraction of true positives in the first n predictions, where n depends on the protein length N and $n = N \cdot f$ where f is indicated by the x-axis. Shown are result for HHBlits alignments (HHE0, full colors) and Pfam alignments (pale colors). Figure taken from [29]. The methods are as described in the text

influence the predictive performance and also the relative improvement when using gap parameters.

Another surprising result of [29] is that a comparable improvement in predictive performance can be obtained by excluding couplings corresponding to gaps from the calculation of the final score. This means to use instead of Equation 2.70 for the calculation of the Frobenius Norm the alternative

$$S_{ij}^{20} = \sqrt{\sum_{a=1,b=1}^{q-1,q-1} J_{ij}(a,b)^2}. \tag{3.19}$$

Using plmDCA with this modification has been termed plmDCA20 in [29]. This should intuitively remove high ranking false positives due to large gap couplings and indeed has a similar effect on the predictive performance as the introduction of gap parameters. Further evidence for equivalence in effect is the fact that using gap parameters *and* the alternative score does not lead to further improvement. This might make the gap parameters look less interesting since it is considerably less cumbersome to implement 3.19 than higher-order terms. The interesting point of [29] is, however, that the inclusion of higher-order terms can lead to an improvement in predictive performance. Since this leads immediately to the question which *other* higher-order terms, besides the ones corresponding to gap stretches, should be introduced, this opens a whole new direction of research.

4 Results: Inference of Protein-Protein Interaction Networks

This section presents the main results of [31] by the authors *Feinauer C, Szurmant H., Weigt M. and Pagnani A.*. Text and Figures are often reused verbatim but many references to the introductory sections of this thesis and other alterations to the text are introduced, in order to streamline the content with the rest of the thesis. The journal's copyright policy permits this explicitly.

Some key points already discussed in the foregoing sections are repeated, such that the reader interested mainly in this section can follow it without having to read the rest of the thesis. Since this section contains many tables, we added an extra section to accommodate them in order to improve the readability.

4.1 Overview

Proteins are the major work horses of the cell. Being part of all essential biological processes, they have catalytic, structural, transport, regulatory and many other functions. Few proteins exert their function in isolation. Rather, most proteins take part in concerted physical interactions with other proteins, forming networks of protein-protein interactions (PPI). Unveiling the PPI organization is one of the most formidable tasks in systems biology today. High-throughput experimental technologies, applied for example in large-scale yeast two-hybrid [47] analysis and in protein affinity mass-spectrometry studies [43], allowed a first partial glance at the complexity of organism-wide PPI networks. However, the reliability of these methods remains problematic due to their high false-positive and false-negative rates [10].

Given the fast growth of biological sequence databases, it is tempting to design computational techniques for identifying protein-protein interactions [41]. Prominent techniques to date include: the genomic co-localization of genes [19, 37] (with bacterial operons as a prominent example), the Rosetta-stone method [61] (which assumes that proteins fused in one species may interact also in others), phylogenetic profiling [74] (which searches for the correlated presence and absence of homologs across species), and similarities between phylogenetic trees of orthologous proteins [52, 72, 90, 100]. Despite the success of all these methods, their sensitivity is limited due to the analysis of coarse global proxies for protein-protein interaction. An approach that exploits more efficiently the large amount of information stored in multiple sequence alignments (MSA) seems therefore promising.

When applied to two interacting protein families, DCA and related methods are able to detect inter-protein contacts [45,70,93] and thereby to guide protein complex assembly [18, 83]. This is notable since contact networks in protein complexes are strongly modular: There are many more intra-protein contacts than inter-protein contacts.

Moreover, DCA helps to shed light on the sequence-based mechanisms of PPI specificity [12, 14, 76].

Here we address an important question: Is the strength of inter-protein residue-residue co-evolution sufficient to *discriminate interacting from non-interacting pairs of protein families*, i.e. to infer PPI networks from sequence information? A positive answer would lever the applicability of these statistical methods from structural biology (residue contact map inference) to systems biology (PPI network inference). An obvious problem in this context is the sparsity of PPI networks, illustrated by the bacterial ribosomal subunits used in the following, see Figures 4.1 and 4.2: The small subunit contains 20 proteins and 21 protein-protein interfaces (11% of all 190 possible pairs). In the large subunit, 29 proteins form 29 interfaces (7% of all 406 pairs). We see that while the number of potential PPI between N proteins is $\binom{N}{2}$, the number of real PPI grows only linearly as $\mathcal{O}(N)$. Furthermore, the number of potentially co-evolving residue-residue contacts across interfaces is much smaller than the number of intra-protein contacts. In the case of ribosomes, only 5.8% of all contacts in the small subunit are inter-protein contacts. In the large subunit this fraction drops down to 4.5%. So the larger the number of proteins, the more our problem resembles the famous search of a needle in a haystack. The noise present in the large number of non-interacting protein family pairs might exceed the co-evolutionary signal of interacting pairs.

It should also be mentioned that the ribosomal structure relies on the existence of ribosomal RNA, which is not included in our analysis. We therefore expect many of the small PPI interfaces to be of little importance for the ribosomal stability and that only large interfaces constrain sequence evolution and thus become detectable by co-evolutionary studies.

Ribosomal proteins and their interactions are essential and thus conserved across all bacteria, and it appears reasonable to wonder whether this makes them a specialized example of a protein complex more amenable to co-evolutionary bias. As a second and smaller interaction network, we therefore considered the enzymes of the tryptophan biosynthesis pathway comprising a set of seven proteins in which only two pairs are known to interact (PDB-ID 1qdl for the TrpE-TrpG complex [58] and 1k7f for the TrpA-TrpB complex [95]). Also here the PPI network is very sparse; most pairs are not known to interact, but might show some degree of coordinated evolution due to the fact that in many organisms these genes show a common spatial co-localization in a single operon and also due to a number of gene fusion events, cf. the discussion below. While widespread, the tryptophan biosynthesis pathway is not essential for viability when environmental tryptophan is present.

In this Section we report the excellent performance of DCA in the prediction of protein-protein interaction partners in the systems tested. In a first step, we analyze the performance on data from an artificial model. This allows for a systematic analysis of the performance of different approaches and of the influence of the number of sequences in the alignment. With this artificial data set we are able to establish a lower-bound on the number of sequences that would make our predictions on the PPI scale completely

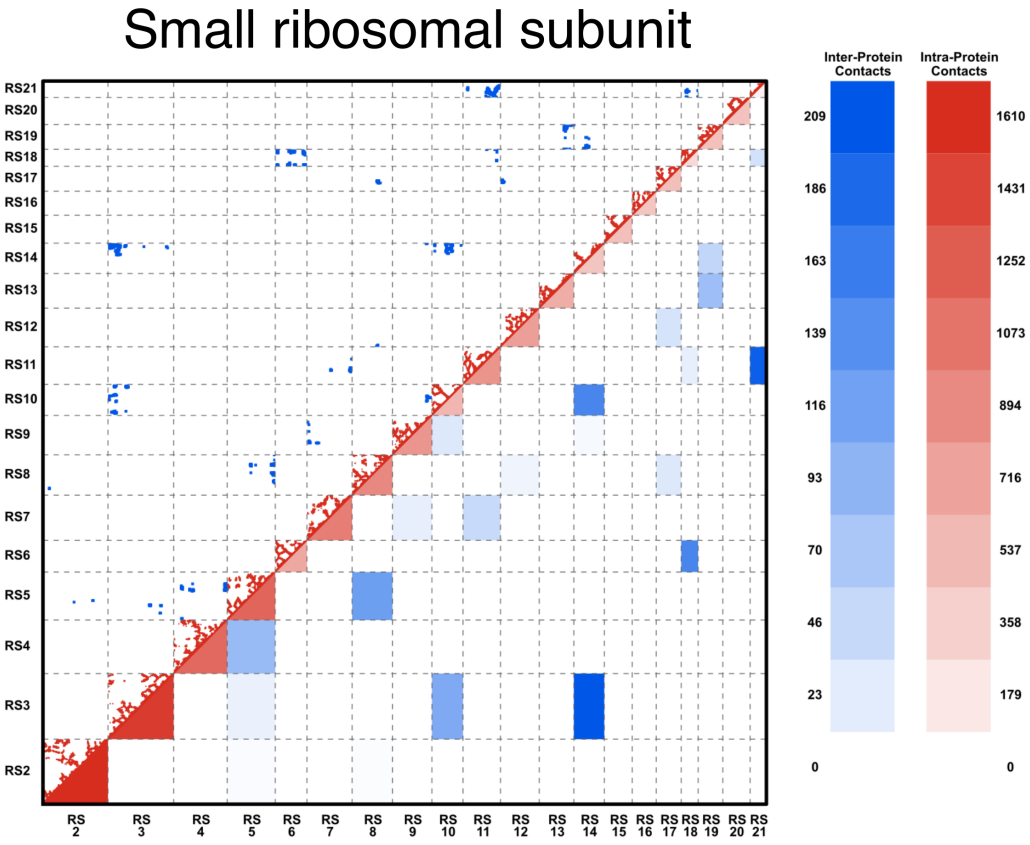


Figure 4.1: **Contact map and protein-protein interaction network of the small ribosomal subunit.** The contact map and the protein-protein interaction network for the small ribosomal subunit (proteins only), using a distance cutoff of 8Å between heavy atoms. The upper diagonal part shows the contact map, with red dots indicating intra-protein contacts, and blue dots inter-protein contacts. The lower triangular part shows the coarse graining into the corresponding protein-protein interaction networks, with the color levels indicating the number of intra- resp. inter-protein contacts, cf. the scales. The sparse character of both the contact network and the interaction network is clearly visible. Figure taken from [31].

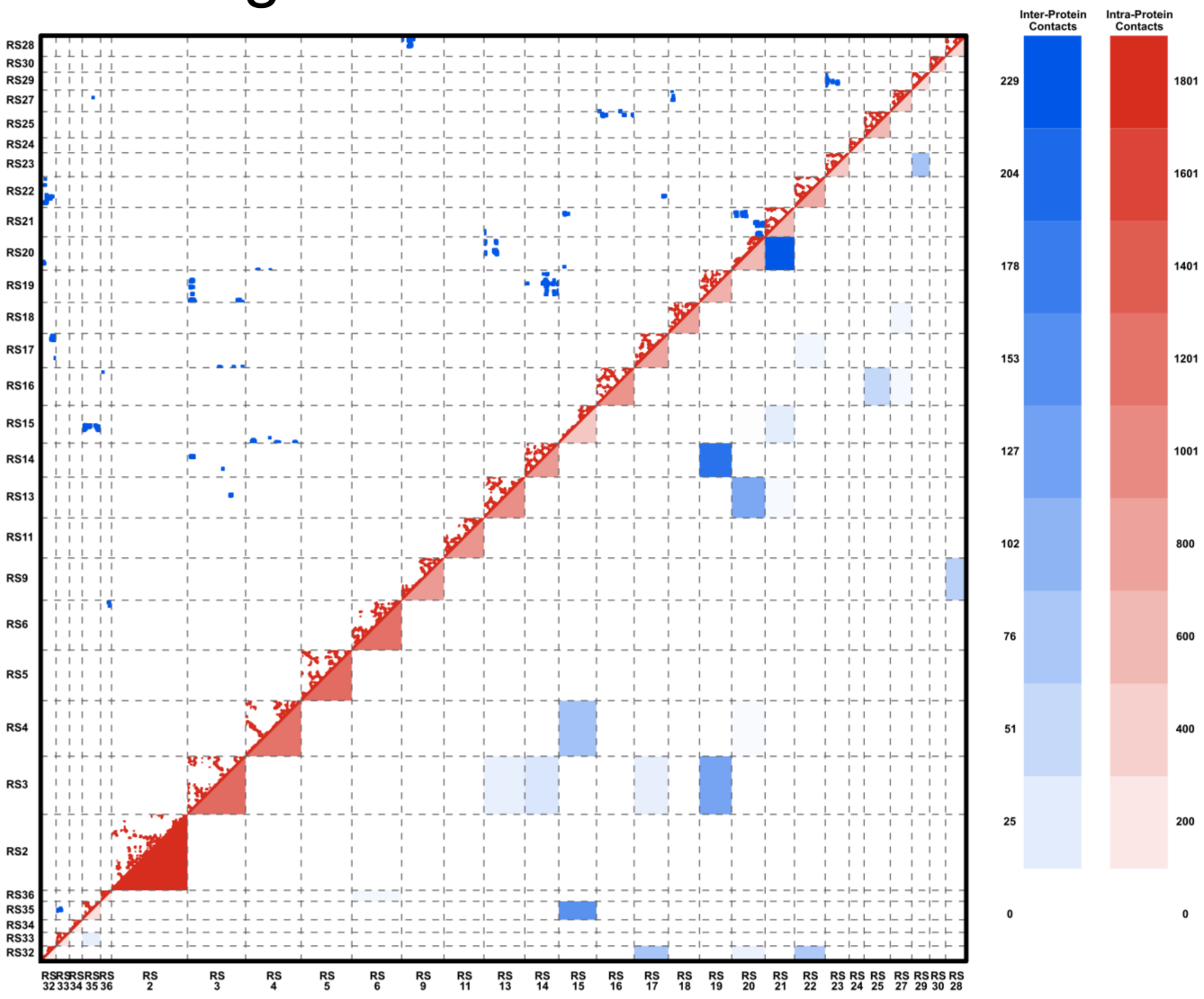


Figure 4.2: **Contact map and protein-protein interaction network of the large ribosomal subunit.** The contact map and the protein-protein interaction network for the large ribosomal subunit (proteins only), using a distance cutoff of 8Å between heavy atoms. The upper and lower parts show the same information as Figure 4.1. Figure taken from [31].

accurate if the generating model was the same model we use for inference. Given the growth-rate of current protein sequence databases (notably UniProt [16]), we expect that such a lower bound could be met in few years. In a second step, we apply the method to the proteins of the bacterial ribosome and to the proteins of the trp operon, and show that the results obtained for simulated data translate well to the biological sequences of this test-set.

The general goal of the present work is to analyze each of the $\binom{N}{2}$ possible pairs of multiple sequence alignments from a given set of N single-protein family alignments, and to extract a pairwise score that measures the co-evolution between the proteins in the alignments. A high co-evolutionary score is then taken as a proxy for interaction. In the spirit of [29] we begin by describing consecutively the *data generation and matching*, the *model* used for analyzing data and the *inference and scoring* mechanism.

4.2 Data Extraction and Matching Paralogs

4.2.1 Data Extraction for Real Proteins

The input data is given by N multiple sequence alignments D_p consisting of M_p sequences of length L_p for every protein family p. These alignments are extracted from UniProt [16] using standard bioinformatics tools, in particular Mafft [54] and HMMer [35].

For all proteins of the small ribosomal subunit (SRU) and the large ribosomal subunit (LRU) the sequence names were extracted from the corresponding PFAM alignments [34]. Using these names, the following procedure was used to create the alignments for the single proteins:

1. Extract sequences corresponding to names from Uniprot [16]
2. Run MAFFT [55] on them using `mafft --anysymbol --auto`
3. Remove columns from the alignment that contain more than 80% gaps
4. Create an Hidden Markov Model (HMM) using `hmmbuild` from the hmmer suite [35]
5. Search Uniprot using `hmmsearch` [35]
6. Remove inserts
7. If there exist in one species two or more sequences that are more than 95% identical, remove all but one.

The number of sequences for the single files can be found in Table 4.1

The alignments for the proteins of the Trp Operon where constructed analogously with some modifications to ensure that only full-length sequences where extracted. Also,

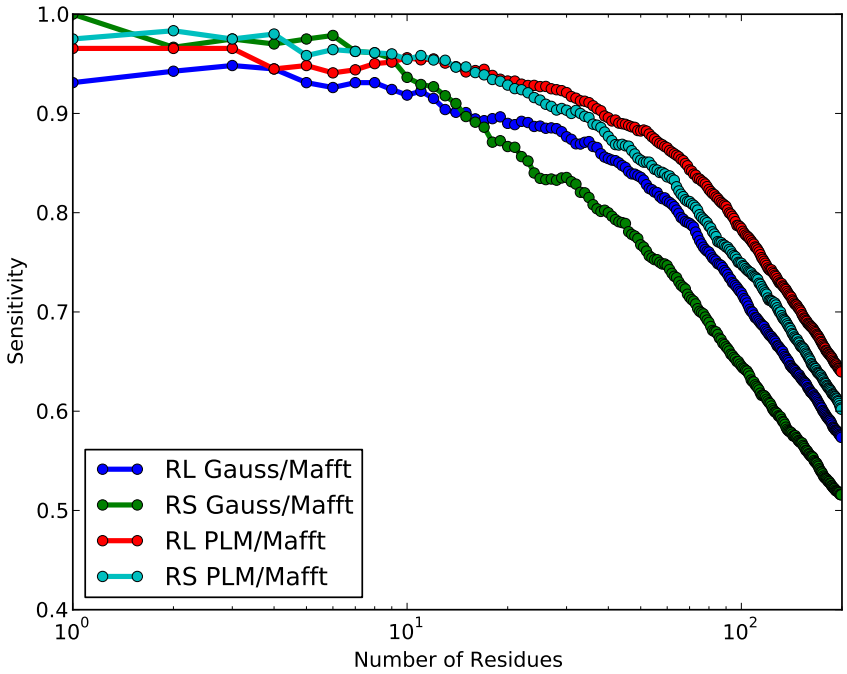


Figure 4.3: Intra-Protein Sensitivity Plots. On the alignments for the single ribosomal proteins the plmDCA algorithm was run and an ordered list of residue pairs obtained. For every number n on the abscissae the fraction of the number of true positives (the sensitivity) in the first n pairs on this list was calculated for every protein. The plot shows the mean of these values for the Gaussian algorithm of [4] and the plmDCA algorithm run on the proteins of the large and small ribosomal subunit. Figure taken from [31].

we chose the `linsi` program of the MAFFT package to create the initial MSAs. The number of sequences for the Trp alignments can be found in Table 4.2.

