Bayesian Model for Prediction of Protein Residue-Residue Contacts

Susann Vorberg

15.10.2017

Dissertation zur Erlangung des Doktorgrades der Fakultt fr Chemie und Pharmazie der Ludwig-Maximilians-Universitt Mnchen

Bayesian Model for Prediction of Protein Residue-Residue Contacts

vorgelegt von
Susann Vorberg
geboren in Leipzig, Germany

Mnchen, den 15.10.2017

Erklrung

Diese Dissertation wurde im Sinne von 7 der Promotionsordnung vom 28. November 2011 von Dr. Johannes Soeding betreut.

Eidesstattliche Versicherung

Diese Dissertation wurde eigenstndig und ohne unerlaubte Hilfe erarbeitet.

....

Ort, Datum Susann Vorberg

Dissertation eingereicht am: 15.10.2017

Erstgutachter: Dr. Johannes Soeding .....

Zweitgutachter: Prof. Dr. Julien Gagneur .....

Tag der mndlichen Prfung: 15.12.2017

Summary

Awesome contact prediction project abstract

Acknowledgements

I thank the world.

Table of Contents

1 Introduction

In his Nobel Prize speech in 1973 [1] Anfinsen postulated one of the basic principles in molecular biology, which is known as *Anfinsen's dogma*: a protein's native structure is uniquely determined by its amino acid sequence. With certain exceptions (e.g. IDP [2]), this dogma has proven to hold true for the majority of proteins.

Ever since, it is regarded as the biggest challenge in structural bioinformatics [3], to realiably predict a protein's structure given only its amino acid sequence. *De-novo* protein structure prediction methods use physical or knowledge based energy potentials to find a protein conformation that minimizes the protein's energy landscape. However, these methods are limited by the complexity of the conformational space and the accuracy of the energy potentials. Considering a protein with 150 amino acids, that has approximately 450 degrees of freedom, Regarding the rotational and translational degrees of freedom of the protein chain, the complexity scales with XXX [1].

Far more successfull are template-based modelling approaches. Given the observation that structure is more conserved than sequence in a protein family [4], the structure of a target protein can be inferred from a homologue protein [5]. The degree of structural conservation is linked to the level of pairwise sequence identity [6]. Therefore, the accuracy of a model crucially depends on the sequence identity between target and template and determines the applicability of the model [7]. By definition, homology derived models are unable to capture new folds [8] and their main limitation lies in the availability of suitable templates.

Unfortunately, the number of solved protein structures increases only slowly, as experimental methods are both time consuming and expensive [8]. The PDB[9] is the main repository for marcomolecular structures and currently (Jul 2017) holds about 120 000 atomic models of proteins. The primary technique for determining protein structures is X-ray crystallography, accounting for roughly 90% of entries in the PDB. About 9% of protein structures have been solved using NMR and less than 1% using EM (see FIG 1).

All three experimental techniques have advantages and limitations with respect

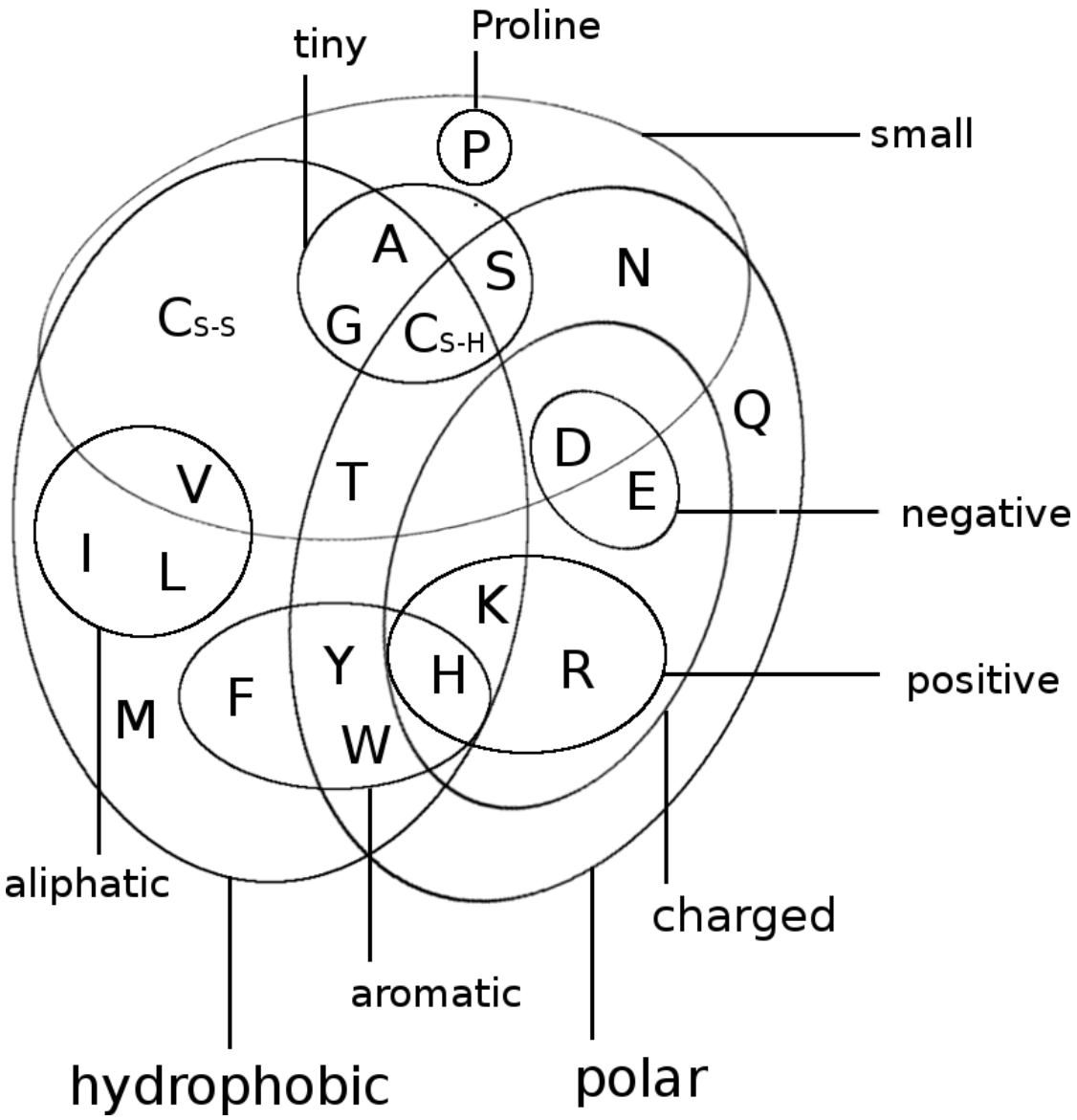


Figure 1.1: Yearly growth of number of solved structures in the PDB[@Berman2000] and protein sequences in the Uniprot[@TheUniProtConsortium2013].

to certain modelling aspects. X-ray crystallography requires the protein to form crystals, which is an arduous and sometimes impossible task. Furthermore, crystal packing forces the protein into a unnatural and rigid environment preventing the observation of conformational flexibility. NMR studies the protein in an physiological environment in solution and enables the study of protein dynamics as ensembles of protein structures can be observed. However, NMR is limited to look at small proteins. Recently, EM has undergone a "resolution revolution" [10] and macromolecular structures have been solved with resolutions up to 2A[citation]. The limit of cryo-EM lies in the size of proteins.