As an assessment of quality for the alignments, sensitivity plots using the pdb files `2Z4K` and `2Z4L` were made. Figure 4.3 shows results for contact predictions based on the GaussDCA [4] and plmDCA algorithm [26].

4.2.2 Matching Paralogs

For the analysis we imagine that two proteins that interact are drawn for every species together from some probability distribution. This joint distribution takes into account that the proteins are under selective pressure to maintain the interaction and that their

evolution is therefore not independent. The samples on which we infer the probability distribution are therefore made from both and as input data we need a concatenated MSA. The problem of generating a concatenated alignment from two MSAs of two different protein families (say MSA_1 and MSA_2) is then to decide which sequence from the first alignment should be concatenated to which sequence from the other alignment. This means to find for any protein p_i^1 in MSA_1 a matching partner p_j^2 in MSA_2 belonging to the same species. The problem is trivially solved in the case when no paralogs are present and each species has one and only one sequence in each individual MSA. In this case we can simply concatenate these two sequences (we term this case *matching by uniqueness*). The problem is that species often have several paralogs, see Figure 4.4. In this case, given that we would like to observe a co-evolutionary signal between protein interaction partners, one would like to match sequences of proteins that are (possibly) interacting.

As long as prokaryotes are concerned, it has been observed that proteins are more likely to interact if their genes are *co-localized* on the DNA [12, 93]. This suggests to try to match proteins that are close on the genome when creating a concatenated MSA.

As a proxy to the genomic distance we use a *distance* between Uniprot accession numbers (UAN). This UAN consists of a 6 digit alphanumeric sequence for every sequence and can be extracted from the sequence annotation, e.g. the "`D8UHT6`" part of the sequence annotation "`D8UHT6_PANSA`".

We define the distance between UANs as follows: Different positions in the UAN can take on different values, some only numeric (0-9) and some alphanumeric values (0-9,A-Z). We define for every position $i \in 1 \dots 6$ the number B_i as the number of different values position i can take, i.e. $B_i = 10$ for the numeric positions and $B_i = 36$ for the alphanumeric positions.

We further map the possible single position values in the UAN to the natural numbers in ascending order, i.e. we assign to the numeric symbols $0-9$ the natural numbers $0-9$ and to the letters the natural numbers following 9 (so to A we assign 10, to B we assign 11 etc.). This leads for example for the UAN `L9XG27` to the numeric sequence $A = (21, 9, 33, 16, 2, 7)$.

Now we can define a unique number N for any UAN that has been mapped to the sequence of natural numbers A_i as

$$N = A_6 + \sum_{i=1}^{5} A_i \left(\prod_{j=i+1}^{6} B_j \right) \tag{4.1}$$

The distance between two UANs that have been mapped to the numbers N_1 and N_2 can now be defined as

$$D_{12} = |N_1 - N_2| \tag{4.2}$$

This procedure induces a distance D_{ij} for any sequence $p_i \in \mathrm{MSA}_1$ and $p_j \in \mathrm{MSA}_2$, where both p_i, p_j belong to the same species. In this way we define a complete weighted

bipartite graph, and the problem of finding the proper pairing can thus be translated into a minimum weighted bipartite matching problem. This problem can be readily solved using a standard linear programming techniques. Finally we discard from the optimal solution sequence pairs whose distance is above a given threshold of 100 (manually optimized on the small ribosomal subunit). In the cases we analyzed, such a threshold moderately increases the quality of the prediction of interaction partners.

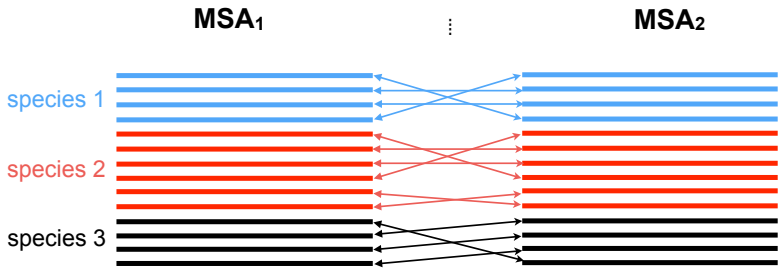


Figure 4.4: **Concatenating two multiple sequence alignments** Figure Caption Sketch of the matching procedure that allows us to concatenate two different MSAs, here MSA_1,MSA_2. π represents the optimal permutation of the sequences on the second MSA computed using a standard linear programming routine. Figure taken from [31].

The average number of paralogs per species varies from system to system: For both ribosomal subunits the proteins have between 1.5 and 3 paralogous sequences per genome. The trp proteins on the other hand have considerably more paralogous sequences and the number of such sequences per genome varies between 4 and 24. This means that especially in the trp operon the matching procedure has the potential to generate much larger alignments than the competing approach of excluding species with paralogous sequences. In fact, using this last approach (which corresponds to setting our threshold parameter to 0) reduces the number of sequences in the alignments on the average by about 10% for the ribosomal proteins and by about 85% for the proteins of the trp operon (see Tables 4.3,4.4,4.5,4.6 and 4.7).

Note that using paralogs may be dangerous since after duplication different paralogs often evolve different functions, and thus lose part of their interactions or gain others. However, our matching strategy based on genomic vicinity excludes proteins coming from isolated genes; it identifies mostly protein pairs coded in gene pairs co-localized inside an operon. It is therefore more likely that the two maintained interaction, when also the ancestral protein pair before duplication was interacting. We will show evidence that, in the interacting protein systems investigated here, this strategy leads to a reinforced co-evolutionary signal. However, an independent and direct test whether protein pairs included in the alignment actually interact would constitute a big step

forward.

Let us recall that the problem of finding a good matching between sequences has already been studied in the past using different strategies [12, 76]. Unfortunately, both methods are computationally too demanding to be used in a case, where hundreds or thousands of protein family pairs have to be matched.

4.2.3 Creating Simulated Data

In order to test the approach in a more controlled setting and to assess the effect of different sampling depths, we generated data from an artificial protein-protein network. As the basis for the simulated data we used a fictitious protein complex consisting of 5 proteins. Each protein has a length of 53 residues. The individual contact map of each one is given by the bovine pancreatic trypsin inhibitor (PDB ID 5pti [98]), which is a small protein performing well for the prediction of internal contacts by DCA. Each P_i has 551 internal contacts. Moreover, each protein interacts with two others in a circular way. The inter-protein contact matrices between P_i and P_{i+1} (as well as between P_1 and P_5) are random binary matrices with a density of 10% of the internal contacts. This models the sparsity of the inter-protein contacts as compared to the intra-protein contacts. A contact map for the artificial complex can be found in Figure 4.5. There are no contacts between other pairs of proteins.

In order to define a probability distribution from which we can draw the samples that make up the artificial data, we used the model described in Equation 2.30. As described in the next paragraph, such a model can be used to described inter-protein co-evolution by using a part of the couplings to describe the co-evolution of residues within one protein and the other part to describe co-evolution of residues in different proteins (see Section *The Generalized Potts Model for Protein-Protein Interaction* below).

In order to define as realistically as possible the coupling parameters of the Potts model used for generating the artificial sequences, we used the Pfam protein family PF00014 of the pancreatic trypsin inhibitor [34]. Note that a member of this family was also used to define the structure. The couplings describing the co-evolution *within* the single proteins were directly extracted from the Pfam MSA using DCA. For the couplings corresponding to the co-evolution *between* the proteins, we used a random subset of the internal parameters and used them to couple sites that are in contact according the contact map as defined above. Non-contacting pairs of sites remain uncoupled between artificial proteins. Using this model, a joint MSA D_{12345} of sequences of length $265 = 5 \times 53$ was generated using standard MC simulations.

The generation of the simulated data contains many steps. To give a more detailed account, we repeat those steps in a point by point list:

1) First, a contact map was defined. This contact map contains the information which residues are in contact. This includes internal residue contacts (where both residues belong to one of the 5 proteins) and inter-protein residue contacts (where one residue

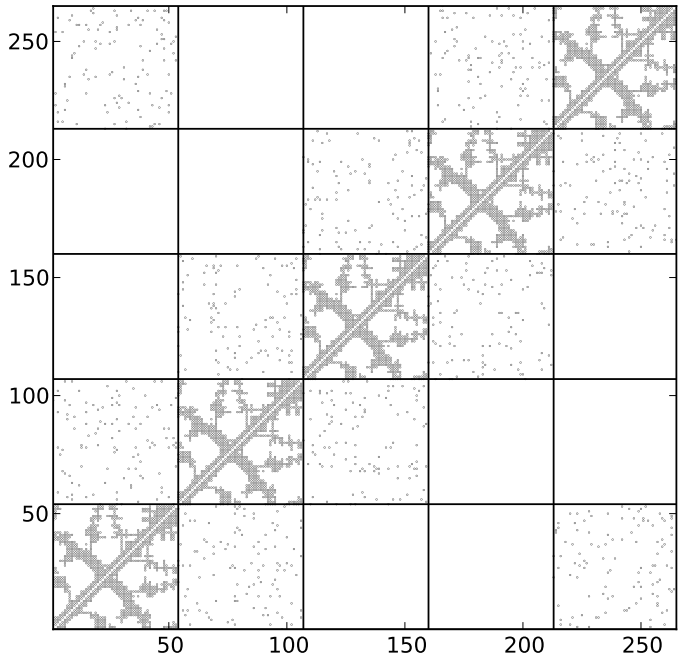


Figure 4.5: Contact map of the artificial protein complex. Figure taken from [31].

belongs to one protein and the other to a different protein). The contact map is therefore a binary, symmetric matrix of size $N_{all} \times N_{all}$ with $N_{all} = N_1 + N_2 + N_3 + N_4 + N_5$ where N_i is the number of residues in the i^{th} protein. As written above, we decided to use the Kunitz domain (PF00014) as a model for the proteins and set all $N_i = 53$. The 53×53 submatrices that define the contacts within each protein were defined by extracting the contacts of the PDB structure 5pti of the Kunitz domain. This implies that the internal structure of every protein is the same.

We defined as contacting proteins the protein pairs $1-2, 2-3, 3-4, 4-5$ and $1-5$. For the 53×53 submatrices that define the contacts between contacting protein pairs we used random binary matrices with 10% of the number of internal contacts. This was done individually for each contacting protein pair such that no two contact matrices between two proteins were the same. For non-contacting protein pairs all entries of the contact matrices were set to 0.

The resulting contact map can be seen in Fig. 4.5.

2) Couplings for every contact in the contact map were defined. As a basis for this, couplings and fields inferred from the PF00014 PFAM alignment (Kunitz Domain) were used. This inference was done using a masking with the PDB structure, such that only couplings corresponding to PDB-contacts were allowed to differ from zero. Given that the same PDB-contacts were used to define the contacts within one protein in the artificial complex, we could use the couplings thus inferred without change for the couplings within the artificial proteins.

Then we defined the couplings for residue contacts between two proteins. For every such a residue contact we chose randomly a coupling of an internal contact as inferred from the Kunitz domain alignment and assigned it to the residue contact.

Notice that the 'coupling' between two sites i and j is actually a 21×21 matrix $J_{ij}(a, b)$ where a and b can be any of the 21 amino acids. Given that the internal structure of these matrices might be important we decided to treat the matrices J_{ij} as single entities and not change their internal structure.

The fields for every residue, a vector of length 21 for every of the $5 \cdot 53$ residues, were randomly chosen from the inferred fields.

From these couplings and fields, sequences were generated by standard MC and inferred by plmDCA. Interestingly, a crude comparison between the histogram of the scores in the artificial model seem to be very close to that obtained for instance for the LRU case as shown in Fig. 4.6.

4.3 DCA for Protein-Protein Interaction Networks

Here we repeat some points presented in Section 2.2.1 and 2.2.3. This should enable the interested reader to follow this Section independently of the others. We nonetheless

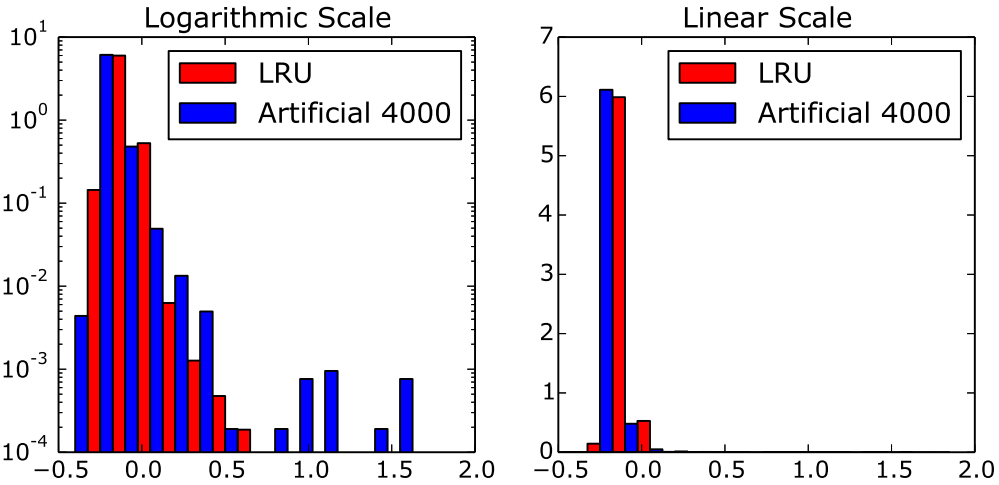


Figure 4.6: Histograms of interaction scores resulting from the analysis of the LRU and the artificial complex (combined strategy). Both intra- and inter-protein scores are included. The plots are normalized such that the area of all bars of a given color sums to one. The data is shown both on a logarithmic (left) and on a linear scale (right). Figure taken from [31].

stress that the model and the inference based on pseudo-likelihoods are described in much more detail in the mentioned Sections.

4.3.1 The Generalized Potts Model for Protein-Protein Interaction

We now repeat the central points of Section 2.2.1 necessary to understand the rest of this section. Within DCA, the probability distribution over amino acid sequences $s = (s_1, ..., s_N)$ of (aligned) length N is modeled by the generalized Potts model, or pairwise Markov Random Field,

$$P(\underline{s} \mid J, h) = \frac{1}{Z(J,h)} \exp\left(\sum_{i=1}^{N}\sum_{j=i+1}^{N} J_{ij}(s_i, s_j) + \sum_{i=1}^{N} h_i(s_i)\right), \qquad \text{(2.30 revisited)}$$

which includes statistical couplings $J_{ij}(s_i, s_j)$ between residue pairs and position-specific biases $h_i(s_i)$ of amino-acid usage [93]. The function of the parameters $Z(J, h)$ is the normalization of $P(\underline{s} \mid J, h)$, which is a probability distribution over all amino-acid sequences of length N. The variable s_i represents the amino acid found at position i in the sequence and can take as values any of the $q = 21$ different possible letters in an MSA (gaps are treated as a 21st amino acid). The model parameters are inferred using MSAs of homologous proteins.

In the case of two concatenated protein sequences $(\underline{s}, \underline{s}') = (s_1, ..., s_N, s'_1, ..., s'_{N'})$, we write the joint probability distribution in the form

$$P(\underline{s}, \underline{s}') = \frac{1}{Z} e^{-H(\underline{s}) - H'(\underline{s}') - H^{int}(\underline{s}, \underline{s}')}, \tag{4.3}$$

where N is the length of the first protein and N' is the length of the second protein. This is of course nothing else than Equation 2.30, written for a protein sequence of length $N + N'$, where the Hamiltonian has been split into three parts: One in which the J_{ij} appear for which $i \leq N$ and $j \leq N$ (the term H), one for which $i > N$ and $j > N$ (the term H'), and one for which $i \leq N$ and $j > N$ (the term H^{int}). The function

$$H^{int}(\underline{s}, \underline{s}') = -\sum_{i \in s, j \in s'} J_{ij}(s_i, s'_j) \tag{4.4}$$

describes the co-evolutionary coupling between the two protein families. In the last expression, s_i is the ith amino acid in sequence $\underline{s}$, and s'_j the jth amino acid in sequence $\underline{s}'$. The sum runs over all inter-protein pairs of residue positions. The $q \times q$ matrices J_{ij} in this term quantify how strongly sites between the two proteins co-evolve in order to maintain their physicochemical compatibility. The matrix contains a real number for each possible amino acid combination at sites i and j and contributes to the probability in Equation 4.3 depending on whether an amino acid combination is favorable or not. The strongest inter-protein couplings are enriched for inter-protein contacts, see [70, 93]and Section 2. The same kind of model can be used to predict the interaction between more than two proteins, with a corresponding number of interaction terms. However, the number of parameters in the model is proportional to $(N_1 + N_2 + .. + N_K)^2$ for K proteins while the number of samples in the concatenated MSA $D_{p_1,....p_K}$ becomes smaller because one has to find matching sequences for K proteins *simultaneously*. This leads us to consider the case $K > 2$ only for artificial proteins where the total length and sample size are controllable.

4.3.2 Inference and Scoring

Following [27], the parameters of the model were inferred by maximizing *pseudo-likelihood functions*. This is an alternative to directly maximizing the likelihood and considerably faster.