Compared to the tedious task of revealing atomic resolution of a protein tertiary structure, it has become very easy to decipher the primary sequence of proteins. With the latest sequencing technologies [examples], it takes only hours to sequence millions of basepaires at low costs [example numbers] and the number of sequenced genomes has risen tremendously. The UniProtKB [11], the leading resource for protein sequences, contains more than 80 million sequence entries (24 July 2017).

Consequently, the gap between the number of protein structures and sequences is still growing and even new developments as single protein structure determination [**???**] are not expected to close this gap near in time. [Figure sequence structure gap]

Protein structure determines protein function. Therefore, structural insights are of uttermost importance. They are essential for a detailed understanding of chemical reactions, regulatory processes and transport mechanisms. They are fundamental for the design of drugs and antibiotics. Moreover structural abnormalities can lead to misfolding and aggregation potentially causing diseases so studying them is pathologically relevant.

The aformentioned trends illustrate the need of computational methods and motivate research to solve *Ansinsens Dogma* to reliably predict protein structures from sequence alone.

1.1 Protein Structure

- Primary: Amino Acid Sewuence
- Secondary: Helices, sheets, coils, repeats,..
- tertiary: interaction of secondary structure elementws
- quartary: interaction of domains

1.1.1 Amino Acid Interactions

The Venn diagram in figure 1.2 displays a typical classification of amino acids with respect to their physico-chemical properties.

The aromatic amino acids tryptophan (W), tyrosine (Y), phenylalanine (F), and histidine (H) contain an aromatic ring system. Generally, aromatic ring systems are planar, and electons are shared over the whole ring structure. Interactions between aromatic residues have very constrained geometries regarding the angle

between the centroid of their rings. The π-electron systems favour T-shaped or offset stacked conformations [12]. Preferred distances between aromatic residues have been observed between $4.5\mathring{A}$ and $7\mathring{A}$ of their ring centroids [13].

Cysteine (C) residues can form disulphide bonds, which are the only covalent bonds between two amino acid side chains. They comprise the strongest side chain interactions in protein structures and their length varies between $3.5\mathring{A}$ to $4\mathring{A}$. Disulphide bonds also have a well defined geometry: there are five dihedral angles in a disulphide bond resulting in 20 different possible configurations. Only one configuration is favoured so that the dihedral angle between the carbon and sulfur atoms is close to 90 degrees [14]. They play a very important role in stabilizing protein structures. The number of disulfide bonds is negatively correlated with protein length: smaller proteins have more disulfide bonds helping to stabilize the structure in absence of strong hydrophobic packing in the core. It has also been found that disulfide bonds are more frequently observed in proteins of hyperthermophilic bacteria, being positively selected for increased stability [15].

Salt bridges are based on electrostatic interactions between positively charged residues (arginine (R) and lysine (K)) and negatively charged residues (aspartic acid (D) and glutamic acid (E)). The strength of electrostatic interactions, as described by Coulomb's law, decreases with distance between the point charges at the functional groups. It has been found to be maximal at $4\mathring{A}$ with respect to the functional groups of the both residues [16].

Hydrogen bonds can be formed between a donor residue which possesses an hydrogen atom attached to a strongly electronegative atom and an acceptor residue which possesses an electronegative atom with a lone electron pair. They are electrostatic interactions as well and thus their strength depends on distance as well. Hydrogen bonds are formed at distances of $2.4\mathring{A}$ to $3.5\mathring{A}$ between the non-hydrogen atoms (Berg JM, Tymoczko JL, 2002).

Salt bridges as well as hydrogen bonds have strong geometric preferences (Kumar and Nussinov, 1999). The geometry of a hydrogen bond depends on the angle between the HB donor, the hydrogen atom and the HB acceptor (Torshin et al., 2002).

Cation–π interactions are formed between positively charged or partially charged amino acids with amino groups (K,R,Q,E) and aromatic residues (W,Y,F,H). The preferential distance of the amino group to the π-electron system has been determined between $3.4\mathring{A}$ and $6\mathring{A}$ [17] [18] Their role in stabilizing protein structures is still under debate [19].

Proline residues are conformationally restricted, with the alpha-amino group of the backbone directly attached to the side chain. The sterical rigidity of the proline side chain restricts the backbone angle and thus affects secondary structure formation. Proline is known as a helix-breaker. Whereas other aromatic side chains are defined by their negatively charged π faces, the face of proline side chains is partially positively charged. Thus, aromatic and proline residues can interact favorably with each other. Once due to the hydrophobic nature of the residues and also due to the interaction between the negatively charged aroamtic π face and the polarized C-H bonds in proline, called a CH/π interaction.

Petersen et al. (2012) found clear secondary structure elements preferences for

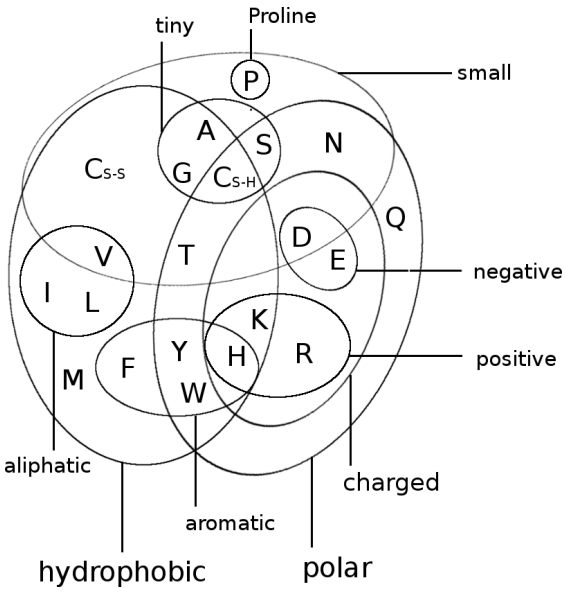


Figure 1.2: Physico-chemical properties of amino acids. The 20 naturally occuring amino acids are grouped with respect to ten physico-chemical properties. Adapted from Figure 1a in [@Livingstone1993].

each amino acid pair. For example, residue pairs containing Alanine and Leucine are predominantly found in buried α-helices, whereas pairs containing Isoleucine and Valine preferentially are located in β-sheet environments. Of course, solvent accessibility represents an important criterion for residue interactions. Hydrophobic residues are rather buried in the structure, whereas polar and charged residues are found more frequently on the protein surface and interact with water molecules.