The inference proceeds by considering the conditional probability distribution

$$P_i(s_i | s_{/i}) = \frac{\exp\left(\sum_{j \neq i} J_{ij}(s_i, s_j) + h_i(s_i)\right)}{\sum_{a=1}^{21} \exp\left(\sum_{j \neq i} J_{ij}(s_i, a) + h_i(a)\right)} \tag{4.5}$$

Given a data set D we can thus maximize the conditional likelihood corresponding to site i by maximizing

$$L_i(J_i, h_i) = \frac{1}{M} \sum_{m=1}^{M} w_m \log P_i(s_i^m | s_{/i}^m) \quad , \tag{4.6}$$

as a function of the J and h that are connected to site i. The w_m are sequence weights that are used to correct for biased sampling and phylogenetic bias (see Section 2.2.2. As customary in many maximum-likelihood inference techniques, we add to the maximization an $L2$ regularization term, so that eventually the extremization procedure turns out to be:

$$\{J_i^*, h_i^*\} = \underset{J_i, h_i}{\operatorname{argmax}} \{L_i - \lambda_J \sum_{j \neq i} \|J_{ij}\|_2 - \lambda_h \|h_i\|_2\} \,, \tag{4.7}$$

with $\|J_{ij}\|_2 = \sum_{a,b=1}^{21} J_{ij}^2(a,b)$, and $\|h_i\|_2 = \sum_{a=1}^{21} h^2(a)$. We refer to the original paper [27] for the details of the implementation, and mention again that a more detailed explanation can be found in Section 2.2.3. We also add that beside the original MATLAB [64] implementation available here, we developed an efficient implementation of the pseudo-likelihood implementation in a new open-source language called Julia [7]. The package can be downloaded here.

Given that the model is mathematically equivalent to the one used in [27] we can use the output of the algorithm (plmDCA) with default parameters as presented there directly for our purposes. This output consists of scores F_{ij}^{APC} that quantify the amount of co-evolution between sites i and j in the alignments and are defined in Equation 2.72. In order to quantify co-evolution between *proteins*, we took the F_{ij}^{APC} corresponding to inter-protein site pairs (i.e. i in $\underline{s}$ and j in $\underline{s}'$) and calculated the mean of the 4 largest. These quantities, a real number for every protein pair, are used to rank protein-protein interaction partners. The number 4 was chosen because it performed well in the small ribosomal subunit, which we used as a test case when designing the algorithm. Subsequent tests on larger systems showed that any number between 1 and 6 performs almost equally well, as can be seen in Figure 4.13.

The list of protein pairs ordered by this score was used for prediction. The first few predictions are shown in Table 4.8. For completeness, we show the same table but with the score calculated by the Gaussian approximation of [4] in Table 4.9. Finally in Table 4.11 we display for the LSU the number of intra/inter-protein contacts, while in Table 4.12 we do the same for the LRU.

Table 4.10 shows the interaction scores for the protein pairs of the Trp Operon.

4.4 Inference Results

4.4.1 Simulated Network

As a first test of our approach, we use *simulated data* generated by Monte Carlo (MC) sampling of a Potts model of the form of Equation 4.3, see Section 4.2.

The main simplifying assumptions in this context are: (i) We assume intra- and inter-protein co-evolution strengths to be the same. (ii) We assume the distribution of inter-protein residues contacts within the possible contacts to be random. (iii) We assume the sequences to be identically and independently distributed according to our model. This model includes the assumption that non- contacting sites have zero couplings. The number of artificial sequences needed for a good performance of our method should therefore be taken at most as a lower bound for the number of biological sequences needed for a comparable performance.

In panel A of Figure 4.7 we show the architecture of our artificial protein complex. It is composed of five fictitious, structurally identical proteins $P_1, ..., P_5$, each one consisting of 53 residues. In order to simulate co-evolution between the proteins, we generate a *joint* MSA D_{12345} for all 5 proteins with a model that contains couplings between inter-protein site pairs. These couplings are modeled in a way to resemble couplings inferred from real proteins (see Section 4.2).

To assess our capability to infer the PPI network of panel A from such data, we adopted two different strategies which we called *combined* and *paired* in panel B of Figure 4.7. The *combined* strategy uses plmDCA on the full-length alignments of length 265 and models the interaction between all proteins pairs *simultaneously*. Given that in this artificial setting we use the same model to generate the data as to analyze it, the approach is guaranteed to infer the model correctly for a large number of analyzed sequences and therefore to assign a higher interaction score to any interacting protein pair than to any non-interacting pair.

To assess the coupling strength between two proteins, we average the four strongest residue coupling strengths between them. This leads to a score oriented toward the strongest signal while also reducing noise by averaging. In panel B of Figure 4.7 we show the results for MSA sizes $M = 2000, 4000, 24,000$ while intermediate values are reported in Figure 4.8. The two lower figures - $M = 2000, 4000$ - represent the lower and upper bound of what we can currently obtain from databases for the proteins analyzed by us. The largest value $M = 24,000$ is what we expect to be available in a few years from now, seen the explosive growth of sequence databases. The thickness of each link in Fig. 4.7 is proportional to the inferred inter-protein interaction score. The five strongest links are colored in green when they correspond to actual PPI according to panel A, and in red when they correspond to non-interacting pairs. For increasing sample size the predictions become more consistent and for $M = 24,000$ any interacting protein-pair has a higher interaction score than any non-interacting pair.

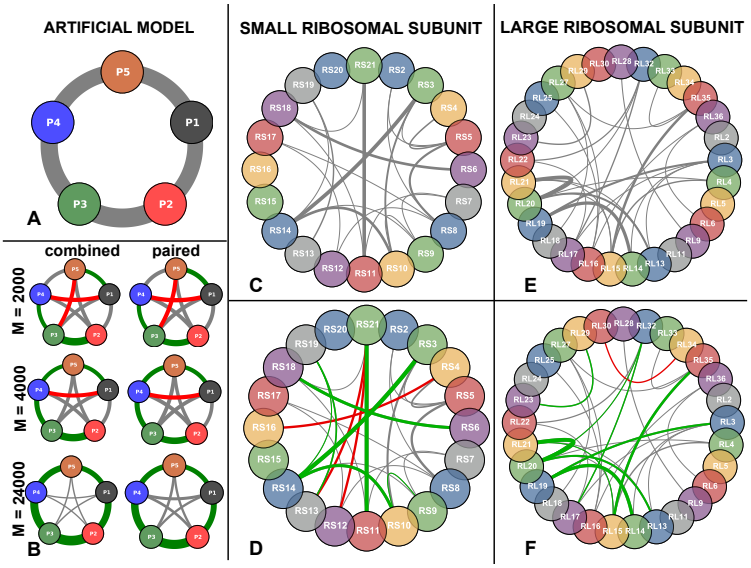


Figure 4.7: **Residue-residue structure of both artificial and ribosomal complex A** Architecture of the *artificial* protein complex. Arcs width are proportional to the number of inter-protein residue contacts. **B** Inferred PPI network for both *paired* and *combined* strategy for different number M of sequences generated from the artificial model. Green arcs are true positives, red false positives, gray low-ranking predictions. Arc widths are proportional to the inter-protein interaction score. **C** SRU architecture (same color code as A). **D** Inferred PPI network (same color code as B). **E** Same as C for LRU. **F** Same as D for LRU. Arc width in panels C-F is provided by the number of inter-protein contacts, as a measure of interface size. It becomes obvious that mainly large interfaces are recognized by our approach. Figure taken from [31].

Due to the running time of plmDCA only alignments for sequences of total length $L \lesssim 1000$ can be analyzed. This is exceeded already by the sum of the lengths of the proteins of the small ribosomal subunit. Additionally, creating a combined multiple sequence alignment for more than two proteins would lead to very low sequence numbers due to the necessary matching (see Section 4.2). Therefore, using the combined strategy is not generally applicable. In the *paired* strategy we therefore analyze each pair of proteins separately. This means that plmDCA is applied to all $\binom{N}{2}$ protein-pair alignments D_{ab}, $1 \leq a < b \leq N$. In panel B of Fig. 4.7 we find that the paired strategy is also able to detect the correct PPI network for large enough M. We observe, however, that the performance of the paired strategy is slightly worse. Couplings between non-interacting proteins are estimated significantly larger than using the combined strategy for large M. Even in the limit $M \to \infty$ we do not expect these links to

disappear: Correlations between, e.g., P_1 and P_3 are generated via the paths $1-2-3$ and $1-5-4-3$, but in the paired strategy these correlations have to be modeled by direct couplings between P_1 and P_3 since the real direct coupling paths are not contained in the data.

After having answered the *'who-with-whom'* question for the artificial protein network, we address the *'how'* question of finding inter-protein contact pairs. Figure 4.9 panel A displays individual residue contact pairs within and between proteins in the artificial complex. Panel B shows the 10 strongest intra-protein couplings for each protein and the 10 strongest inter-protein couplings inferred by plmDCA ($M = 4000$, combined strategy). Green links correspond to contact pairs and red links to non-contact pairs. We see that the intra-protein prediction is perfect, whereas a few errors appear for inter-protein predictions in agreement with the results of Figure 4.7.

Finally, in Table 4.13 we compare the ranks of the strongest inter-protein residue interaction scores in the generating model and the inferred model. The first column represents the rank of the inter-protein residue interaction in the generating model, the second column the rank of the same residue interaction in the inferred model. The model was inferred with the combined strategy and with 4000 sequences. The numbering is treating the complex as one large protein.

4.4.2 The PPI network of bacterial ribosomes

As a more realistic test we apply the method to the bacterial large and small ribosomal subunits (LRU, SRU). To define contacts and protein interaction partners we used high-resolution crystal structures with PDB-IDs 2z4k (SRU) and 2z4l (LRU) [9]. The contact network is summarized by the contact maps in Figures 4.1 and 4.2. The ribosomal RNA is ignored in our analysis.

Panels C, E of Figure 4.7 display the architectures of both SRU and LRU. The SRU (LRU) complex consists of 20 (29) proteins of lengths 51-218 (38-271); 21 (29) out of $\binom{20}{2} = 190$ ($\binom{29}{2} = 406$) pairs are in contact. The interfaces contain between 3-209 (1-229) residue pairs. The width of the inter-protein links in the PPI network Figure 4.7 in panels C, E are proportional to these numbers. The number of contacts within the individual proteins ranges from 297 to 2337 (303-2687). Globally, there are 22644 (30555) intra-protein and 1401 (1,439) inter-protein contacts, so the contacts relevant for our study comprise only 5.8% (4.5%) of all contacts.

Fig. 4.7 panel D shows the inferred SRU PPI architecture. As expected, the biological case is harder than the artificial case where the data are independently and identically distributed according to the generating model. Even though the histograms of the inferred interaction scores for both cases are very similar (see Figure 4.6), biological data are expected to show non-functional correlations due to the effect of phylogeny or sequencing efforts which are biased to model species and known pathogens.

Nonetheless, among the top ten predicted interacting protein pairs the method makes

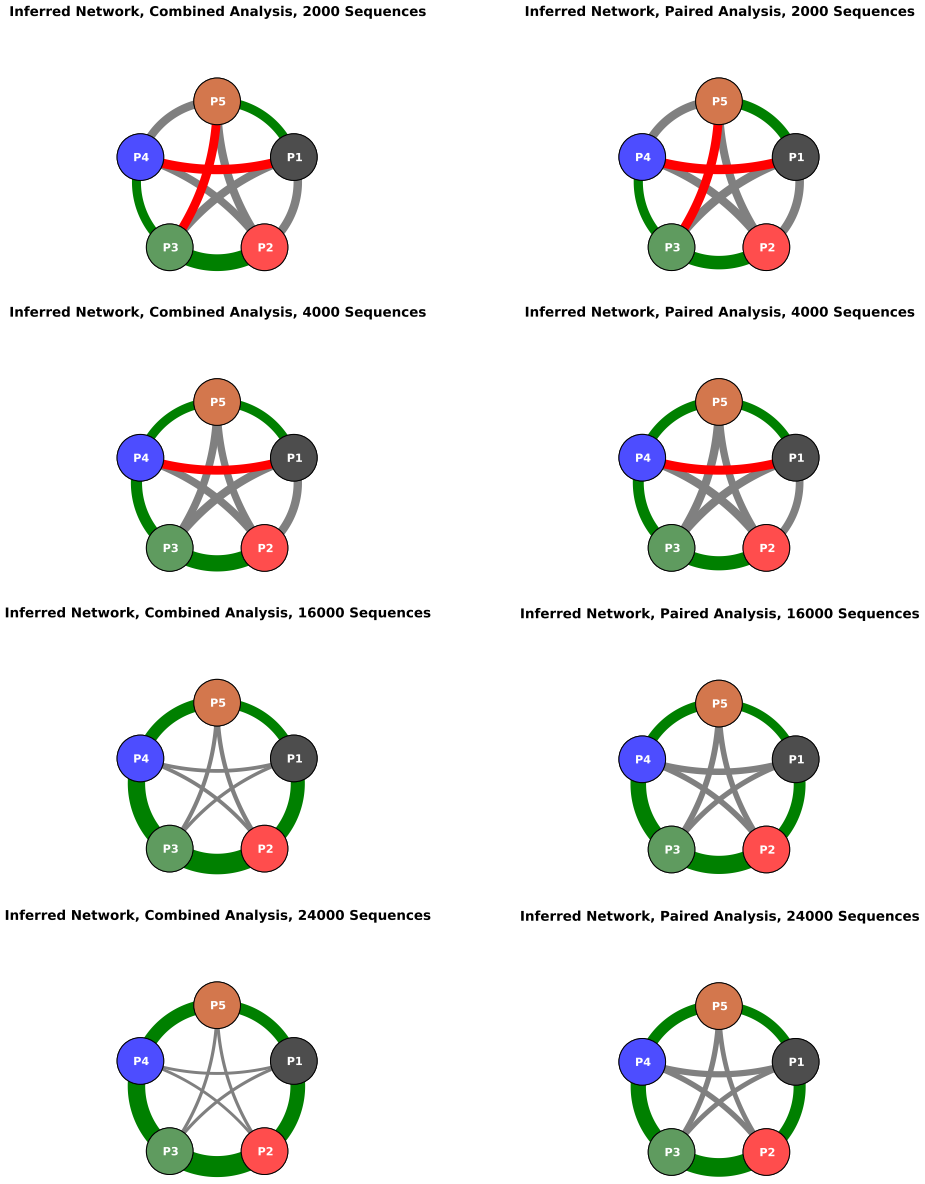


Figure 4.8: (Caption on following page.)

Figure 4.8: Inferred protein network for different sample sizes; the line-thickness is proportional to the inferred interaction scores between the proteins (mean of the 4 highest residue interaction scores). The thickness has been normalized in the sense that the scores have been divided by the mean of the scores of the network. The color code is applied for the first 5 predictions and shows a green line if the prediction is a true positive and a red line if the prediction is a false positive. Predictions after the first 5 are grey.
Combined Analysis: The complete sequences in their whole length were used for the inference and calculation of the scores
Paired Analysis: Every protein family was independently cut out of the generated sequences and thus a MSA for only this protein created. These single MSAs were then paired for all protein pairs and used for inference and calculation of the scores. Figure taken from [31].

only three errors (true-positive rate 70% as compared to $21/190 \simeq 11\%$ true PPI between all protein pairs, with an overall area under ROC curve (AUC) of 0.69, see Figure 4.15). The method spots correctly the pairs with larger interaction surfaces whereas the small ones are lost. Two of the false-positive (FP) predictions include protein RS21, which has the smallest paired alignments with other proteins (M between 1468 and 1931). Also the third FP, corresponding to the pair RS4-RS18, is probably due to a small MSA with $M = 2064$. At the same time, the interaction of RS21 with RS11, which is one of the largest interfaces (199 contacts), is still detected despite the low $M = 1729$. The same procedure for the LRU (406 protein pairs) performs even better: 9 out of the 10 first PPI predictions are correct (see Figure 4.7 panel F), and the AUC is 0.81.

The results on the residue scale for both SRU and LRU are depicted in panels D and F of Figure 4.9. Shown are the first 20 intra-protein residue contact predictions for each protein (excluding contacts with linear sequence separations below 5 to concentrate on non-trivial predictions) and the first 20 inter-protein residue contact predictions. In the SRU case of panel D for example, the results are qualitatively similar to the artificial case, albeit with a slightly reduced true-positive rate of 60% among the first 20 inter-protein residue contact predictions (compared to the ratio of 1401 actual inter-protein residue contacts and 2,403,992 possible inter-protein residue contacts, i.e., 0.058%). Again 3 out of the 8 false positives are related to RS21, which due to the smaller MSA size is also the only one having a considerable false-positive rate in the intra-protein residue contact prediction. About 95% of the displayed 400 highest intra-protein residue contacts are actually contacts (see Figure 4.3). Analogous considerations with a somewhat larger accuracy (85%) hold for LRU as displayed in Figure 4.9 panel F.

4.4.3 The PPI network of the tryptophan biosynthetic pathway

As a distinct test case for our methodology we analyzed the 7 enzymes (TrpA, B,C,D,E,F,G) that comprise the well characterized tryptophan biosynthesis pathway. In contrast to the ribosomal proteins, these enzymes are only conditionally essential in the absence of environmental tryptophan and their genes are only expressed under deplete tryptophan conditions. In this particular system, only two protein-protein interactions are known and resolved structurally: TrpA-TrpB (PDB-ID 1k7f [95]) and TrpG-TrpE (PDB-ID 1qdl [58]). Whereas the TrpG-TrpE pair catalyzes a single step in the pathway and their interaction is thus essential for correct functioning, the TrpA-TrpB pair catalyzes the last two steps in tryptophan biosynthesis. Both enzymes function in isolation but their interactions are known to increase substrate affinity and reaction velocity by up to two orders of magnitude. All other proteins catalyze individual reactions, but one might speculate that the efficiency of the pathway could benefit from co-localization of enzymes involved in subsequent reactions. Interestingly, the Pfam database [34] reports that in many species pairs of genes in the operon appear to be fused, suggesting that some of the fused pairs are actually PPI candidates. An example is the TrpCF protein, which is fused in *Escherichia coli* and related species (but not in the majority of species).

After applying our method to all 21 protein pairs we find elevated interaction scores only for TrpA-TrpB and TrpE-TrpG, which are the only known interacting pairs (see Figure 4.10 and Table 4.10 for the interaction scores of all pairs). Those two pairs have interaction scores of 0.375 and 0.295, while the other pairs are distributed between 0.071 and 0.167. Even though we do not define a significance threshold for prediction (see Section 4.4.4), these two pairs would be discernible as interesting candidates even if we did not have the 3D structures.