1.2 Structure Prediction

Despite the knowledge of Anfinsen's postulate, we are not able to reliably predict the structure of a protein from its sequence alone. Generally it is assumed that a protein folds into a unique, well-defined native structure that is near the global free energy minimum (fig:folding_funnel). Levinthal's paradox [20] describes the complexity of the folding process towards this minimum. It stresses the problem that it is not possible for a protein to exhaustively search the conformational space to get to its native fold. Due to the "combinatorial explosion" of possible conformations, an exhaustive search would take unreasonably long. Hence, it is not a feasible approach for structure prediction to scan all possible conformations. Different approaches have been developed over time to overcome or elude this problem.

1.2.1 Template-based methods

Homology modeling is by far the most successful approach to structure prediction. The basic concept of this strategy relates to the fact that structure is more conserved than sequence [4]. After detecting a homologous protein of known structure,

that has sufficient sequence similarity, it can be used as a template to model the structure of the target protein.

The degree of structural conservation is linked to the level of pariwise sequence identity [6]. Homology Modelling is assumed to yield reliably accurate models when query and target protein share more than 30% sequence similarity, depending on the sequence length (*safe homology zone*) [5]. Below a threshold of ˜20-35% pairwise sequence identity (*twighlight-zone*) the number of false positives regarding structural similarity explodes and structural inference becomes less reliable and more than 95% of structures are dissmilar [21]. Advances in remote homology detection and alignment generation have improved the quality of models, even beyond the once postulated limit of the *twighlight-zone* [22]. Integration of multiple templates has also proved to increase model quality [23]

After the identification of a suitable template, there are different strategies that can be followed to obtain a model for the target protein. The the backbone of the model is generated by simply copying the coordinates of the target backbone atoms onto the model. Non-aligned residues due to gaps in the alignment have to be modelled *de-novo*, meaning from scratch. This can be done by a knowledge-based search for suitable fragments in the PDB or by true energy-based *de-novo* modelling. When the backbone is generated, the side chains are modelled, usually by searching rotamer libraries for energetically favoured residue conformations. Finally, the model is energetically optimized in an iterative procedure. Force fields are applied to correct the backbone and side chain conformations [24]. Several automated pipelines for homology modelling are well-established (Modeller [25], 3D-Jigsaw [26], SwissModel [27]) which allow more or less manual intervention in the modelling process.

Fold Recognition describes the inverse folding problem [Bowie1993]: instead of finding the compatible structure for a given sequence, one tries to find sequences that fit onto a given structure. Whether the query sequence fits a structure from the database is not determined by sequence similarities but rather energetic or environment specific measures. Thus, fold recognition methods are able to recognize structural similarity even in the absence of sequence similarity. The rationale basis for this strategy is the assumption that the fold space is limited. It has been found that seemingly unrelated proteins often adopt similar folds. This might be due to divergent evolution (proteins are related, but homology cannot be detected at the corresponding sequence level) or convergent evolution (functional requirements lead to similar folds for unrelated proteins) [Gu2009]. Early approaches include profile based methods. Here, the structural information of the protein is encoded into profiles, which subsequently are aligned to the sequences [Bowie1991,Fischer1996,Ouzounis1993]. Advanced techniques are known as "threading" techniques, describing the process of threading a sequence through a structure and determining the optimal fit via energy functions. [Jones1992,Jones1998,Lemer1995]

1.2.2 Template-free structure prediction

Ab initio or de-novo modeling techniques implement Anfinsen's Dogma most closely in mimicking the folding process based only on physico-chemical princi-

ples. Energy functions (physical or knowledge-based) are used to describe the folding landscape and are minimized to arrive at the global energy minimum corresponding to the native conformation. Since the native conformation can be found near the global energy minimum of the folding landscape, energy functions (physical or knowledge-based) have been developed to describe this landscape. With respect to the idea of a folding funnel, the energy function is minimized to mimic the folding process that automatically leads to the global minimum. Again, there exist numerous webservers that combine energy minimization, threading techniques and fragment-based approaches, e.g. Rosetta [\citep{]Simons1999], Tasser [Zhang2004, Touchstone II Zhang2003].

Drawbacks of these methods are the time requirements due to the computational complexity of energy functions as well as their inaccuracy.

Minimize a physical or knowledge-based energy function for the protein. This has huge complexity due to large conformational space that needs to be sampled.

1.2.3 contact assisted de-novo predictions

Structure Reconstruction from true contacts maps works well. Even a small number of contacts is sufficient to reconstruct the fold of the protein. Distance maps work even better.

What is the optimal distance cutoff to define a contact? Duarte et al 2010: between 8 and 12A Dyrka et al 2016 Konopka et al 2014 Sathyapriya et al 2009

Many studies that successfuly predict structures denovo with the help of predicted contact.

Vice versa, because contacts at large primary distances are rare, they are most informative for protein structure prediction: Izarzugaza J, Gran a O, Tress M, Valencia A, Clarke N (2007) Assessment of intramolecular contact predictions for CASP7

1.3 Contact Prediction

Contact prediction refers to the prediction of physical contacts between amino acid side chains in the 3D protein structure, given the protein sequence as input.

Historically, contact prediction was motivated by the idea that compensatory mutations between spatially neighboring residues can be traced down from evolutionary records [28]. As proteins evolve, they are under selective pressure to maintain their function and correspondingly their structure. Consequently, residues and interactions between residues constraining the fold, protein complex formation, or other aspects of function are under selective pressure. Highly constrained residues and interactions will be strongly conserved [29]. Another possibility to maintain structural integrity is the mutual compensation of unbeneficial mutations. For example, the unfavourable mutation of a small amino acid residue into a bulky residue in the densely packed protein core might have been compensated in the course of evolution by a particularly small side chain in a neighboring position.

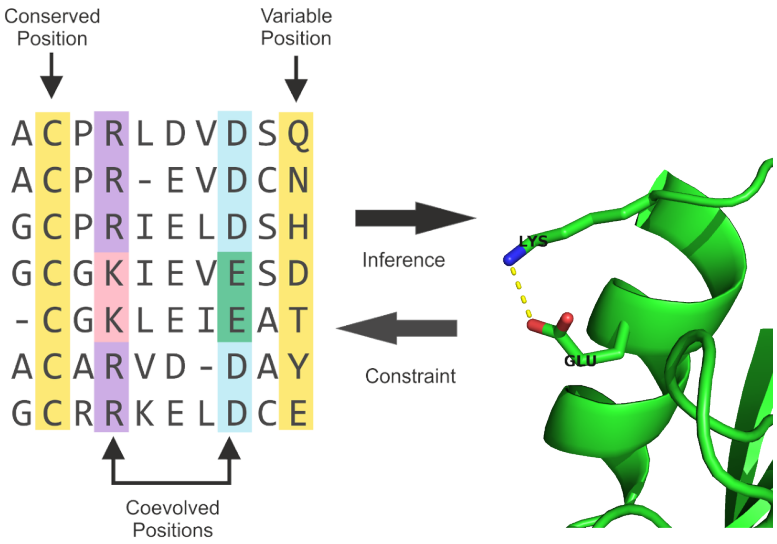


Figure 1.3: The evolutionary record of a protein family reveals evidence of compensatory mutations between spatially neighboring residues that are under selective pressure with respect to some physico-chemical constraints. Mining protein family sequence alignments for residue pairs with strong coevolutionary signals using statistical methods allows inference of spatial proximity for these residue pairs.

Other physico-chemical quantities such as amino acid charge or hydrogen bonding capacity can also induce compensatory effects[30]. The MSA of a protein family comprises homolog sequences that have descended from a common ancestor and are aligned relative to each other. According to the hypothesis, compensatory mutations show up as correlations between the amino acid types of pairs of MSA columns and can be used to infer spatial proximity of residue pairs (see Figure 1.3).

The following sections will give an overview over important methods and developments in the field of contact prediction.

1.3.1 Local Statistical Models

Early contact prediction methods used local pairwise statistics to infer contacts that regard pairs of amino acids in a sequence as statistically independent from another. A significant drawback of these methods is their inability to account for transitive effects arising from chains of correlations between multiple residue pairs as described in the section on Transitive Effects.

Several of these methods use correlation coefficient based measures, such as Pearson correlation between amino acid counts, properties associated with amino acids or mutational propensities at the sites of a MSA [28,30–33].

Many methods have been developed that are rooted in information theory and use MI measures to describe the dependencies between sites in the alignment [34–36]. Phylogenetic and entropic biases have been identified as the major sources of noise that confound the true coevolution signal [36–38]. Different variants of MI based approaches address these effects and improve on the signal-to-noise ratio

[37,39,40]. The most prominent correction for background noises is APC that drastically removes background noise attributed to entropic effects and is discussed in section 1.4.4 [41].

Another popular method is *OMES* that essentially computes a chi-squared statistic to detect the differences between observed and expected pairwise amino acid frequencies for a pair of columns [42,43].

Eventhough these methods cannot compete with modern predictors, *OMES* and MI based scores often serve as a baseline in performance benchmarks for contact prediction [44,45].

1.3.2 Global Statistical Models

A huge leap forward was the development of sophisticated statistical models that make predictions for a single residue pair while considering all other pairs in the protein. These global models allow for the distinction between transitive and causal interactions which has been referred to in the literature as DCA [46,47].

In 1999 Lapedes et al. were the first to propose a global statistical approach for the prediction of residue-residue contacts in order to disentangle transitive effects [46]. They consider a Pott's model that can be derived under a maximum entropy assumption and use the model specific coupling parameters to infer interactions. At that time the wider implications of this advancement went unnoted, but meanwhile the Pott's Model has become the most prominent statistical model for contact prediction. Section 1.4 deals extensively with the derivation and properties of the Pott's model, its application to contact prediction and its numerous realizations.

A global statistical model not motivated by the maximum entropy approach was proposed by Burger and Nijmwegen in 2010 [48,49]. Their fast Bayesian network model incorporates additional prior information and phylogenetic correction via APC but cannot compete with the pseudo-likelihood approaches presented in section 1.4.3.1.

1.3.3 Machine Learning Methods and Meta-Predictors

With the steady increase in protein sequence data, machine learning based methods have emerged that extract features from MSAs in order to learn associations between input features and residue-residue contacts. Sequence features typically include predicted solvent accessibility, predicted secondary structure, contact potentials, conservation scores, global protein features, pairwise coevolution statistics and averages of certain features over sequence windows.

Numerous sequence-based methods have been developed using machine learning algorithms, such as support support vector machines (*SVMCon* [50], *SVM-SEQ* [51]), random forests (*ProC_S3* [52], *TMhhcp* [53], *PhyCMap* [54]), neural networks (*NETCSS* [55], *SAM* [56], [57], *SPINE-2D* [58], *NNCon* [59]) deep neural networks (*DNCon* [60], *CMAPpro* [61]) and ensembles of genetic algorithm classfiers (*GaC* [62]). Their performance is less dependent on the number of available sequence

homologs compared to coevolution methods and therefore they can outperform pure coevolution methods in low data ranges [54,63].

Sequence-based contact predictors and coevolution methods provide orthogonal information on the likelihood that a pair of residues makes a contact. The next logical step in method development therefore constitutes the combination of several base predictors and classical sequence-derived features in the form of meta-predictors.

The first published meta-predictor was *PconsC* in 2013, combining *PSICOV* and *plmDCA* and in a later version *PhyCMap*, *gaussianDCA* and *plmDCA* with sequence features [64,65]. *EPC-MAP* uses *GREMLIN* as coevolution feature and physicochemical information from predicted ab initio protein structures [66]. In 2015, *MetaPSICOV* was released combining predictions from *PSICOV*, *mfDCA* and *CCMpred* with other sequence derived feautures [67]. *RaptorX* uses *CCMpred* predictions as coevolution feature and other standard contact prediction features within an ultra-deep neural network [68]. The newest developments *EPSILON-CP* and *NeBcon* both comprise the most comprehensive usage of contact prediction methods so far, combining five and eight state-of-the-art contact predictors, respectively [69,70].

Eventhough a benchmark comparing the recently developed meta-predictors is yet to be made, it becomes clear from the recent CASP experiments, that meta-predictors outperform pure coevolution methods [71]. As coevolution scores comprise the most informative feautures among the set of input features, it is clear that meta-predictors will benefit from further improvements of pure coevolution methods [68,69].

1.4 Modelling Protein Families with Potts Model

Infering contacts from a joint probability distribution over all residues in a protein sequence instead of using simple pairwise statistics has been proven to enable the distinction of direct statistical dependencies between residues from indirect dependencies mediated through other residues. The global statistical model that is commonly used to describe this joint probability distribution is the *Potts model*. It is a well-established model in statistical mechanics and can be derived from a maximum entropy assumption which is explained in the following.

The principle of maximum entropy, proposed by Jaynes in 1957 [72,73], states that the probability distribution which makes minimal assumptions and best represents observed data is the one that is in agreement with measured constraints (prior information) and has the largest entropy. In other words, from all distributions that are consistent with measured data, the distribution with maximal entropy should be chosen.