We speculate therefore that the fusions in many species do not imply strong inter-protein co-evolution. To further investigate this aspect, we took a closer look at the protein pair TrpC-TrpF. For this protein pair, a high resolution structure of a fused version exists (PDB-ID 1pii [97]). We ran our algorithm on the complete multiple sequence alignment, the multiple sequence alignment with fused sequence pairs removed and only on the fused sequences. In none of these cases did we observe a statistically significant interaction score or a statistically significant prediction of inter-protein contacts present in the structure of the fused protein.

Our results are corroborated by the finding that all scores measuring the co-evolution between a ribosomal protein and an enzyme from the tryptophan synthesis pathway are small (see the following subsection). No indication for an interaction between the two systems is found, as to be expected from the disjoint functions of the two systems.

4.4.4 Inference in a Network Combining All Tested Proteins

It is interesting to assemble a larger-scale system out of the three systems (SRU, LRU, Trp). To this end, we created all possible pairings between the proteins used in the present study (SRU vs. RU, SRU vs. Trp, LRU vs Trp, SRU vs SRU, LRU vs. LRU, and Trp vs. Trp). This leads to a total of 1540 pairs, out of which only 49 pairs are known to interact (which we defined as true positives). We present the findings in Figure 4.15 and in Figures 4.12,4.13 and 4.14. Figure 4.12 shows the true-negative rate, which is the fraction of true negatives in the indicated number of predictions with the *lowest* interaction scores. As it can be seen our scoring produces a false negative just after 420 true negatives.

Figures 4.15 and 4.13 show true positive rates for the complete system and the individual systems. We also show true positive rates for alternative ways to calculate the interaction score between protein pairs, i.e. a different number of inter-protein residue-residue interaction scores to average. We notice that in the complete system the performance is similar to the performance in individual systems. All of the 10 highest-scoring protein pairs are known to interact, and 75% of the first 20 protein-pairs. After these first 20 pairs, the true positive rate drops to around 45% in the first 40 predictions. This is analogous to the case of protein contact prediction, where methods based on the same model are able to extract a number of high confidence contacts but see a large drop in performance afterwards [68]. The area under the AUC for the whole system is 0.83 (see Fig. 4.15). This is stable when averaging different numbers of residue-contact scores to arrive at a protein-protein interaction score, but the performance seems to worsen when using more than 6. This is probably because only a few inter-protein residue contacts have a large score and averaging over too many only adds noise. It can also be seen that averaging over 4 performs very well in the small ribosomal subunit, which is why we have chosen this value for the large part of the analysis. On the larger-scale system, though, any number between 1 and 6 performs almost identically.

A further question is whether it is possible to define a threshold allowing to reliably discriminate between interacting and non interacting pairs in terms of the interaction score. Figure 4.14 shows two normalized histograms of the interaction scores. The rightmost tail of the interacting pairs distribution is well separated from the rightmost tail of non-interacting one, but the highest scores of non-interacting pairs are strongly overlapping with the lowest scores of the interacting ones. The situation is therefore analogous to what is observed in the case of the inference of contacts within single protein families [4,27,68,93], where the same technique is known to produce relatively few high confidence contacts in the topmost scoring residue pairs. To conclude, while high scores seem to reliably predict interacting pairs, and low scores non-interacting pairs, there is a large gray zone prohibiting a clear discrimination between interacting and non-interacting pairs.

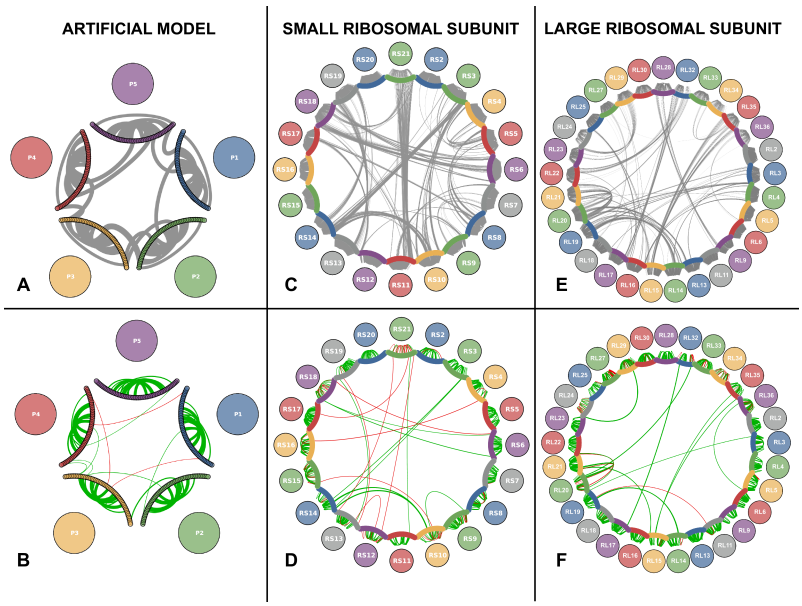


Figure 4.9: **Architecture and inferred protein-protein interaction network of the artificial protein complex A** Residue-residue interaction structure of the generating model for the artificial data. Colored arcs represent the protein chain. Non-zero couplings in the coupling matrix of the generating model are represented as curves between the nodes. The width of the curves is proportional to the interaction score. Only the 10 strongest intra/inter-protein scores are shown. **B** Same as **A**, but based on the inferred couplings. Green arcs are true positives, red false positives. Note that not all green arcs have a corresponding arc in **A** due to our choice to display only the 10 strongest couplings, which not always correspond to the strongest score. **C** Same as **A** for SRU. All links represent a contact in the PDB structure and have equal width. **D** Same as **B** for SRU. **E** Same as **C** for LRU. All links represent a contact in the PDB structure and have equal width. **F** Same as D for LRU. Figure taken from [31].

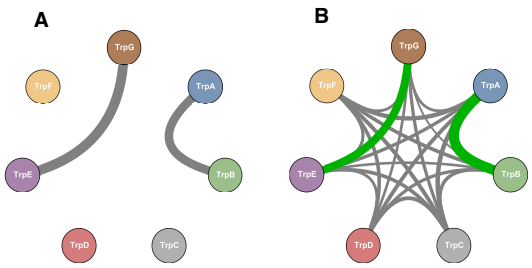


Figure 4.10: **Tryptophan biosynthesis pathway A** Architecture of the known protein-protein interaction among the 7 enzymes which are coded in the Trp operon. The widths of the arcs are proportional to the number of inter-protein residues (which in this case is almost equal for the two interacting pairs). **B** Inferred PPI network, here the width of the arcs is proportional to the interaction score. Green arc correspond to the protein pairs for which a known structure exist. Figure taken from [31].

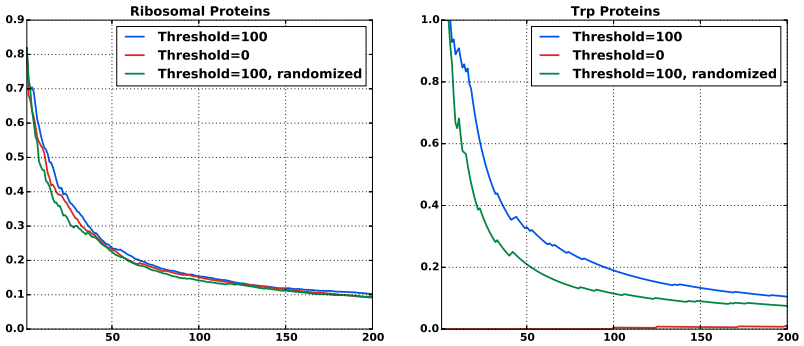


Figure 4.11: **Efficacy of the different matching procedures** True-positive rates for inter-protein residue contact prediction for different matching procedures. Shown are means for all protein pairs that have at least 100 residue pairs in contact. The ribosomal and the trp proteins were tested independently. The red curves correspond to a matching including only protein sequences without paralogs inside the same species ("matching by uniqueness in genome"). The low performance of this approach on Trp proteins is due to a very low number of species without homologs, which leads to very small matched alignments. The blue curves show the results for our matching procedure as described in the text. The green curves correspond to alignments that have been obtained by first applying our matching procedure and then randomizing the matching within individual species. The definition of "contact" was the same as used above (a distance of less than 8.0Å between two heavy atom in the residues). Figure taken from [31].

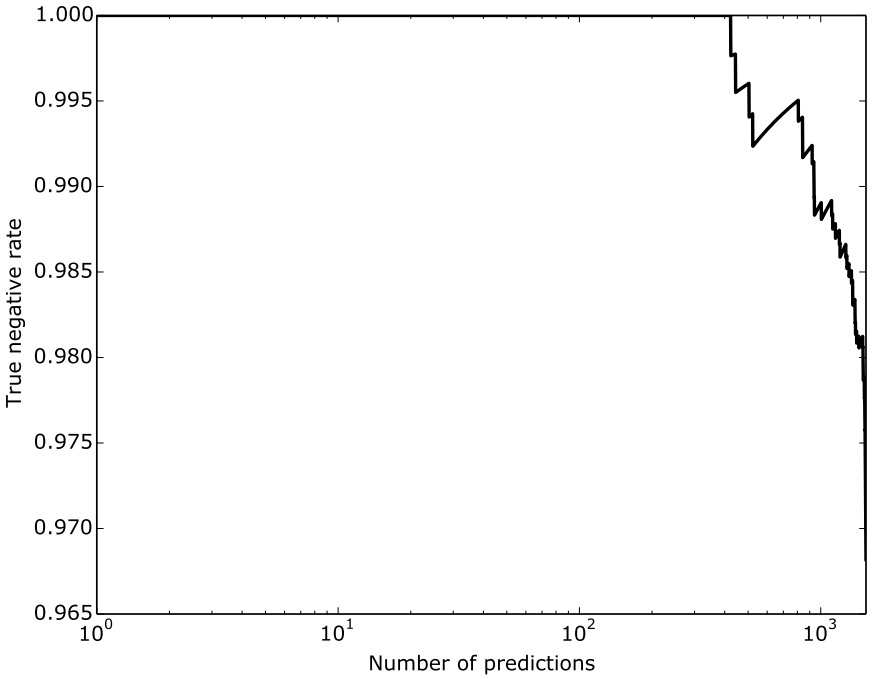


Figure 4.12: True negative rate; all possible protein pairs between RS,RL and Trp proteins are considered and the protein-protein interaction score is defined as the average of the 4 largest interaction scores on the residue level (as in the main paper). The true negative rate is the fraction of true negatives in the N pairs with the lowest interaction score, where N is the value indicated by the x-axis. Figure taken from [31].

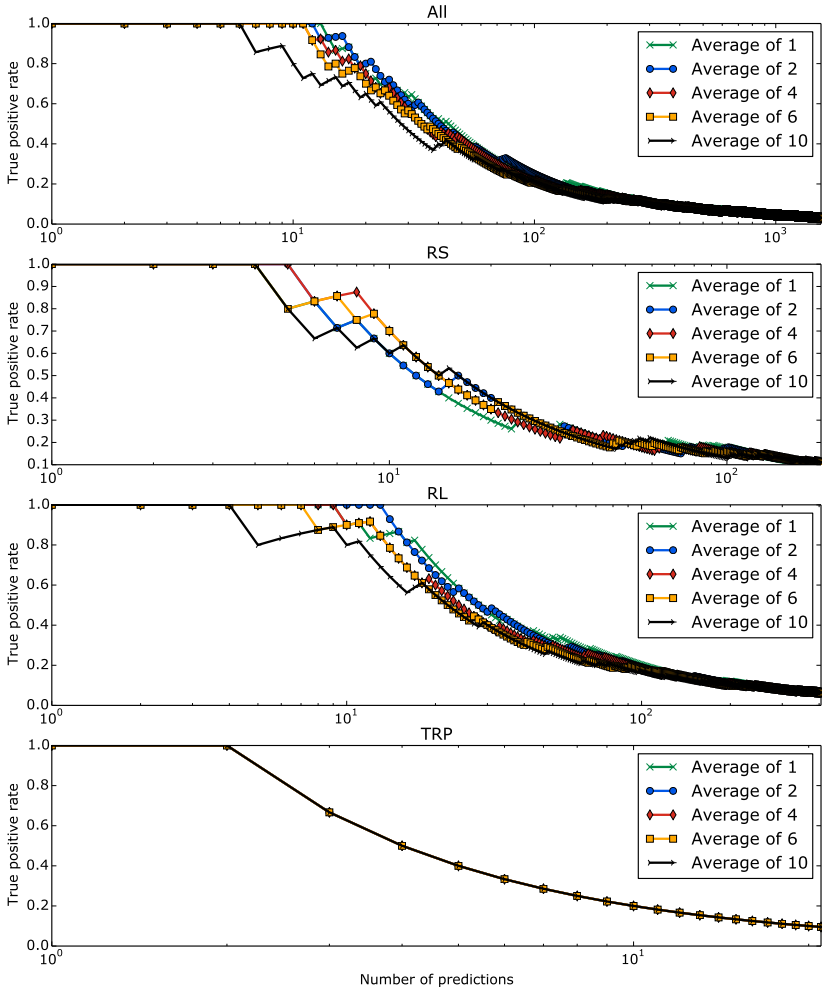


Figure 4.13: True positive rates at a given number of predictions; All: All possible protein pairs between RS, RL and Trp proteins are considered; RS: Protein pairs within the small ribosomal subunit; RL: Protein pairs within the large ribosomal subunit; Trp: Protein pairs of the Trp operon. Different lines indicate a different number of averaged inter-protein scores on the residue level to get a protein-protein interaction score. Figure taken from [31].

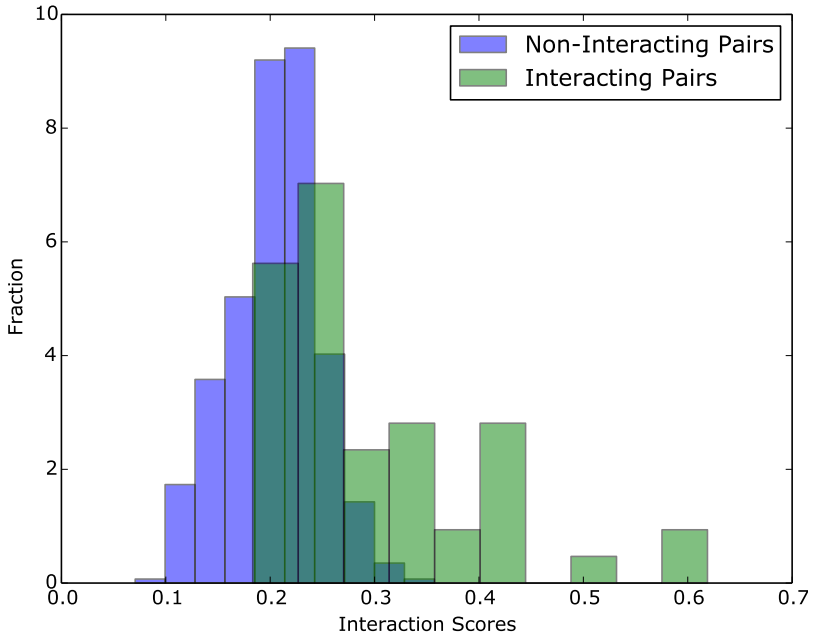


Figure 4.14: Histograms of interaction scores in the network comprising all possible protein pairs between RS, RL and Trp proteins. The protein-protein interaction scores were calculated averaging the 4 largest inter-protein residue interaction scores (as in the main paper). The histogram shows true positives and true negatives separately. Both histograms are normalized. Figure taken from [31].

4.5 Conclusion

To conclude this Section, we have shown that DCA performs excellently in the systems tested when used to predict protein-protein interaction partners. In the small and large ribosomal subunit our tests resulted in a true positive rate of 70% and 90% in the first 10 predictions (AUC of 0.69 and 0.81) while in the trp operon the two largest interaction scores corresponded to the only two interactions experimentally known (AUC 1). The performance is summarized in Figure 4.15. The figure shows both the high quality of the first predictions, but also a drop in performance after a fraction of all interacting pairs (about 40% in our test case). This is analogous to the case of protein contact prediction by DCA and related methods, where the performance drops after a limited number of high-confidence predictions [68]. In the same context and with the same caveat, an excellent performance in predicting inter-protein contacts on the residue level has been shown. The artificial data have shown that the performance of our approach depends crucially on the size of the alignments. Only for very large MSA ($M = 24,000$ sequences in our data) a perfect inference of the artificial PPI network was achieved. MSA for real proteins pairs are typically much smaller. Even for pairs of ribosomal proteins, which exist in all bacterial genomes, only about 1500-3200 sequence pairs could be recovered. This places these data towards the lower detection threshold of PPI. We therefore expect the performance of the presented approach to improve in the near future thanks to the ongoing sequencing efforts (the number of sequence entries in Uniprot [16] has been growing from about 10 millions in 2010 to 90 millions in early 2015) and improved inference schemes. The strong performance of the same algorithm on different and dissimilar systems naturally prompts us to expect that the approach can be used to detect interactions experimentally unknown so far. In fact, if we trust our results on the trp operon we can already draw some speculative biological inferences. While there are many high-resolution structures of the ribosome available, one might have expected that in the trp operon there could be more transient previously unreported interactions in the tryptophan biosynthesis pathway beyond the two interactions that have been structurally characterized. As mentioned, various enzyme fusions can be observed in the databases, suggesting that there is an evolutionary benefit to co-localizing the enzymes of the pathway in the cell. An obvious benefit of such co-localization would be that the pathway intermediates do not have to diffuse throughout the cell from one enzyme component to the next. In the tryptophan biosynthesis pathway in particular, there are numerous phosphorylated intermediates that need to be protected from unspecific cellular phosphatase activity. Organizing the enzymes in the pathway in a multi-protein complex would seem like an efficient way to protect the intermediates from decay. However, our data indicate that the only statistically relevant co-evolutionary signals that can be observed are restricted to the known strong interactions between TrpA with TrpB and TrpE with TrpG. This could be interpreted in a number of ways: *(i)* The most obvious explanation is that there are no additional protein-protein interactions beyond those that are known and that no multi-enzyme complex exists for the tryptophan biosynthesis pathway. Alternatively *(ii)* it seems plausible that there are numerous structural solutions to

form a tryptophan biosynthesis complex and that there is no dominant structure from which a co-evolutionary pattern can be observed in the sequence databases. Lastly *(iii)* it is not out of question that the enzymes of the pathway do not directly form a complex but that they are jointly interacting with an unidentified scaffold component. Of course we cannot exclude that our method is not able to capture other potentially present interactions.