A protein family is represented by a MSA $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ of N protein sequences. Every protein sequence of the protein family represents a sample drawn from a target distribution $p(\mathbf{x})$, so that each protein sequence is associated with a probability. Every sequence $\mathbf{x} = (x_1, \ldots, x_L)$ is of length L and every position

in the sequences constitutes a categorical variables x_i that can take one of $q = 21$ values representing the 20 naturally occuring amino acids and a gap ('-'). The measured constraints are given by the empirically observed single and pairwise amino acid frequencies that can be calculated as

$$f_i(a) = f(x_i{=}a) = \frac{1}{N}\sum_{n=1}^{N} I(x_{ni}{=}a) \ , \tag{1.1}$$

$$f_{ij}(a,b) = f(x_i{=}a, x_j{=}b) = \frac{1}{N}\sum_{n=1}^{N} I(x_{ni}{=}a, x_{nj}{=}b) \ . \tag{1.2}$$

According to the maximum entropy principle, the distribution $p(\mathbf{x})$ should have maximal entropy and reproduce the empirically observed amino acid frequencies, so that

$$\begin{aligned} f(x_i{=}a) &\equiv p(x_i{=}a) \\ &= \sum_{\mathbf{x\prime}_1,\ldots,\mathbf{x\prime}_L=1}^{q} p(x\prime) I(x\prime_i{=}a) \end{aligned} \tag{1.3}$$

$$\begin{aligned} f(x_i{=}a, x_j{=}b) &\equiv p(x_i{=}a, x_j{=}b) \\ &= \sum_{\mathbf{x\prime}_1,\ldots,\mathbf{x\prime}_L=1}^{q} p(x\prime) I(x\prime_i{=}a, x\prime_j{=}b) \ . \end{aligned} \tag{1.4}$$

Solving for the distribution $p(\mathbf{x})$ that maximizes the Shannon entropy $S = -\sum_{\mathbf{x\prime}} p(\mathbf{x\prime}) \log p(\mathbf{x\prime})$ while satisfying the constraints given by the empircial amino acid frequencies in eq. (1.4) by introducing Lagrange multipliers $\mathbf{w}_{ij}$ and v_i, results in the formulation of the *Potts model*,

$$p(\mathbf{x}|\mathbf{v},\mathbf{w}) = \frac{1}{Z(\mathbf{v},\mathbf{w})} \exp\left(\sum_{i=1}^{L} v_i(x_i) \sum_{1\le i<j\le L}^{L} w_{ij}(x_i,x_j)\right) \ . \tag{1.5}$$

The Lagrange multipliers $\mathbf{w}_{ij}$ and v_i remain as model parameters to be fitted to data. Z is a normalization constant also known as *partition function* that ensures the total probabilty adds up to one by summing over all possible assignments to $\mathbf{x}$,

$$Z(\mathbf{v},\mathbf{w}) = \sum_{\mathbf{x\prime}_1,\ldots,\mathbf{x\prime}_L=1}^{q} \exp\left(\sum_{i=1}^{L} v_i(x_i) \sum_{1\le i<j\le L}^{L} w_{ij}(x_i,x_j)\right) \ . \tag{1.6}$$

1.4.1 Model Properties

The Potts model is specified by singlet terms v_{ia} which describe the tendency for each amino acid a to appear at position i, and pair terms w_{ijab}, also called

couplings, which describe the tendency of amino acid a at position i to co-occur with amino acid b at position j. In contrast to mere correlations, the couplings explain the causative dependence structure between positions by jointly modelling the distribution of all positions in a protein sequence and thus account for transitive effects. By doing so, a major source of noise in contact prediction methods is eliminated.

To get some intuition for the coupling coefficients, note that $w_{ijab} = 1$ corresponds to a 2.7-fold higher probability for a and b to occur together than what is expected from the singlet frequencies if a and b were independent. Pairs of residues that are not in contact tend to have negligable couplings, $\mathbf{w}_{ij} \approx 0$, whereas pairs in contact tend to have vectors significantly different from 0. For contacting residues i and j in real world MSAs typical coupling strengths are on the order of $||\mathbf{w}_{ij}|| \approx 0.1$ (regularization dependent).

Maximum entropy models naturally give rise to exponential family distributions that express useful properties for statistical modelling, such as the convexity of the likelihood function which consequently has a unique, global minimum [74,75].

The Potts model is a discrete instance of what is referred to as a pairwise Markov random field in the statistics community. MRFs belong to the class of undirected graphical models, that represent the probability distribution in terms of a graph with nodes and edges characterizing the variables and the dependence structure between variables, respectively.

1.4.1.1 Gauge Invariance

As every variable x_{ni} can take $q = 21$ values, the model has $L \times q + L(L-1)/2 \times q^2$ parameters. But the parameters are not uniquely determined and multiple parametrizations yield identical probability distributions.

For example, adding a constant to all elements in v_i for any fixed position i or similarly adding a constant to v_{ia} for any fixed position i and amino acid a and subtracting the same constant from the qL coefficients w_{ijab} with $b \in \{1, \ldots, q\}$ and $j \in \{1, \ldots, L\}$ leaves the probabilities for all sequences under the model unchanged, since such a change will be compensated by a change of $Z(\mathbf{v}, \mathbf{w})$ in eq. (1.6).

The overparametrization is referred to as *gauge invariance* in statistical physics literature and can be eliminated by removing parameters [47,76]. An appropriate choice of which parameters to remove, referred to as *gauge choice*, reduces the number of parameters to $L \times (q-1) + L(L-1)/2 \times (q-1)^2$. Popular gauge choices are the *zero-sum gauge* or *Ising-gauge* used by Weigt et al. [47] imposed by the restraints,

$$\sum_{a=1}^{q} v_{ia} = \sum_{a=1}^{q} w_{ijab} = \sum_{a=1}^{q} w_{ijba} = 0 \tag{1.7}$$

for all i, j, b or the *lattice-gas gauge* used by Morcos et al [76] and Marks et al [77] imposed by restraints

$$\mathbf{w}_{ij}(q,a) = \mathbf{w}_{ij}(a,q) = v_i(q) = 0 \tag{1.8}$$

for all i, j, a [78].

Alternatively, the indeterminacy can be fixed by including a regularization prior (see next section). The regularizer selects for a unique solution among all parametrizations of the optimal distribution and therefore eliminates the need to choose a gauge [79–81].

1.4.2 Inferring Parameters for the Potts Model

Typically, parameter estimates are obtained by maximizing the log-likelihood function of the parameters over observed data. For the Potts model, the log-likelihood function is computed over sequences in the alignment $\mathbf{X}$:

$$\begin{aligned} \mathrm{LL}(\mathbf{v}, \mathbf{w}|\mathbf{X}) &= \sum_{n=1}^{N} \log p(\mathbf{x}_n) \\ &= \sum_{n=1}^{N} \left[\sum_{i=1}^{L} v_i(x_{ni}) + \sum_{1 \le i < j \le L} w_{ij}(x_{xn}, x_{nj}) - \log Z \right] \end{aligned} \tag{1.9}$$

The number of parameters in a Potts model is typically larger than the number of observations, i.e. the number of sequences in the MSA. Considering a protein of length $L = 100$, there are approximately 2×10^6 parameters in the model whereas the largest protein families comprise only around 10^5 sequences (see Figure 1.8). An underdetermined problem like this renders the use of regularizers neccessary in order to prevent overfitting.