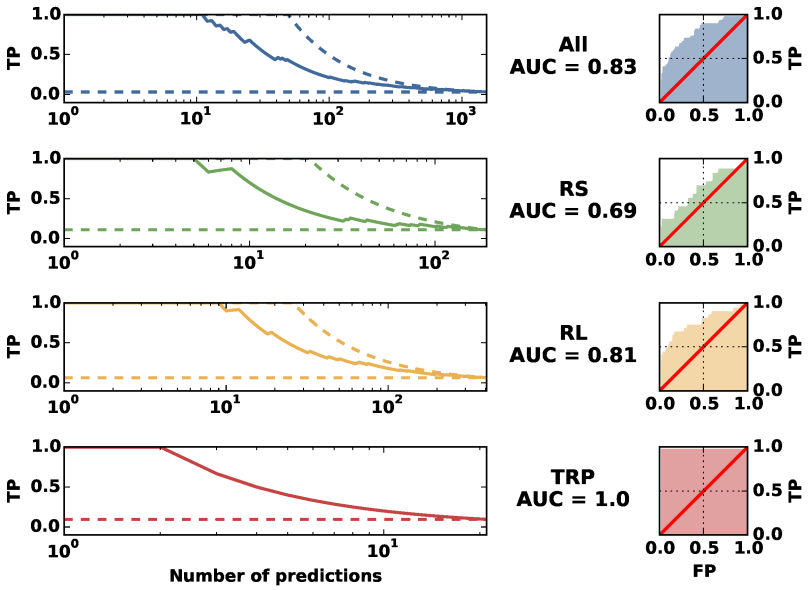


Figure 4.15: **Performance Summary** The plots illustrate the performance in predicting protein interaction partners. The left panels show the fraction of true positives among the first n PPI predictions, with n being the number indicated on the horizontal axis (solid lines). The dashed lines show the best possible (upper dashed line) and the mean of a random prediction (lower dashed line). The right panels show ROC-curves, which indicate the dependence of the true-positive predictions (TP/P) from the false positive predictions (FP/N). The area under the curve (AUC) is a global global measure for the prediction quality; it is 1/2 for a random, and 1 for a perfect prediction. A protein pair is identified as an interacting (true positive) pair, if at least one PDB structure with at least one inter-protein contacts exists. Figure taken from [31].

From a methodological point of view, one possible algorithmic improvement is creating better MSAs for protein pairs. The vast majority of protein families show genomic amplification within species. This raises the issue of which sequence in one MSA should be matched with which sequence in the other MSA when concatenating the two MSAs, as shown in Fig. 4.4. In the absence of prior knowledge and as long as only prokaryotes are concerned, we showed that it is possible to use the simple

criterion of *matching by genomic proximity*. This criterion is based on the observation that two sequences are more likely to interact if they are genomically co-localized. Our results have shown that in the case of the ribosomal network better inference results can be obtained by using this matching criterion than by using a random matching or using a conservative matching taking only species with a single sequence in both MSAs into account, see Figure 4.11. However, we found it beneficial for the predictive performance to introduce a threshold distance above which we simply discarded candidate sequences. This is not based on biological principles.

We believe that our *naive* matching strategy can be improved substantially. Even if closeness of sequence pairs on the genome is a good proxy for interaction in some cases, for example if they belong to the same operon, excluding all distal pairs is a very crude criterion. This criterion is known to be erroneous in many cases, for example in the bacterial two component signal transduction system [12, 14, 76]. It would therefore be interesting to include the matching into the inference procedure itself, *e.g.* to find a matching that maximizes the inter-protein sequence covariation, see [12] for a related idea. However, for highly amplified protein families this leads to a computationally hard optimization task. Simple implementations get stuck in local minima and do not lead to improvements over the simple and straight-forward scheme proposed here.

4.6 Tables

All tables in this section are taken from [31].

	L	M	P	S
RS2	219	6053	1.743	5.978
RS3	216	6235	1.716	7.761
RS4	171	8522	2.175	11.305
RS5	164	5075	1.678	5.845
RS6	105	4132	1.563	6.630
RS7	147	5733	1.595	4.962
RS8	127	5761	1.700	5.992
RS9	127	4983	1.663	5.917
RS10	100	4560	1.511	4.232
RS11	120	5136	1.520	4.019
RS12	124	5607	1.581	4.036
RS13	116	5729	1.856	5.763
RS14	96	5555	1.689	4.780
RS15	89	5361	1.646	6.036
RS16	83	4463	1.507	5.851
RS17	82	4774	1.616	5.481
RS18	73	4512	1.483	4.879
RS19	89	5364	1.537	4.700
RS20	88	3848	1.676	7.460
RS21	65	3209	1.456	4.188

	L	M	P	S
RL3	205	6077	2.025	6.522
RL4	198	5671	1.906	6.810
RL5	177	5032	1.636	6.245
RL6	178	5308	1.765	6.894
RL9	149	4199	1.698	7.621
RL11	141	5027	1.683	5.517
RL13	147	5091	1.717	6.458
RL14	120	5145	1.528	4.358
RL15	140	5926	1.964	6.754
RL16	133	5673	1.604	4.904
RL17	121	4345	1.612	7.637
RL18	111	4961	1.674	6.570
RL19	116	4079	1.511	6.454
RL20	119	4476	1.554	5.864
RL21	102	4123	1.551	6.486
RL22	108	6378	1.918	5.790
RL23	87	5632	1.711	6.292
RL24	99	9062	3.073	12.820
RL25	186	3272	1.680	6.109
RL27	89	3989	1.486	5.419
RL28	74	4051	1.584	5.694
RL29	66	4456	1.540	6.024
RL30	60	4356	1.671	5.313
RL32	60	4206	1.463	4.997
RL33	49	4604	1.678	4.943
RL34	45	3195	1.346	4.280
RL35	65	3691	1.502	5.889
RL36	38	3779	1.408	3.103

Table 4.1: Alignment sizes (M) and lengths (L) for proteins of the small (RSXX) and large (RLXX) ribosomal subunit. (P) indicates the average number of paralogs per species and (S) the standard deviation of this number.

	L	M	P	S
TrpA	259	10220	4.457	32.604
TrpB	399	46557	16.992	145.826
TrpC	254	10323	4.536	39.868
TrpD	337	17582	7.130	59.693
TrpE	460	28173	11.749	124.933
TrpF	197	8713	4.122	32.400
TrpG	192	78265	24.713	187.331

Table 4.2: Alignment sizes (M) and lengths (L) for proteins of the Trp Operon. (P) indicates the average number of paralogs per species and (S) the standard deviation of this number.

	RL2	RL3	RL4	RL5	RL6	RL7	RL8	RL9	RL10	RL11	RL12	RL13	RL14	RL15	RL16	RL17	RL18	RL19	RL20	RL21
RL2		2914	2537	2458	2224	2825	2833	2491	2457	2839	2664	2342	2511	2748	2462	2373	2515	2842	2109	1740
RL3			2947	2719	2430	3109	3223	2531	2680	3097	2922	2577	2992	2694	2645	2686	2659	3213	2123	1907
RL4				2411	1837	2719	2812	2214	2314	2802	2528	2463	2522	2319	2064	2354	2182	2765	1774	1468
RL5					2231	2613	2736	2508	2607	2623	2410	2532	2517	2381	2221	2699	2142	2657	2127	1743
RL6						2206	2251	2216	2200	2204	2041	2117	1938	2169	2430	2226	2590	2263	2116	1931
RL7							3001	2469	2580	2914	3172	2452	2753	2650	2414	2524	2483	2937	2089	1711
RL8								2539	2782	3098	2831	2654	3004	2707	2494	3037	2497	3402	2114	1786
RL9									2466	2564	2348	2400	2284	2383	2204	2469	2188	2489	2103	1755
RL10										2579	2423	2460	2443	2378	2212	2711	2144	2784	2100	1734
RL11											2810	2618	2849	2694	2417	2604	2497	3008	2083	1729
RL12												2295	2646	2507	2224	2369	2303	2828	1925	1542
RL13													2395	2188	2174	2502	2117	2564	2060	1712
RL14														2420	2169	2510	2398	2920	1804	1529
RL15															2417	2348	2461	2679	2115	1753
RL16																2212	2532	2474	2116	1925
RL17																	2127	2918	2097	1735
RL18																		2484	2043	1867
RL19																			2096	1767
RL20																				1683
RL21																				
	2520	2740	2370	2439	2191	2612	2726	2348	2424	2633	2463	2349	2453	2422	2306	2447	2328	2689	2036	1738

Table 4.3: Matched Alignment Sizes for Small Ribosomal Subunit, at threshold 100

	RL2	RL3	RL4	RL5	RL6	RL7	RL8	RL9	RL10	RL11	RL12	RL13	RL14	RL15	RL16	RL17	RL18	RL19	RL20	RL21
RL2		2594	2143	2343	2149	2608	2611	2342	2333	2592	2379	2095	2256	2533	2318	2303	2311	2599	2051	1692
RL3			2219	2373	2371	2615	2628	2363	2348	2579	2406	2097	2267	2535	2506	2341	2444	2656	2057	1871
RL4				1895	1722	2178	2140	1893	1888	2117	2010	1707	1886	2072	1877	1858	1877	2146	1653	1394
RL5					2156	2356	2364	2344	2333	2322	2156	2078	1984	2313	2160	2320	2084	2319	2069	1707
RL6						2135	2189	2153	2146	2134	1960	2063	1840	2138	2376	2150	2251	2180	2071	1879
RL7							2617	2327	2326	2596	2494	2088	2267	2536	2304	2304	2310	2605	2043	1665
RL8								2338	2341	2623	2379	2113	2302	2570	2385	2336	2333	2669	2057	1743
RL9									2323	2324	2156	2071	1996	2315	2155	2303	2102	2320	2057	1700
RL10										2327	2153	2090	1996	2301	2159	2302	2096	2330	2055	1693
RL11											2386	2091	2280	2559	2318	2291	2318	2596	2040	1685
RL12												1920	2145	2324	2094	2120	2069	2395	1866	1508
RL13													1806	2077	2091	2052	2054	2086	2003	1661
RL14														2213	2037	1980	2109	2290	1735	1485
RL15															2316	2287	2304	2539	2043	1697
RL16																2149	2451	2373	2066	1877
RL17																	2077	2321	2047	1687
RL18																		2308	1998	1827
RL19																			2033	1734
RL20																				1617
RL21																				
	2329	2383	1930	2193	2109	2335	2355	2189	2186	2325	2154	2013	2046	2299	2211	2170	2175	2342	1977	1691

Table 4.4: Matched Alignment Sizes for Small Ribosomal Subunit, at threshold 0 (matching by uniqueness)

	RL2	RL3	RL4	RL5	RL6	RL9	RL11	RL13	RL14	RL15	RL16	RL17	RL18	RL19	RL20	RL21	RL22	RL23	RL24	RL25	RL27	RL28	RL29	RL30	RL32	RL33	RL34	RL35	RL36
RL2		2699	2720	2875	2824	2142	2505	2461	3077	2658	3101	2438	2672	2190	2509	2174	2957	3075	2435	1739	2164	1932	2904	2471	2296	2163	1970	2094	2328
RL3			2789	2626	2923	2149	2382	2395	2873	2604	2626	2456	2649	2184	2161	2164	3132	2661	2338	1733	2184	1964	2591	2290	2033	1902	1993	2108	1984
RL4				2639	2709	2167	2407	2418	2637	2676	2647	2438	2871	2209	2167	2168	2788	2695	2805	1747	2195	1962	2652	2333	2040	1894	2001	2134	2011
RL5					2902	2232	2492	2498	2799	2692	3134	2608	2775	2312	2327	2309	2688	3035	2483	1773	2299	2014	2744	2389	2164	1945	2084	2203	2136
RL6						2216	2551	2506	3043	2768	2839	2651	2828	2283	2277	2275	2990	2773	2495	1785	2286	2005	2828	2455	2114	1937	2039	2207	2101
RL9							2154	2156	2168	2161	2174	2191	2238	2283	2224	2237	2153	2165	502	1792	2259	2025	2230	1877	2099	1810	2106	2190	1768
RL11								2422	2492	2375	2468	2223	2499	2217	2179	2174	2370	2539	2314	1732	2187	1973	2457	2131	2040	2024	1991	2133	1777
RL13									2491	2482	2498	2246	2493	2208	2197	2198	2340	2482	1127	1755	2217	1980	2445	2110	2053	1852	1999	2155	1800
RL14										2643	3080	2465	2752	2232	2574	2227	3166	3012	2328	1735	2208	1989	2606	2241	2345	2181	2003	2126	2345
RL15											2616	2509	2740	2189	2169	2160	2714	2700	2354	1760	2196	1964	2706	2388	2024	1848	1970	2109	2040
RL16												2488	2730	2240	2564	2229	2812	3348	2314	1759	2213	1991	2610	2259	2372	2191	2012	2142	2325
RL17													2755	2385	2176	2341	2465	2530	2207	1726	2380	2146	2689	2180	2190	1917	2131	2181	2302
RL18														2422	2223	2369	2734	2739	2934	1772	2417	2170	2886	2454	2227	1975	2176	2216	2193
RL19															2331	2437	2188	2262	580	1774	2507	2277	2434	1913	2361	1906	2225	2315	1948
RL20																2311	2483	2518	411	1787	2297	2011	2248	1868	2450	2161	2048	2477	2152
RL21																	2202	2242	542	1754	2692	2163	2380	1887	2258	1890	2177	2259	1913
RL22																		2942	2380	1739	2208	1970	2595	2251	2297	2160	1989	2120	2294
RL23																			2405	1748	2254	2007	2727	2397	2381	2221	2044	2152	2337
RL24																				391	503	528	2459	2093	449	1111	522	437	1468
RL25																					1770	1595	1745	1547	1649	1564	1598	1761	1362
RL27																						2234	2427	1915	2300	1928	2232	2295	1931
RL28																							2148	1719	2185	1935	2015	2039	1750
RL29																								2584	2223	1957	2163	2251	2160
RL30																									1765	1579	1732	1851	1738
RL32																										2183	2074	2130	2132
RL33																											1741	1819	1921
RL34																												2089	1779
RL35																													1800
RL36																													
	2485	2378	2390	2471	2486	2067	2257	2214	2494	2365	2492	2336	2497	2172	2189	2148	2469	2514	1604	1664	2168	1953	2459	2086	2101	1918	1961	2064	1993

Table 4.5: Matched Alignment Sizes for Large Ribosomal Subunit, at threshold 100

	RL2	RL3	RL4	RL5	RL6	RL9	RL11	RL13	RL14	RL15	RL16	RL17	RL18	RL19	RL20	RL21	RL22	RL23	RL24	RL25	RL27	RL28	RL29	RL30	RL32	RL33	RL34	RL35	RL36
RL2		2144	2173	2333	2307	2079	2277	2286	2568	2115	2547	2041	2313	2120	2325	2095	2325	2552	217	1698	2100	1895	2257	1964	2182	1875	1918	2052	1900
RL3			2139	2179	2162	2087	2152	2169	2165	2102	2168	2033	2178	2126	2109	2095	2112	2177	177	1703	2115	1899	2140	1851	1965	1651	1933	2068	1648
RL4				2205	2188	2099	2175	2191	2190	2118	2197	2045	2207	2140	2119	2102	2117	2193	189	1704	2126	1912	2170	1865	1978	1663	1944	2075	1652
RL5					2425	2176	2316	2319	2388	2151	2370	2162	2379	2255	2257	2221	2150	2369	221	1735	2235	1960	2394	2038	2093	1727	2005	2161	1771
RL6						2164	2307	2310	2344	2149	2337	2134	2368	2221	2214	2187	2130	2324	221	1725	2204	1949	2379	2016	2045	1694	1981	2144	1713
RL9							2088	2106	2110	2095	2122	2142	2178	2232	2176	2187	2102	2119	167	1735	2224	1967	2181	1824	2059	1697	2018	2147	1720
RL11								2305	2317	2117	2300	2061	2319	2143	2129	2115	2109	2299	219	1693	2133	1922	2289	1978	1994	1672	1938	2074	1668
RL13									2312	2130	2306	2064	2323	2152	2141	2129	2121	2305	205	1713	2145	1918	2292	1984	1988	1670	1952	2090	1668
RL14										2146	2600	2089	2349	2165	2392	2155	2388	2606	217	1710	2151	1940	2318	2012	2259	1941	1961	2085	1998
RL15											2166	2062	2162	2137	2120	2107	2120	2132	181	1713	2127	1917	2110	1842	1975	1657	1936	2073	1653
RL16												2089	2335	2171	2370	2144	2347	2539	216	1724	2155	1935	2304	2001	2226	1902	1963	2095	1931
RL17													2302	2346	2146	2280	2049	2121	222	1677	2337	2099	2305	1798	2155	1724	2094	2144	1802
RL18														2366	2177	2310	2144	2381	293	1731	2366	2127	2521	2056	2177	1774	2126	2166	1821
RL19															2260	2370	2138	2208	235	1748	2438	2156	2392	1875	2248	1816	2190	2260	1889
RL20																2231	2345	2379	174	1737	2226	1960	2197	1840	2294	1952	2003	2177	2011
RL21																	2125	2179	224	1720	2367	2089	2317	1838	2191	1777	2130	2203	1849
RL22																		2373	170	1707	2132	1917	2107	1810	2215	1908	1946	2081	1936
RL23																			227	1716	2187	1955	2351	2006	2289	1957	1988	2107	1999
RL24																				116	238	211	288	169	224	180	207	182	195
RL25																					1733	1539	1713	1517	1602	1474	1566	1704	1323
RL27																						2164	2376	1863	2243	1814	2194	2236	1879
RL28																							2109	1654	2036	1697	1980	1989	1685
RL29																								2052	2188	1771	2132	2210	1835
RL30																									1724	1427	1711	1821	1441
RL32																										1988	2048	2084	2033
RL33																											1655	1708	1693
RL34																												2051	1730
RL35																													1763
RL36																													
	2095	1980	1996	2107	2084	2000	2040	2046	2138	1975	2127	2019	2141	2100	2088	2062	2040	2144	207	1613	2090	1878	2132	1785	2018	1695	1904	1998	1722

Table 4.6: Matched Alignment Sizes for Large Ribosomal Subunit, at threshold 0 (matching by uniqueness)

P1	P2	tr=100	tr=0
TrpC	TrpG	4272	18
TrpE	TrpF	2519	830
TrpA	TrpD	2823	743
TrpD	TrpG	6249	28
TrpB	TrpF	3643	95
TrpB	TrpD	3737	95
TrpB	TrpG	8053	41
TrpE	TrpG	5324	8
TrpD	TrpF	2819	695
TrpC	TrpF	3825	1578
TrpA	TrpC	3198	1546
TrpC	TrpD	3392	748
TrpA	TrpF	3357	1433
TrpA	TrpE	3118	905
TrpD	TrpE	2681	482
TrpB	TrpC	3326	82
TrpB	TrpE	3911	53
TrpC	TrpE	2976	930
TrpF	TrpG	3635	32
TrpA	TrpB	4374	95
TrpA	TrpG	4646	22

Table 4.7: Matched Alignment Sizes for Trp for different matching thresholds (threshold 0 corresponds to matching by uniqueness)

P1	P2	Score	Interacting
RS10	RS14	0.618890	1
RS18	RS6	0.422457	1
RS14	RS3	0.394753	1
RS10	RS9	0.347508	1
RS13	RS19	0.317640	1
RS13	RS21	0.306248	0
RS11	RS21	0.296700	1
RS14	RS19	0.291335	1
RS12	RS21	0.290965	0
RS16	RS4	0.287438	0
RS21	RS7	0.287102	0
RS13	RS15	0.284783	0
RS12	RS16	0.283105	0
RS19	RS21	0.282142	0
RS10	RS18	0.279595	0

P1	P2	Score	Interacting
RL20	RL21	0.576795	1
RL14	RL19	0.514107	1
RL15	RL35	0.440323	1
RL15	RL21	0.439233	1
RL17	RL32	0.425920	1
RL20	RL32	0.421733	1
RL23	RL29	0.414060	1
RL13	RL20	0.334348	1
RL19	RL3	0.328640	1
RL30	RL34	0.326368	0
RL22	RL32	0.324540	1
RL16	RL36	0.318915	1
RL16	RL33	0.313083	0
RL33	RL36	0.307188	0
RL27	RL34	0.306283	0

Table 4.8: Ordered List of Interaction Candidates SRU (left) and LRU (right) based on plmDCA scores; the fourth column indicates whether the protein pair is indeed interacting

P1	P2	Score	Interacting
RS10	RS9	1.123465	1
RS10	RS14	1.102428	1
RS12	RS21	1.079407	0
RS13	RS18	1.029537	0
RS14	RS17	1.001716	0
RS12	RS15	0.997813	0
RS18	RS6	0.963688	1
RS11	RS13	0.943144	0
RS19	RS21	0.942921	0
RS15	RS18	0.938286	0
RS14	RS15	0.933949	0
RS13	RS15	0.933337	0
RS13	RS19	0.918528	1
RS18	RS21	0.918101	1
RS10	RS13	0.917482	0

P1	P2	Score	Interacting
RL20	RL21	1.665182	1
RL14	RL19	1.430611	1
RL15	RL21	1.333611	1
RL15	RL35	1.134808	1
RL23	RL29	1.086992	1
RL20	RL32	1.037364	1
RL22	RL32	1.029724	1
RL30	RL34	1.008776	0
RL17	RL32	1.002790	1
RL34	RL36	0.983223	0
RL21	RL2	0.977507	0
RL21	RL34	0.958441	0
RL18	RL34	0.942494	0
RL36	RL6	0.925895	1
RL33	RL36	0.898444	0

Table 4.9: Ordered List of Interaction Candidates SRU (left) and LRU (right) based on Gaussian scores; the fourth column indicates whether the protein pair is indeed interacting

TrpA	TrpB	0.375
TrpE	TrpG	0.295
TrpA	TrpC	0.167
TrpA	TrpF	0.162
TrpC	TrpF	0.146
TrpA	TrpD	0.144
TpC	TrpD	0.141
TrpB	TrpF	0.136
TrpC	TrpE	0.135
TrpD	TrpF	0.135
TrpB	TrpC	0.132
TrpA	TrpE	0.126
TrpC	TrpG	0.121
TrpB	TrpD	0.120
TrpE	TrpF	0.115
TrpD	TrpE	0.107
TrpF	TrpG	0.107
TrpA	TrpG	0.104
TrpD	TrpG	0.100
TrpB	TrpE	0.096
TrpB	TrpG	0.071

Table 4.10: Ordered List of Interaction Scores for the Trp Operon based on plmDCA scores

SRU Intra-Protein		
	SEP=0	SEP=5
RS2	2337	1610
RS3	2217	1494
RS4	1728	1152
RS5	1684	1175
RS6	1002	666
RS7	1494	982
RS8	1334	903
RS9	1240	799
RS10	878	557
RS11	1220	822
RS12	1136	731
RS13	1024	623
RS14	790	440
RS15	823	489
RS16	685	436
RS17	733	487
RS18	482	293
RS19	748	482
RS20	792	464
RS21	297	110
SUM:	22644	14715

SRU Inter-Protein		
RS2	RS5	4
RS2	RS8	3
RS3	RS5	17
RS3	RS10	105
RS3	RS14	209
RS4	RS5	84
RS5	RS8	120
RS6	RS18	150
RS7	RS9	19
RS7	RS11	46
RS8	RS12	12
RS8	RS17	28
RS9	RS10	28
RS9	RS14	7
RS10	RS14	150
RS11	RS18	20
RS11	RS21	199
RS12	RS17	34
RS13	RS19	80
RS14	RS19	50
RS18	RS21	36
SUM:		1401
FRACTION	SEP=0	0.058
FRACTION	SEP=5	0.087

Table 4.11: Left table: number of intra-protein contacts below 8Å of all residues (SEP=0 column), and considering only those with a distance on the sequence of at least 5 residues (SEP = 5 column) for the SRU. Right table: number of inter-protein contacts below 8Åfor the SRU. Fractions are defined as $\frac{\#Intra}{\#Intra+\#Inter}$ where $\#Inter$ is computed assuming SEP=0,5 respectively.

LRU Intra-Protein		
	SEP=0	SEP=5
RL32	324	157
RL33	399	256
RL34	303	145
RL35	495	268
RL36	332	208
RL2	2687	1801
RL3	1931	1263
RL4	1869	1199
RL5	1887	1257
RL6	1811	1217
RL9	1360	855
RL11	1390	903
RL13	1464	959
RL14	1266	869
RL15	920	481
RL16	1343	915
RL17	1194	767
RL18	1150	777
RL19	1043	669
RL20	1045	600
RL21	915	600
RL22	1085	720
RL23	735	461
RL24	386	233
RL25	893	597
RL27	692	442
RL29	538	303
RL30	511	321
RL28	587	351
SUM:	30555	19594

LRU Inter-Protein		
RL32	RL17	78
RL32	RL20	17
RL32	RL22	73
RL33	RL35	21
RL35	RL15	149
RL35	RL27	1
RL36	RL6	10
RL36	RL16	1
RL3	RL13	20
RL3	RL14	34
RL3	RL17	21
RL3	RL19	123
RL4	RL15	83
RL4	RL20	6
RL9	RL28	63
RL13	RL20	118
RL13	RL21	8
RL14	RL19	191
RL15	RL20	2
RL15	RL21	24
RL16	RL25	53
RL16	RL27	9
RL17	RL22	12
RL18	RL27	12
RL20	RL21	229
RL23	RL29	81
SUM:		1439
FRACTION	SEP=0	0.045
FRACTION	SEP=5	0.068

Table 4.12: Left table: number of intra-protein contacts below 8Å of all residues (SEP=0 column), and considering only those with a distance on the sequence of at least 5 residues (SEP = 5 column) for the LRU. Right table: number of inter-protein contacts below 8Åfor the LRU. Fractions are defined as $\frac{\#Intra}{\#Intra+\#Inter}$ where $\#Inter$ is computed assuming SEP=0,5 respectively.

Original Rank	Inferred Rank
1	101
2	13806
3	10658
4	64
5	4
6	9575
7	1
8	15890
9	6712
10	1035
7	1
32	2
41	3
5	4
11	5
11473	6
22464	7
53	8
1877	9
26	10

Table 4.13: Original vs. inferred rank for the 10 largest original inter-protein residue interaction scores and the 10 largest inferred inter-protein residue interaction scores

5 Some Preliminary Results and Outlook: Energy Landscapes and Folding Prediction

In this short Section, we will introduce the topic of fitness landscapes. We will show how the machinery of the preceding sections can be used to reason in this context and show some preliminary results based on data published in [85], that will hopefully lead to further research. This also means that this section is necessarily more sketchy than the other ones, and contains a considerable amount of speculation. The outlook, however, is in our opinion encouraging and at the end we will try to give a short perspective on where one could go from here.

5.1 Energy Landscapes and Mutation Analysis

The improvement of genotype-fitness maps has emerged as an important field of biological sequence analysis [21] and has given major contributions for example in the design of immunogenes for HIV [32], in the assessment of the significance of protein mutations in cancer [80] and the exploration of evolutionary pathways between homologs [94].

A central factor that makes fitness landscapes interesting is epistasis [17], which means that the effect of a mutation is not independent of the background in which it occurs. Thus it seems that global probabilistic models, as used extensively in this thesis, are suited to describe the fitness landscape of proteins. In fact, in many ways the Potts Model of Equation 2.30 can be seen as one of the most simple models to take epistasis into account. Additionally, many works already use models very similar or even identical to our model for modeling the fitness landscape [32].

Another encouraging factor is that more and more large-scale data is being published. Mutant libraries of modern experiments contain up to several hundred thousand sequences [75], and statistical methods and models are needed to analyze this data.

Recently, it has been shown in the context of beta lactamase TEM-1 that the energy of the Potts Model used in DCA (see Equation 2.30), trained on homologous sequences, is very well correlated to the capability of the proteins to confer antibiotic drug resistance (measured as minimum inhibitory concentration) [33] and in fact outperforms established methods.

Given that the methods was originally designed to capture structural information, it is not unreasonable to speculate that most of this success in predicting fitness (or proxies thereof) of proteins is actually a prediction whether the protein will fold correctly to exert its function (not excluding, of course, that the pairwise distribution might capture other biologically relevant information).

Another interesting experimental information would therefore be the binary information whether a given protein sequences folds or not. This information could be compared to the energy of the sequences in the Potts Model, and if a good correlation

is observed, the Potts Model could be used to predict the folding properties of new sequences. In the next sections we will try to lay the groundwork for this, based on the work and data found in [85]. There the authors created, based on a set of homologous proteins of the WW-domain (a small protein domain implicated in protein-protein interaction [46]), a set of new sequences and assessed their folding properties. We used this data as a test set and will show positive evidence that the prediction of folding properties is indeed possible in the context of DCA.

5.2 Preliminary Results on the WW-Domain

5.2.1 Creating Artificial Protein Sequences by Simulated Annealing

We are now going to report some ideas and results presented in [85]. There, artificial protein sequences for the WW-domain (Pfam PF00397) were created based on a *MCMC* algorithm, using an MSA of homologous sequences. This is of interest, since the technique presented there allows to sample from a distribution for which one does not know the parameters explicitly, based on the MSA. However, as shown in [8], this distribution is the Potts Model of Equation 2.30 which we used extensively in the preceding sections. This led us to consider this model to analyze the specific data published in [85].

The work presented in [85] is concerned with the *creation* of artificial proteins. As described in Section 1.2, a *Multiple Sequence Alignment* (MSA) of the protein sequences of a protein family contains many (several thousand in the cases interesting for us) amino acid sequences that fold into a similar structure. The authors now ask whether one can produce new protein sequences with a similar structure as the ones in the MSA with solely the information given in the MSA. Furthermore, they pose the question *what* information in the MSA is important for such a method of protein design.

The underlying idea is that covariances are enough. This means that in order to create a new set of protein sequences similar in structure to the ones in MSA Z, one could create a new MSA $\hat{Z}$ in which the covariances

$$C_{ij}(a,b) = f_{ij}(a,b) - f_i(a)f_j(b) \tag{5.1}$$

are the same as in Z. We recall that with $f_i(a)$ we mean the frequency of observing amino acid a at residue i in the data and with $f_{ij}(a,b)$ the frequency of co-occurrence of amino acid a at residue i with amino acid b at residue j, see Section 2.2.1 and specifically Equation 2.37. Of course, even if we assume that the new MSA $\hat{Z}$ has the same number of sequences as the old one Z, there are many possible $\hat{Z}$ that have the correct covariances.

The authors therefore use a probabilistic method to generate one single new MSA $\hat{Z}$, using *MCMC* with *simulated annealing*.

The basic idea is to take the original multiple sequence alignment Z and calculate the covariance matrix, denoted by C^{emp}.

Now the authors shuffle all symbols *within* the columns randomly. This means that for a fixed i the symbols Z_i^m get assigned a random new m. The new MSA is denoted by $\hat{Z}$.

Notice that this leaves the frequency $f_i(a)$ of any symbol a at some position i unchanged but leads to new covariances $\hat{C}$ that are approximately zero.

For the new covariance matrix $\hat{C}$ an energy function χ^2 is now defined that measures the distance to the original covariance matrix C:

$$\chi^2\left(\hat{C}\right) = \sum_{i<j}^{q,q} \sum_{a,b} \left(\hat{C}_{ij}(a,b) - C_{ij}(a,b)\right)^2 . \tag{5.2}$$

This energy function reaches its minimum if and only if the covariances in the new alignment $\hat{Z}$ are the same as in the old alignment Z. In order to arrive at such a $\hat{Z}$ an *MCMC* method with *simulated annealing* is employed.

The basic move is a random exchange of two symbols within a column. This means that randomly a position i and two sequences m_1 and m_2 are chosen. Exchanging the symbols $Z_i^{m_1}$ and $Z_i^{m_2}$ for each other leads to a new covariances matrix $\hat{\hat{C}}$ and therefore to an energy change

$$\Delta\chi^2 = \chi(\hat{\hat{C}})^2 - \chi(\hat{C})^2 \tag{5.3}$$

The move is accepted if $\Delta\chi^2 < 0$ or if for a random number $\xi \in [0,1]$

$$\xi < \exp\left(-\frac{1}{T}\Delta\chi^2\right) \tag{5.4}$$

The temperature T is lowered during the simulation in order to find states with smaller and smaller energies.

At the end of the annealing procedure, the MSA $\hat{Z}$ should have similar covariance to the original MSA Z but will generally differ in any higher-order moments.

The most important result of [85] is that many sequences produced by this technique indeed fold into a structure similar to the proteins in the original alignment. This is somewhat surprising because understanding of protein folding and protein structure prediction from sequence information alone is one of the outstanding problems in biophysics and bioinformatics [22]. Here the authors show that even though there is no general method known on how to arrive from the sequence of a protein to its structure, the creation of a new sequence with a given structure is a relatively easy

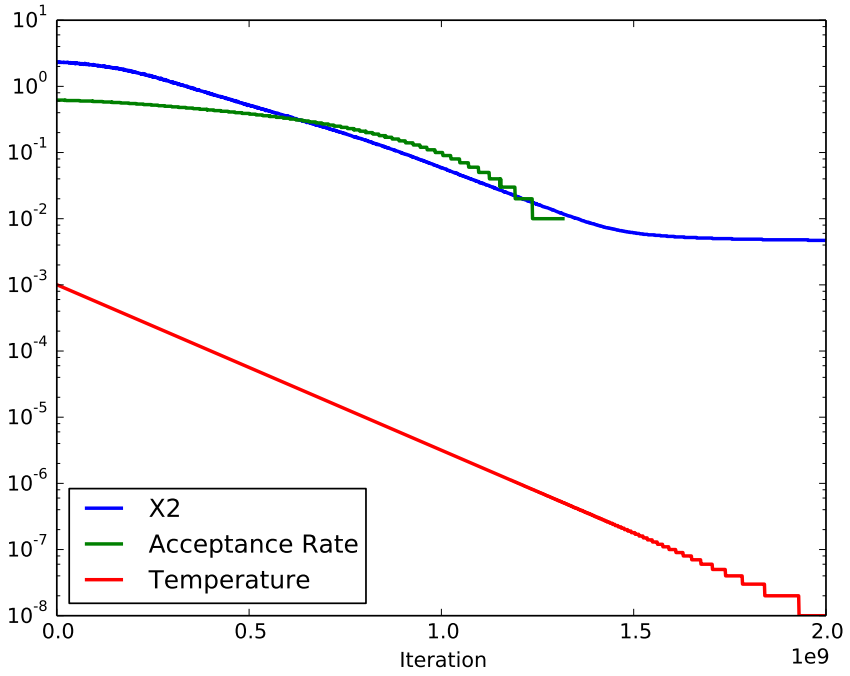


Figure 5.1: Characteristics of the MC algorithm run on the protein PF00397. The x-axis represents the iterations (MC-Steps) and the y-axis the numerical values of the characteristics: The acceptance rate (green), the temperature (red) and the energy χ^2 called X2 in the plot (blue). Notice the log-scale on the y-axis. The acceptance rate stops being plotted when it drops below 10^{-2}.

task (with a reasonable success rate). Additionally, a quite low amount of extracted evolutionary information is necessary for this task: covariances are enough.