Typically, an L2-regularization is used that pushes the single and pairwise terms smoothly towards zero and is equivalent to the logarithm of a zero-centered Gaussian prior,

$$\begin{aligned} R(\mathbf{v}, \mathbf{w}) &= \log \left[\mathcal{N}(\mathbf{v}|\mathbf{0}, \lambda_v^{-1} I) \mathcal{N}(\mathbf{w}|\mathbf{0}, \lambda_w^{-1} I) \right] \\ &= -\frac{\lambda_v}{2} ||\mathbf{v}||_2^2 - \frac{\lambda_w}{2} ||\mathbf{w}||_2^2 + \text{const.} , \end{aligned} \tag{1.10}$$

where the strength of regularization is tuned via the regularization coefficients λ_v and λ_w [82–84].

However, optimizing the log-likelihood requires computing the partition function Z given in eq. (1.6) that sums q^L terms. Computing this sum is intractable for realistic protein domains with more than 100 residues. Consequently, evaluating the likelihood function at each iteration of an optimization procedure is infeasible due to the exponential complexity of the partition function in protein length L.

Many approximate inference techniques have been developed to sidestep the infeasible computation of the partition function for the specific problem of predicting contacts that are briefly explained in the next section.

1.4.3 Solving the Inverse Potts Problem

In 1999 Lapedes et al. were the first to propose maximum entropy models for the prediction of residue-residue contacts in order to disentangle transitive effects [46]. They used an iterative Monte Carlo procedure to obtain estimates of the partition function. As the calculations involved were very time-consuming and at that time required supercomputing resources, the wider implications were not noted yet.

Ten years later Weight et al proposed an iterative message-passing algorithm, here referred to as *mpDCA*, to approximate the partition function [47]. Eventhough their approach is computationally very expensive and in practice only applicable to small proteins, they obtained remarkable results for the two-component signaling system in bacteria.

Balakrishnan et al were the first to apply pseudo-likelihood approximations to the full likelihood in 2011 [85]. The pseudo-likelihood optimizes a different objective and replaces the global partition function Z with local estimates. Balakrishnan and colleagues applied their method *GREMLIN* to learn sparse graphical models for 71 protein families. In a follow-up study in 2013, the authors proposed an improved version of *GREMLIN* that uses additional prior information [84].

Also in 2011, Morcos et al. introduced a naive mean-field inversion approximation to the partition function, named *mfDCA* [76]. This method allows for drastically shorter running times as the mean-field approach boils down to inverting the empirical covariance matrix calculated from observed amino acid frequencies for each residue pair i and j of the alignment. This study performed the first high-throughput analysis of intradomain contacts for 131 protein families and facilitated the prediction of protein structures from accurately predicted contacts in [77].

The initial work by Balakrishnan and collegueas went almost unnoted as it was not primarily targeted to the problem of contact prediction. Ekeberg and collegueas independently developed the pseudo-likelihood method *plmDCA* in 2013 and showed its superior precision over *mfDCA* [80].

A related approach to mean-field approximation is sparse inverse covariance estimation, named *PSICOV*, developed by Jones et al. (2012) [45]. PSICOV uses an L1-regularization, known as graphical Lasso, to invert the correlation matrix and learn a sparse graphical model [86]. Both procedures, *mfDCA* and *PSICOV*, assume the model distribution to be a multivariate Gaussian. It has been shown by Banerjee et al. (2008)that this dual optimization solution also applies to binary data, as is the case in this application, where each position is encoded as a 20-dimensional binary vector [87].

Another related approach to *mfDCA* and *PSICOV* is *gaussianDCA*, proposed in 2014 by Baldassi et al. [88]. Similar to the other both approaches, they model the data as multivariate Gaussian but within a simple Bayesian formalism by using a suitable prior and estimating parameters over the posterior distribution.

So far, pseudo-likelihood has proven to be the most successful approximation of the likelihood with respect to contact prediction performance. Currently, there exist several implementations of pseudo-likelihood maximization that vary in slight

details, perform similarly and thus are equally popular in the community, such as CCMpred [82], plmDCA[83] and GREMLIN [84].

1.4.3.1 Maximum Likelihood Inference for Pseudo-Likelihood

The pseudo-likelihood is a rather old estimation principle that was suggested by Besag already in 1975 [89]. It represents a different objective function than the full likelihood and approximates the joint probability with the product over conditionals for each variable, i.e. the conditional probability of observing one variable given all the others:

$$\begin{aligned} p(\mathbf{x}|\mathbf{v},\mathbf{w}) &\approx \prod_{i=1}^{L} p(x_i|\mathbf{x}_{\setminus xi},\mathbf{v},\mathbf{w}) \\ &= \prod_{i=1}^{L} \frac{1}{Z_i} \exp\left(v_i(x_i) \sum_{1\leq i<j\leq L} w_{ij}(x_i,x_j)\right) \end{aligned} \tag{1.11}$$

Here, the normalization term Z_i sums only over all assignments to one position i in sequence:

$$Z_i = \sum_{a=1}^{q} \exp\left(v_i(a) \sum_{1\leq i<j\leq L} w_{ij}(a,x_j)\right) \tag{1.12}$$

Replacing the global partition function in the full likelihood with local estimates of lower complexity in the pseudo-likelihood objective resolves the computational intractability of the parameter optimization procedure. Hence, it is feasible to maximize the pseudo-log-likelihood function,

$$\begin{aligned} \mathrm{pLL}(\mathbf{v},\mathbf{w}|\mathbf{X}) &= \sum_{n=1}^{N}\sum_{i=1}^{L} \log p(x_i|\mathbf{x}_{\setminus xi},\mathbf{v},\mathbf{w}) \\ &= \sum_{n=1}^{N}\sum_{i=1}^{L} \left[v_i(x_{ni}) + \sum_{j=i+1}^{L} w_{ij}(x_{ni},x_{nj}) - \log Z_{ni}\right] , \end{aligned} \tag{1.13}$$

plus an additional regularization term in order to prevent overfitting and to fix the gauge to arrive at a MAP estimate of the parameters,

$$\hat{\mathbf{v}},\hat{\mathbf{w}} = \underset{\mathbf{v},\mathbf{w}}{\operatorname{argmax}}\ \mathrm{pLL}(\mathbf{v},\mathbf{w}|\mathbf{X}) + R(\mathbf{v},\mathbf{w}) \ . \tag{1.14}$$

Eventhough the pseudo-likelihood optimizes a different objective than the full-likelihood, it has been found to work well in practice for many problems, including contact prediction [75,79–81]. The pseudo-likelihood function retains the concavity of the likelihood and it has been proven to be a consistent estimator in the limit of infinite data for models of the exponential family [79,89,90]. That is, as the number of sequences in the alignment increases, pseudo-likelihood estimates converge towards the true full likelihood parameters.