In a follow-up paper, the authors show that many of these designed proteins not only have a similar structure as the original proteins but are also functional [82].

Our version of the algorithm can be found in Algorithm 1, and some characteristics a run on the WW-domain (Pfam PF00337) with $2 \cdot 10^9$ MC steps in Figure 5.1.

An interesting aspect is the problem that for low sequence numbers the algorithm simply reproduces the original alignment: For small M, the energy χ^2 drops to zero after some iterations and the resulting MSA $\hat{Z}$ is equal to the original alignment Z. This works for subsets of size of about $M^{subset} \approx 150$, so for that number of sequences the $\binom{N}{2}q^2$ numbers defining the covariance matrix determine the data probably *uniquely*.

The problem can be quantified by a measure of sequence similarity between Z and $\hat{Z}$. For this, for every sequence of $\hat{Z}$ the minimal Hamming Distance to any sequence in the original alignment Z was determined. The mean θ of all these values gives the average Hamming Distance of the sequences in $\hat{Z}$ to their most similar sequence in Z. The quantity $\tau = 1 - \frac{\theta}{N}$ then represents the average *maximal sequence similarity*. The authors in [85] report this average maximal sequence similarity for the data-set they produced to be $\theta = 0.58$ with a standard deviation of 0.07, while for our data-se we found $\theta = 0.74$ with a standard deviation of 0.1. These differences could arise from a different original data-set (we used the publicly available Pfam data-set [77]). Given that small M result in a reproduction of the original alignment and therefore at $\theta = 1.0$, a larger data-set would probably result in a lower θ.

Another explanation is that maybe my annealing procedure finds lower energy states than the one of the authors. Annealing too fast or stopping the algorithm before some sort of saturation has been reached would also result in lower θ. Given that no details of the annealing procedure of the authors are given, this is mere speculation, though.

Algorithm 1 Pseudo-code version of the algorithm presented in Sec. 2

Require: Z,$\hat{Z}$,dT,T_0,$iter_{max}$,$iter_T$
  $C \leftarrow get\_C(Z)$ ▷ The get_C function calculates the quantities defined by Eq. 5.1
  $\hat{C} \leftarrow get\_C(\hat{Z})$
  $\chi^2 \leftarrow get\_\chi^2(C, \hat{C})$ ▷ The get_χ^2 function calculates the quantity defined in Eq. 5.2
  $iter \leftarrow 0$
  while $iter < iter_{max}$ **do**
    $i \leftarrow rand(1 : N)$
    $m_1 \leftarrow rand(1 : M)$
    $m_2 \leftarrow rand(1 : M)$
    $\Delta\chi^2 \leftarrow get\_\Delta\chi^2(C, \hat{C}, Z, \hat{Z}, i, m_1, m_2)$ ▷ The get_$\Delta\chi^2$ calculates the quantity defined in Eq. 5.3
    if $rand() < \exp(delta\_chi2)$ **then**
      $\hat{Z}_i^{m_2} \leftrightarrow \hat{Z}_i^{m_1}$ ▷ The double-headed arrow means exchange of the values
      $\hat{C} \leftarrow get\_C(\hat{Z})$
      $\chi^2 \leftarrow \chi^2 + \Delta\chi^2$
    end if
    $iter \leftarrow iter + 1$
    if $iter_T \mod iter == 0$ **then** ▷ Every $iter_T$ the temperature T gets changed by dT
      $T \leftarrow T - dT$
    end if
  end while

5.2.2 Connection to the Generalized Potts Model

The procedure described above leads to protein sequence out of which a significant percentage (around 30% according to the data in [85]) fold correctly. A problem is that given the nature of the procedure there is no a priori measure *which* sequences are candidates for good folding. The idea that is presented in this Section is to use the probability of the Potts Model of Equation 2.30 as a score for correct folding.

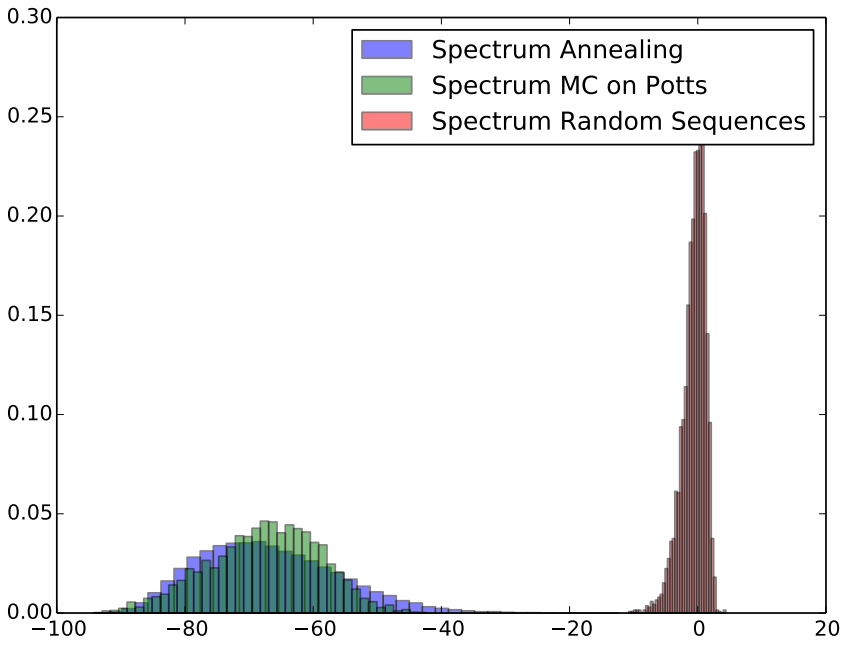
A good reason why this should work is found in [8]. Here, the authors show that for $M \to \infty$ and $T \to 0$ the distribution of the sequences in the new alignment $\hat{Z}$ should indeed converge to the maximum entropy model fitted to the covariances of the original alignment Z. In retrospect, this is not surprising. The equality of the covariances is the only constraint put on the alignment $\hat{Z}$, and thus one would expect intuitively the least constraint model with respect to all other characteristics. But this is nothing else than the maximum entropy model (see Section 2.2.1 for a discussion of the maximum entropy principle). The equality is remarkable since it seems possible to sample from the maximum entropy distribution *without knowing it*. Given that maximum entropy distributions are very popular in theoretical biology, this technique could be potentially useful [81].

A way to test this is to look at the energy spectrum of the two distributions with respect to the Potts Model. To this end, we inferred the Potts Model of Equation 2.30 using plmDCA (see Section 2.2.3) using the original alignment Z. Using this model, we can calculate the energy of the sequences coming out of the annealing procedure $\hat{Z}$ and another set of sequences, sampled from the same model. Figure 5.2 shows that the two energy distributions are virtually the same, so we are confident in the result of [8] (we added the spectrum of a set of random sequences, created by exchanging amino acids in the alignment Z randomly).

Knowing the numerical values of the parameters J and h of the model 2.30 has nonetheless a strong advantage since they allow assign an energy to every sequence. Figure 5.3 shows that the lower energy regime is indeed strongly enriched for folding sequences, when the data-set of [85] is analyzed. The Figure has two parts: The lower parts shares the energy axis with the upper part and shows a bar for every sequence in the 4 data-sets of [85]: 1) *CC sequences* are created with the same algorithm as presented above. 2) *IC sequences* are random sequences from a distribution that reproduces the same single-site frequencies as the original alignment Z. 3) *R sequences* are random sequences only reproducing the overall amino acid distribution in the alignment 4) *NAT sequences* are natural sequences of the WW-domain. Grey bars correspond to sequences that do not fold according to the criteria in [85], while red bars do fold. Notice that some natural sequences do not fold. This is due to experimental problems and can used as a rough indication on how sure we can be of the results on the other sequences sets.

The upper part shows the energy spectra of different data sets in a Potts Model (see Equation 2.30 inferred with plmDCA (see Section 2.2.3). The data sets mirror the

Figure 5.2: Energy spectra for different data sets, normalized. X-axis: Energy, Y-axis: Frequency of observation



ones in the lower part: The blue spectrum stems from sequences sampled from the inferred Potts Model, i.e. from a distribution that reproduces (approximately) the covariances in the Pfam alignment. The green spectrum stems from an independent site model, i.e. one that reproduces only the single site amino acids frequencies. The red spectrum corresponds to random sequences, which only reproduce the overall amino acid distribution in the alignment.

Notice that only the CC and NAT sets contain any folding sequence. This is to be expected and can be interpreted that interactions between residues, which are not captured by an independent or random model, are important for structural features. It is furthermore interesting that all sequences that fall in an energy region in which the pairwise model still has a large probability but the independent model has a low probability fold. If one therefore would search for good candidates for folding in a set of sequences, one could select those sequences that would be typical for a pairwise model but atypical for an independent model.

We can summarize this finding by saying that the Potts Hamiltonian is a good predictor for the folding properties of sequences of the WW-domain.

5.3 Outlook

We have shown here that this kind of folding prediction works well for the WW-domain, but more data is needed to asses the performance of the model in folding prediction more rigorously. This data is expected to be available in the near future, when more and maybe large-scale experiments probing energy landscapes become available.

This also leads to a new and different research direction. If the Generalized Potts Model is good in predicting whether some sequence will fold or not, it should be also interesting to synthesize sequences directly from samples from this distribution. This would enable researchers to correlate the probability of a sequence in the Potts Model in an unbiased way with their performance in the cell. This would allow to probe in a more detailed way to what extend the pairwise distribution is actually a good model for the fitness landscape. To answer this question, fitness data from sequences that do not follow the Potts Model distribution may be misleading, since being a good predictor for folding does not necessarily imply that the model is also a good generator.

If such experiments meet with success, this would lever the applicability of the model from its original purpose of contact prediction to protein design [57].

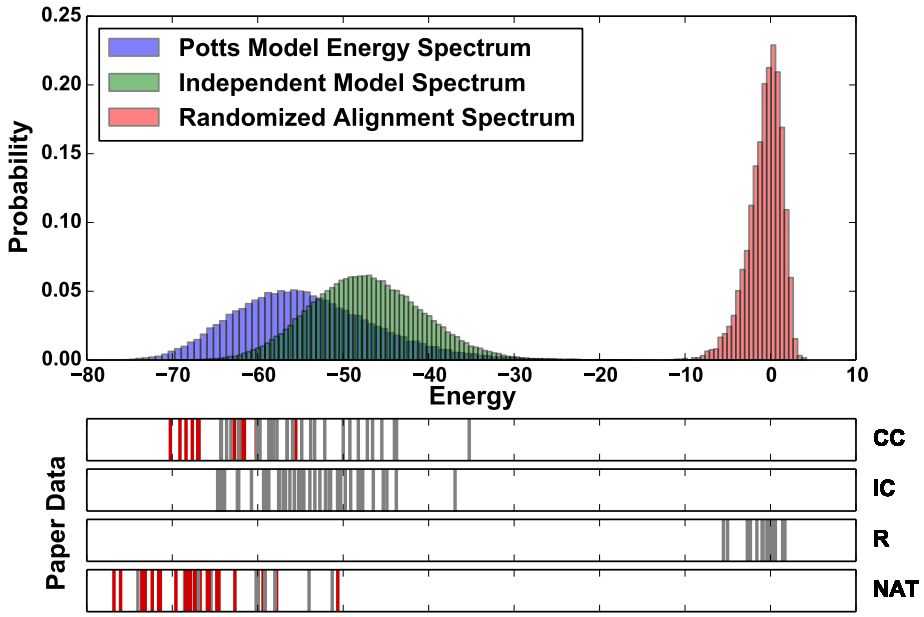


Figure 5.3: **Upper part**: Energy spectra of different sequence sets in the Potts Model inferred on the Pfam-alignment PF00397. Potts Model Energy Spectrum: Sequences drawn from the same Potts Model; Independent Model Spectrum: Sequences drawn from a model conserving single site frequencies of the original alignment; Randomized Alignment Spectrum: Sequence resulting from a random shuffling of all amino acids in the original alignment. **Lower Part**: Energies of sequences from [85]. Red bars indicate folding sequences according to the classification found there. Different data sets are explained in the main text.

6 Synopsis and Conclusion

In this work we have touched three major areas related to the statistical mechanics approach to protein sequence data.

The first area, contact prediction, is the one that is responsible for the recent interest in applying the kind of models used in this thesis to protein data. In this context we have presented in Section 3.1 the work found in [4], which presents a fast and accurate approach for contact prediction based on a Gaussian approximation. Such fast approaches are needed since they make excellent building blocks for the emerging meta-methods that begin to dominate for example in CASP [67]. In Section 3.2 another work in the context of contact prediction was presented. It considers the possibility to extend the pairwise model of Equation 2.30 with higher-order terms [29]. This specific work is concerned with modeling long gap stretches, which appear in MSAs as artifacts of the alignment process, and creates an encouraging outlook since it shows that going beyond the pairwise model can lead to significantly better performance. It is clear that the task at hand is now to find a process by which to choose which *other* many-body interactions are advantageous to include.

The second area, protein-protein interaction network inference, represents the major part of this thesis and the results presented here are based on [30]. The excellent performance of the method in the systems tested is encouraging and motivates further research. A major advance would be to test the approach on a large-scale data set with several thousand possibly interacting proteins. Even though the major tasks in this context (like data generation and validation of the results) might be challenging, a large-scale test set would allow us to assess the performance of the method in detecting unknown protein-protein interactions in a much more precise manner. Also, the biological information coming from the analysis of such a test-set would certainly be itself very interesting and lift the approach from a methodological exercise to actual biological knowledge acquisition.

The third area, mutation analysis and energy landscapes, is a new field in which DCA could give a major contribution. The preliminary data we have shown here (together with recent works such as [33]) give evidence that the energy function of the Potts Model can be used beyond its original purpose (the prediction of residue contacts) and is a good predictor for the folding properties of sequences outside of the set the model was inferred on (exemplified by the WW-domain data in [85]). This could be exploited for example for protein design, the reconstruction of evolutionary pathways between mutants or in medical applications [32].

References

[1] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. Molecular Biology of the Cell (Garland Science, New York, 2002). *There is no corresponding record for this reference*, 1997.

[2] Christian B. Anfinsen. *Studies on the principles that govern the folding of protein chains.* 1972.

[3] Sivaraman Balakrishnan, Hetunandan Kamisetty, Jaime G. Carbonell, Su-In Lee, and Christopher James Langmead. Learning generative models for protein fold families. *Proteins: Structure, Function, and Bioinformatics*, 79(4):1061–1078, April 2011.

[4] Carlo Baldassi, Marco Zamparo, Christoph Feinauer, Andrea Procaccini, Riccardo Zecchina, Martin Weigt, and Andrea Pagnani. Fast and accurate multivariate Gaussian modeling of protein families: predicting residue contacts and protein-interaction partners. *PloS one*, 9(3):e92721, 2014.

[5] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, TN Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[6] Julian Besag. Statistical analysis of non-lattice data. *The statistician*, pages 179–195, 1975.

[7] Jeff Bezanzon, Stefan Karpinski, Viral Shah, and Alan Edelman. Julia: A fast dynamic language for technical computing. In *Lang.NEXT*, April 2012.

[8] William Bialek and Rama Ranganathan. Rediscovering the power of pairwise interactions. *arXiv:0712.4397 [q-bio]*, December 2007. arXiv: 0712.4397.

[9] Maria A Borovinskaya et al. Structural basis for aminoglycoside inhibition of bacterial ribosome recycling. *Nature Struct. Mol. Biol.*, 14(8):727–732, 2007.

[10] Pascal Braun et al. An experimentally derived confidence score for binary protein-protein interactions. *Nature methods*, 6(1):91–97, 2008.

[11] Joseph D. Bryngelson, Jose Nelson Onuchic, Nicholas D. Socci, and Peter G. Wolynes. Funnels, pathways, and the energy landscape of protein folding: a synthesis. *Proteins: Structure, Function, and Bioinformatics*, 21(3):167–195, 1995.

[12] Lukas Burger and Erik Van Nimwegen. Accurate prediction of protein−protein interactions from sequence alignments using a bayesian method. *Molecular Systems Biology*, 4(165):165, 2008.

[13] Lukas Burger and Erik van Nimwegen. Disentangling Direct from Indirect Co-Evolution of Residues in Protein Alignments. *PLoS Computational Biology*, 6(1):e1000633, January 2010.

[14] Ryan R Cheng, Faruck Morcos, Herbert Levine, and José N Onuchic. Toward rationally redesigning bacterial two-component signaling systems using coevolutionary information. *Poc. Natl. Acad. Sci.*, 111(5):E563–E571, 2014.

[15] Simona Cocco and Rémi Monasson. Adaptive Cluster Expansion for Inferring Boltzmann Machines with Noisy Data. *Physical Review Letters*, 106(9), March 2011. arXiv: 1102.3260.

[16] The UniProt Consortium. Uniprot: a hub for protein information. *Nucleic Acids Research*, 43(D1):D204–D212, 2015.

[17] Heather J. Cordell. Epistasis: what it means, what it doesn't mean, and statistical methods to detect it in humans. *Human Molecular Genetics*, 11(20):2463–2468, October 2002.

[18] Angel E. Dago, Alexander Schug, Andrea Procaccini, James A. Hoch, Martin Weigt, and Hendrik Szurmant. Structural basis of histidine kinase autophosphorylation deduced by integrating genomics, molecular dynamics, and mutagenesis. *Poc. Natl. Acad. Sci.*, 109(26):E1733–E1742, 2012.

[19] Thomas Dandekar, Berend Snel, Martijn Huynen, and Peer Bork. Conservation of gene order: a fingerprint of proteins that physically interact. *Trends in biochemical sciences*, 23(9):324–328, 1998.

[20] David de Juan, Florencio Pazos, and Alfonso Valencia. Emerging methods in protein co-evolution. *Nature Reviews Genetics*, 2013.

[21] J. Arjan G. M. de Visser and Joachim Krug. Empirical fitness landscapes and the predictability of evolution. *Nature Reviews Genetics*, 15(7):480–490, July 2014.

[22] Ken A. Dill and Justin L. MacCallum. The protein-folding problem, 50 years on. *Science (New York, N.Y.)*, 338(6110):1042–1046, November 2012.

[23] S.D. Dunn, L.M. Wahl, and G.B. Gloor. Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction. *Bioinformatics*, 24(3):333–340, December 2007.

[24] Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

[25] S R Eddy. Profile hidden markov models. *Bioinformatics*, 14(9):755–763, 1998.

[26] Magnus Ekeberg, Tuomo Hartonen, and Erik Aurell. Fast pseudolikelihood maximization for direct-coupling analysis of protein structure from many homologous amino-acid sequences. *Journal of Computational Physics*, 276:341–356, November 2014.

[27] Magnus Ekeberg, Cecilia Lövkvist, Yueheng Lan, Martin Weigt, and Erik Aurell.

Improved contact prediction in proteins: Using pseudolikelihoods to infer Potts models. *Physical Review E*, 87(1):012707, 2013.

[28] S. Walter Englander and Leland Mayne. The nature of protein folding pathways. *Proceedings of the National Academy of Sciences*, 111(45):15873–15880, 2014.

[29] Christoph Feinauer, Marcin J. Skwark, Andrea Pagnani, and Erik Aurell. Improving contact prediction along three dimensions. *PLOS Comp Biol*, 10(10):e1003847, 2014.

[30] Christoph Feinauer, Hendrik Szurmant, Martin Weigt, and Andrea Pagnani. Inferring protein-protein interaction networks from inter-protein sequence co-evolution. *arXiv preprint arXiv:1512.05420*, 2015.

[31] Christoph Feinauer, Hendrik Szurmant, Martin Weigt, and Andrea Pagnani. Inter-protein sequence co-evolution predicts known physical interactions in bacterial ribosomes and the trp operon. *PLoS ONE*, 11(2):e0149166, 02 2016.

[32] Andrew L. Ferguson, Jaclyn K. Mann, Saleha Omarjee, Thumbi Ndung'u, Bruce D. Walker, and Arup K. Chakraborty. Translating HIV sequences into quantitative fitness landscapes predicts viral vulnerabilities for rational immunogen design. *Immunity*, 38(3):606–617, March 2013.

[33] M. Figliuzzi, H. Jacquier, A. Schug, O. Tenaillon, and M. Weigt. Coevolutionary landscape inference and the context-dependence of mutations in beta-lactamase TEM-1. *Molecular Biology and Evolution*, page msv211, October 2015.

[34] Robert D. Finn, Alex Bateman, Jody Clements, Penelope Coggill, Ruth Y. Eberhardt, Sean R. Eddy, Andreas Heger, Kirstie Hetherington, Liisa Holm, Jaina Mistry, Erik L. L. Sonnhammer, John Tate, and Marco Punta. Pfam: the protein families database. *Nucleic Acids Research*, 42(D1):D222–D230, 2014.

[35] Robert D. Finn, Jody Clements, and Sean R. Eddy. HMMER web server: interactive sequence similarity searching. *Nucleic Acids Research*, 39(Web Server issue):W29–W37, July 2011.

[36] Robert D. Finn, John Tate, Jaina Mistry, Penny C. Coggill, Stephen John Sammut, Hans-Rudolf Hotz, Goran Ceric, Kristoffer Forslund, Sean R. Eddy, Erik L. L. Sonnhammer, and Alex Bateman. The pfam protein families database. *Nucleic Acids Research*, 36(suppl 1):D281–D288, 2008.

[37] Michael Y Galperin and Eugene V Koonin. Who's your neighbor? new computational approaches for functional genomics. *Nature biotechnology*, 18(6):609–613, 2000.

[38] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 2003.

[39] A. Georges and J. S. Yedidia. How to expand around mean-field theory using high-temperature expansions. *Journal of Physics A: Mathematical and General*, 24(9):2173, 1991.

[40] Ulrike Gobel, Chris Sander, Reinhard Schneider, and Alfonso Valencia. Correlated mutations and residue contacts in proteins. *Proteins-Structure Function and Genetics*, 18(4):309–317, 1994.

[41] Eoghan D Harrington, Lars J Jensen, and Peer Bork. Predicting biological networks from genomic data. *FEBS letters*, 582(8):1251–1258, 2008.

[42] John Harris, Jeffry L. Hirst, and Michael Mossinghoff. *Combinatorics and Graph Theory*. Springer New York, September 2008.

[43] Yuen Ho et al. Systematic identification of protein complexes in saccharomyces cerevisiae by mass spectrometry. *Nature*, 415(6868):180–183, 2002.

[44] T.A. Hopf, L.J. Colwell, R. Sheridan, B. Rost, C. Sander, and D.S. Marks. Three-dimensional structures of membrane proteins from genomic sequencing. *Cell*, 2012.

[45] Thomas A Hopf, Charlotta P I Schärfe, João P G L M Rodrigues, Anna G Green, Oliver Kohlbacher, Chris Sander, Alexandre M J J Bonvin, and Debora S Marks. Sequence co-evolution gives 3d contacts and structures of protein complexes. *eLife*, 3, 2014.

[46] Robert J. Ingham, Karen Colwill, Caley Howard, Sabine Dettwiler, Caesar S. H. Lim, Joanna Yu, Kadija Hersi, Judith Raaijmakers, Gerald Gish, Geraldine Mbamalu, Lorne Taylor, Benny Yeung, Galina Vassilovski, Manish Amin, Fu Chen, Liudmila Matskova, Gösta Winberg, Ingemar Ernberg, Rune Linding, Paul O'Donnell, Andrei Starostine, Walter Keller, Pavel Metalnikov, Chris Stark, and Tony Pawson. WW Domains Provide a Platform for the Assembly of Multiprotein Networks. *Molecular and Cellular Biology*, 25(16):7092–7106, August 2005.

[47] Takashi Ito, Tomoko Chiba, Ritsuko Ozawa, Mikio Yoshida, Masahira Hattori, and Yoshiyuki Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Poc. Natl. Acad. Sci.*, 98(8):4569–4574, 2001.

[48] Edwin T. Jaynes. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957.

[49] Edwin T. Jaynes. *Probability theory: the logic of science*. Cambridge university press, 2003.

[50] David T. Jones, Daniel WA Buchan, Domenico Cozzetto, and Massimiliano Pontil. PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics*, 28(2):184–190, 2012.

[51] David T. Jones, Tanya Singh, Tomasz Kosciolek, and Stuart Tetchner. MetaPSICOV: combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins. *Bioinformatics (Oxford, England)*, 31(7):999–1006, April 2015.

[52] David Juan, Florencio Pazos, and Alfonso Valencia. High-confidence prediction of global interactomes based on genome-wide coevolutionary networks. *Poc. Natl. Acad. Sci.*, 105(3):934–939, 2008.

[53] Kazutaka Katoh, Kei-ichi Kuma, Hiroyuki Toh, and Takashi Miyata. MAFFT version 5: improvement in accuracy of multiple sequence alignment. *Nucleic acids research*, 33(2):511–518, 2005.

[54] Kazutaka Katoh, Kazuharu Misawa, Kei-ichi Kuma, and Takashi Miyata. Mafft: a novel method for rapid multiple sequence alignment based on fast fourier transform. *Nucleic acids research*, 30(14):3059–3066, 2002.

[55] Kazutaka Katoh and Daron M. Standley. Mafft multiple sequence alignment software version 7: Improvements in performance and usability. *Molecular Biology and Evolution*, 30(4):772–780, 2013.

[56] Maurice George Kendall and others. The advanced theory of statistics. *The advanced theory of statistics.*, (2nd Ed), 1946.

[57] George A. Khoury, James Smadbeck, Chris A. Kieslich, and Christodoulos A. Floudas. Protein folding and de novo protein design for biotechnological applications. *Trends in Biotechnology*, 32(2):99–109, January 2014.

[58] Thorsten Knöchel, Andreas Ivens, Gerko Hester, Ana Gonzalez, Ronald Bauerle, Matthias Wilmanns, Kasper Kirschner, and Johan N. Jansonius. The crystal structure of anthranilate synthase from sulfolobus solfataricus: Functional implications. *Proceedings of the National Academy of Sciences*, 96(17):9479–9484, 1999.

[59] Victor Kunin, Ildefonso Cases, Anton J Enright, Victor de Lorenzo, and Christos A Ouzounis. Myriads of protein families, and still counting. *Genome Biology*, 4(2):401, 2003.

[60] David J. C. MacKay. *Information Theory, Inference and Learning Algorithms.* Cambridge University Press, September 2003.

[61] Cynthia J Verjovsky Marcotte and Edward M Marcotte. Predicting functional linkages from gene fusions with confidence. *Applied bioinformatics*, 1(2):93–100, 2002.

[62] Debora S. Marks, Lucy J. Colwell, Robert Sheridan, Thomas A. Hopf, Andrea Pagnani, Riccardo Zecchina, and Chris Sander. Protein 3d Structure Computed from Evolutionary Sequence Variation. *PLoS ONE*, 6(12):e28766, December 2011.

[63] Debora S. Marks, Thomas A. Hopf, and Chris Sander. Protein structure prediction from sequence variation. *Nature Biotechnology*, 30(11):1072–1080, November 2012.

[64] MATLAB. *version R2014a.* The MathWorks Inc., Natick, Massachusetts, 2014.

[65] Marina Meil\ua and Tommi Jaakkola. Tractable Bayesian learning of tree belief networks. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 380–388, 2000.

[66] Marc Mezard and Andrea Montanari. *Information, physics, and computation.* Oxford University Press, 2009.

[67] Bohdan Monastyrskyy, Daniel D'Andrea, Krzysztof Fidelis, Anna Tramontano, and Andriy Kryshtafovych. New encouraging developments in contact prediction: Assessment of the CASP11 results. *Proteins: Structure, Function, and Bioinformatics*, 2015.

[68] Faruck Morcos, Andrea Pagnani, Bryan Lunt, Arianna Bertolino, Debora S. Marks, Chris Sander, Riccardo Zecchina, José N. Onuchic, Terence Hwa, and Martin Weigt. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, 108(49):E1293–E1301, 2011.

[69] David L. Nelson, Albert L. Lehninger, and Michael M. Cox. *Principles Of Biochemistry, 4th Edition.* Macmillan, 2008.

[70] Sergey Ovchinnikov, Hetunandan Kamisetty, and David Baker. Robust and accurate prediction of residue–residue interactions across protein interfaces using evolutionary information. *eLife*, 3, 2014.

[71] Sergey Ovchinnikov, David E. Kim, Ray Yu-Ruei Wang, Yuan Liu, Frank DiMaio, and David Baker. Improved de novo structure prediction in CASP11 by incorporating Co-evolution information into rosetta. *Proteins: Structure, Function, and Bioinformatics*, 2015.

[72] Florencio Pazos and Alfonso Valencia. In silico two-hybrid system for the selection of physically interacting protein pairs. *Proteins: Structure, Function, and Bioinformatics*, 47(2):219–227, 2002.

[73] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, 1988.

[74] Matteo Pellegrini, Edward M Marcotte, Michael J Thompson, David Eisenberg, and Todd O Yeates. Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Poc. Natl. Acad. Sci.*, 96(8):4285–4288, 1999.

[75] Anna I. Podgornaia and Michael T. Laub. Pervasive degeneracy and epistasis in a protein-protein interface. *Science*, 347(6222):673–677, February 2015.

[76] Andrea Procaccini, Bryan Lunt, Hendrik Szurmant, Terence Hwa, and Martin Weigt. Dissecting the specificity of protein-protein interaction in bacterial two-component signaling: orphans and crosstalks. *PloS one*, 6(5):e19729, 2011.

[77] M. Punta, P. C. Coggill, R. Y. Eberhardt, J. Mistry, J. G. Tate, C. Boursnell, N. Pang, K. Forslund, G. Ceric, J. Clements, A. Heger, L. Holm, E. L. L.

Sonnhammer, S. R. Eddy, Al. Bateman, and R. D. Finn. The Pfam protein families database. *Nucleic Acids Res.*, 40:D290, 2012.

[78] Oliver C. Redfern, Benoit Dessailly, and Christine A. Orengo. Exploring the structure and function paradigm. *Current Opinion in Structural Biology*, 18(3):394–402, June 2008.

[79] Michael Remmert, Andreas Biegert, Andreas Hauser, and Johannes Söding. HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nature Methods*, 9(2):173–175, February 2012.

[80] Boris Reva, Yevgeniy Antipin, and Chris Sander. Predicting the functional impact of protein mutations: application to cancer genomics. *Nucleic Acids Research*, 39(17):e118, September 2011.

[81] Yasser Roudi, Sheila Nirenberg, and Peter E. Latham. Pairwise Maximum Entropy Models for Studying Large Biological Systems: When They Can Work and When They Can't. *PLoS Comput Biol*, 5(5):e1000380, May 2009.

[82] William P. Russ, Drew M. Lowery, Prashant Mishra, Michael B. Yaffe, and Rama Ranganathan. Natural-like function in artificial WW domains. *Nature*, 437(7058):579–583, September 2005.

[83] Alexander Schug, Martin Weigt, José N. Onuchic, Terence Hwa, and Hendrik Szurmant. High-resolution protein complexes from integrating genomic information with molecular simulation. *Proceedings of the National Academy of Sciences*, 106(52):22124–22129, 2009.

[84] Marcin J. Skwark, Daniele Raimondi, Mirco Michel, and Arne Elofsson. Improved Contact Predictions Using the Recognition of Protein Like Contact Patterns. *PLoS Comput Biol*, 10(11):e1003889, November 2014.

[85] Michael Socolich, Steve W. Lockless, William P. Russ, Heather Lee, Kevin H. Gardner, and Rama Ranganathan. Evolutionary information for specifying a protein fold. *Nature*, 437(7058):512–518, September 2005.

[86] Ann M. Stock, Victoria L. Robinson, and and Paul N. Goudreau. Two-Component Signal Transduction. *Annual Review of Biochemistry*, 69(1):183–215, 2000.

[87] Ludovico Sutto, Simone Marsili, Alfonso Valencia, and Francesco Luigi Gervasio. From residue coevolution to protein conformational ensembles and functional dynamics. *Proceedings of the National Academy of Sciences*, 112(44):13567–13572, 2015.

[88] Graham Upton and Ian Cook. *A Dictionary of Statistics*. Oxford University Press, 2 edition, January 2008.

[89] William S. J. Valdar. Scoring residue conservation. *Proteins*, 48(2):227–241, August 2002.

[90] Alfonso Valencia and Florencio Pazos. Computational methods for the prediction of protein interactions. *Current Opinion in Structural Biology*, 12(3):368 – 373, 2002.

[91] Michele Vendruscolo, Edo Kussell, and Eytan Domany. Recovery of protein structure from contact maps. *Folding and Design*, 2(5):295–306, October 1997.

[92] Junmei Wang, Romain M. Wolf, James W. Caldwell, Peter A. Kollman, and David A. Case. Development and testing of a general amber force field. *Journal of computational chemistry*, 25(9):1157–1174, 2004.

[93] Martin Weigt, Robert A White, Hendrik Szurmant, James A Hoch, and Terence Hwa. Identification of direct residue contacts in protein–protein interaction by message passing. *Poc. Natl. Acad. Sci.*, 106(1):67–72, 2009.

[94] Daniel M. Weinreich, Nigel F. Delaney, Mark A. Depristo, and Daniel L. Hartl. Darwinian evolution can follow only very few mutational paths to fitter proteins. *Science (New York, N.Y.)*, 312(5770):111–114, April 2006.

[95] Michael Weyand, Ilme Schlichting, Anna Marabotti, and Andrea Mozzarelli. Crystal structures of a new class of allosteric effectors complexed to tryptophan synthase. *Journal of Biological Chemistry*, 277(12):10647–10652, 2002.

[96] Paul C. Whitford, Jeffrey K. Noel, Shachi Gosavi, Alexander Schug, Kevin Y. Sanbonmatsu, and José N. Onuchic. An all-atom structure-based potential for proteins: bridging minimal models with all-atom empirical forcefields. *Proteins*, 75(2):430–441, May 2009.

[97] Matthias Wilmanns, John P. Priestle, Thomas Niermann, and Johan N. Jansonius. Three-dimensional structure of the bifunctional enzyme phosphoribosylanthranilate isomerase: Indoleglycerolphosphate synthase from escherichia coli refined at 2.0 å resolution. *Journal of Molecular Biology*, 223(2):477 – 507, 1992.

[98] Alexander Wlodawer, Jochen Walter, Robert Huber, and Lennart Sjölin. Structure of bovine pancreatic trypsin inhibitor: Results of joint neutron and x-ray refinement of crystal form ii. *Journal of Molecular Biology*, 180(2):301–329, 1984.

[99] F. Y. Wu. The Potts model. *Reviews of Modern Physics*, 54(1):235–268, January 1982.

[100] Chen-Hsiang Yeang and David Haussler. Detecting coevolution in and among protein domains. *PLoS Comput Biol*, 3(11):e211, 11 2007.