1.4.4 Computing Contact Maps

Model inference as described in the last section yields MAP estimates of the couplings $\hat{\mathbf{w}}_{ij}$. In order to obtain a scalar measure for the coupling strength between two residues i and j, all available methods presented in section 1.4.3 heuristically map the $q \times q$ dimensional coupling matrix $\mathbf{w}_{ij}$ to a single scalar quantity.

mpDCA [47] and *mfDCA* [76,77] employ a score called DI, that essentially computes the MI for two positions i and j using the couplings $\mathbf{w}_{ij}$ instead of pairwise amino acid frequencies. All pseudo-likelihood methods (*plmDCA* [80,83], *CCMpred* [82], *GREMLIN* [84]) compute the *Frobenius norm* of the coupling matrix $\mathbf{w}_{ij}$ to obtain a scalar contact score C_{ij},

$$C_{ij} = ||\mathbf{w}_{ij}||_2 = \sqrt{\sum_{a,b=1}^{q} w_{ijab}^2} \; . \tag{1.15}$$

The Frobenius norm improves prediction performance over DI and further improvements can be obtained by computing the Frobenius norm only on the 20×20 submatrix thus ignoring contributions from gaps [80,88,91]. *PSICOV* [45] uses an L1-norm on the 20×20 submatrix instead of the Frobenius norm.

Furthermore it should be noted that the Frobenius norm is gauge dependent and is minimized by the *zero-sum gauge* [47]. Therefore, in [80,82,83,88] the coupling matrices are transformed to *zero-sum gauge* before computing the Frobenius norm:

$$\mathbf{w}\prime_{ij} = \mathbf{w}_{ij} - \mathbf{w}_{ij}(\cdot, b) - \mathbf{w}_{ij}(a, \cdot) + \mathbf{w}_{ij}(\cdot, \cdot) \; , \tag{1.16}$$

where $\cdot$ denotes average over the respective indices.

Another commonly applied heuristic known as APC has been found to substantially boost contact prediction performance [41,84]. Dunn et al. introduced APC in order to remove the influence of background noise arising from correlations between positions with high entropy or phylogenetic couplings [41]. APC was first adopted by *PSICOV* [45] but is now used by most methods to adjust scores. It substracts a term that is computed as the product over average row and column contact scores $\overline{C_i}$ divided by the average contact score over all pairs $\overline{C_{ij}}$,

$$C_{ij}^{APC} = C_{ij} - \frac{\overline{C_i}\,\overline{C_j}}{\overline{C_{ij}}} \; . \tag{1.17}$$

It was long under debate why APC works so well and how it can be interpreted. Zhang et al. showed that APC essentially approximates the first principal component of the contact matrix and therefore removes the highest variability in the matrix that is assumed to arise from background biases [92]. Furthermore, they studied an advanced decomposition technique, called LRS matrix decomposition, that decomposes the contact matrix into a low-rank and a sparse component, representing background noise and true correlations, respectively.
Inferring contacts from the sparse component works astonishing well, improving precision further over APC independent of the underlying statistical model.

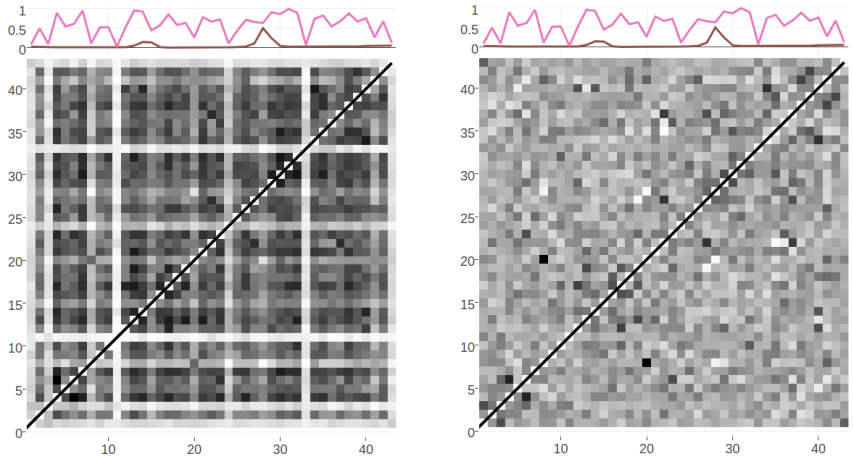


Figure 1.4: Contact maps computed from pseudo-likelihood couplings. Subplot on top of the contact maps illustrates the normalized Shannon entropy (pink line) and percentage of gaps for every position in the alignment (brown line). **Left**: Contact map computed with Frobenius norm as in eq. (1.15). Overall coupling values are dominated by entropic effects, i.e. the amount of variation for a MSA position, leading to striped brightness patterns. For example, positions with high column entropy (e.g. positions 7, 12 or 31) have higher overall coupling values than positions with low column entropy (e.g. positions 11, 24 or 33). **b**: previous contact map but corrected for background noise with the APC as in eq. (1.17).

Dr Stefan Seemayer could show that the main component of background noise can be attributed to entropic effects and that a substantial part of APC amounts to correcting for these entropic biases (unpublished). In his doctoral thesis, he developed a proper entropy correction, computed as the geometric mean of per-column entropies, that correlates well with the APC correction term and yields similar precision for predicted contacts. The entropy correction has the advantage that it is computed from input statistics and therefore is independent of the statistical model used to infer the couplings. In contrast, APC and other denoising techniques such as LRS [92] discussed above, estimate a background model from the final contact matrix, thus depending on the statistical model used to infer the contact matrix.

The general "smoothing" effect on the contact maps observed when applying APC is illustrated in Figure 1.4.

1.5 Evaluating Contact Prediction Methods

Choosing an appropriate benchmark for contact prediction methods is determined by the further utilization of the predictions. Most prominently, predicted contacts are used to assist structure prediction as outlined in section 1.2.3. Therefore, one could assess the quality of structural models computed with the help of predicted contacts. However, predicting structural models adds not only another layer of computational complexity but also raises questions about implementation details of the folding protocol. It has been found that in general a small number of

accurate contacts is sufficient to constrain the overal protein fold as discussed in section 1.2.3.

From these considerations emerged various standard benchmarks that have been established by the CASP community over many years [71,93,94]. CASP, the well-respected and independent competition for the structural bioinformatic's community introduced the contact prediction category in 1996. Taking place every two years, the progress in the field is assessed in a blind competition and the community discusses the outcome in a subsequent meeting.

According to the CASP regulations, a pair of residues is defined to be in physical contact when the distance between their C_β atoms ($C\alpha$ in case of glycine) is less than 8 $\mathring{A}$ in the reference protein structure. The overall performance of a contact predictor is evaluated by the mean precision over a testset of proteins with known high quality 3D structures against the top scoring predictions from every protein. The number of top scoring predictions per protein is typically normalized with respect to protein length L and precision is defined as the number of true contacts among the top scoring predicted contacts.

A popular variant of this benchmark plot shows the mean precision of a certain fraction of top ranked predictions (e.g. L/5 top ranked predictions) against specific properties of the test proteins such as protein length or alignment depth [95].

During CASP11 further evaluation metrics have been introduced, such as Matthews correlation coefficient, area under the precision-recall curve or F1 measure but they are rarely used in studies [71].

Currently best methods perform in the range XXX TODOOOPLOT

1.5.1 Sequence Separation

Local residue pairs separated by only some positions in sequence (e.g $|i-j| < 6$) are usually filtered out for evaluation of contact prediction methods. They are trivial to predict as they typically correspond to contacts within secondary structure elements and reflect the local geometrical constraints. Figure 1.5 shows the distribution of C_β distances for various minimal sequence separation thresholds.

Without filtering local residue pairs (sequence separation 1), there are several additional peaks in the distribution around 5.5$\mathring{A}$, 7.4$\mathring{A}$ and 10.6$\mathring{A}$ that can be attributed to local interactions in e.g. helices (see Figure 1.6).

Commonly, sequence separation bins are applied to distuinguish short ($6 < |i-j| \leq 12$), medium ($12 < |i-j| \leq 24$) and long range ($|i-j| > 24$) contacts [71,94]. Especially long range contacts are of importance for structure prediction as they are the most informative and able to constrain the overal fold of a protein [**???**].

1.6 Challenges in Coevolutionary Inference

Coevolution methods face several challenges when interpreting the covariation signals obtained from a MSA. Some of these challenges have been successfully met

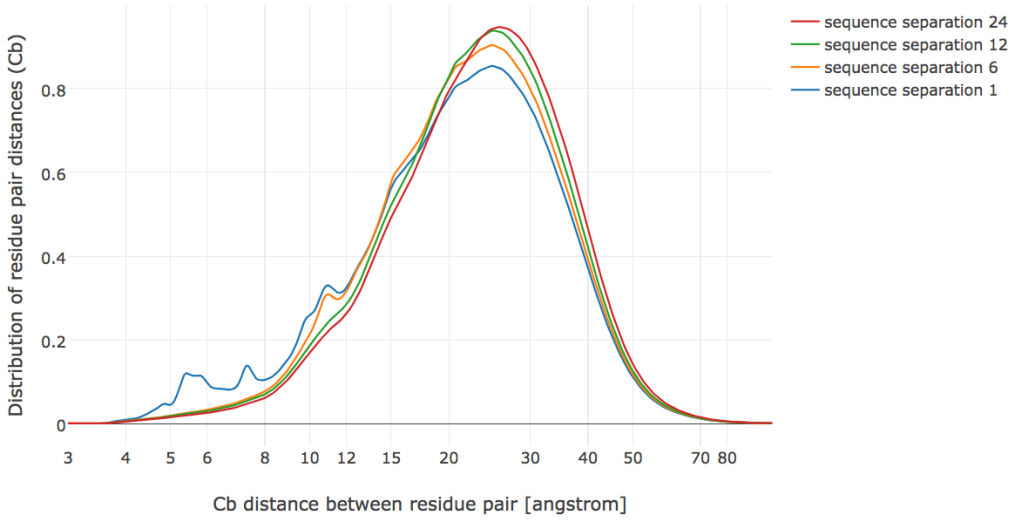


Figure 1.5: Distribution of residue pair C_β distances over 6741 proteins in the dataset (see Methods 6.1) at different minimal sequence separation thresholds.

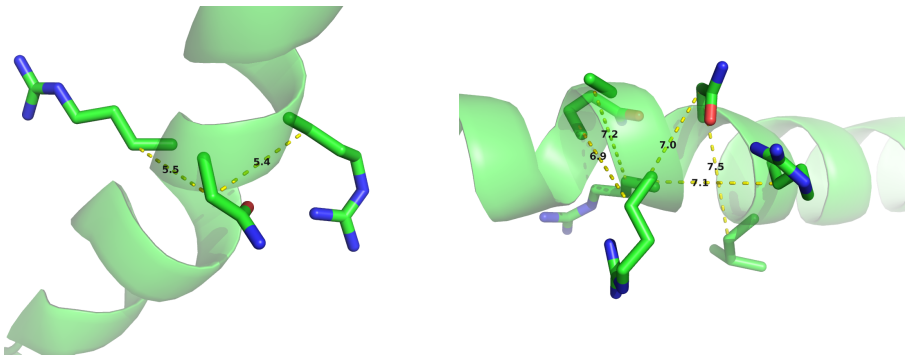


Figure 1.6: C_β distances between neighboring residues in α-helices. Left: Direct neighbors in α-helices have C_β distances around $5.4\mathring{A}$ due to the geometrical constraints from α-helical architecture. Right: Residues separated by two positions ($|i-j| = 2$) are less geometrically restricted to C_β distances between $7\mathring{A}$ and $7.5\mathring{A}$.

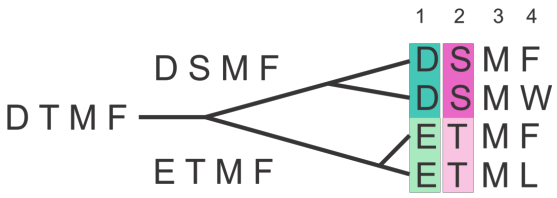


Figure 1.7: The phylogenetic dependence of closely related sequences can produce covariation signals. Here, two independent mutation events in two branches of the tree result in a perfect covariation signal for two positions.

(e.g. disentangling transitive effects with global statistical models), others are still open or open up new perspectives, such as dissecting different sources of coevolution signals.

Phylogenetic Bias

Sequences in MSAs do not represent independent samples of a protein family. In fact, there is selection bias from sequencing species of special interest (e.g human pathogens) or sequencing closely related species, e.g multiple strains. This uneven sampling of a protein family's sequence space leaves certain regions unexplored whereas others are statistically overrepresented [76,78,96].

Furthermore, due to their evolutionary relationship, sequences of a protein family have a complicated dependence structure. Closely related sequences can cause spurious correlations between positions, as there was not sufficient time for the sequences to diverge from their common ancestor [40,46,49]. Figure 1.7 illustrates a simplified example, where dependence of sequences due to phylogeny leads to a covariation signal.

To reduce the effects of redundant sequences, a popular sequence reweighting strategy has been found to improve contact prediction performance, where every sequence receives a weight that is the inverse of the number of similar sequences according to an identity threshold (see section 6.2.3) [45,76,78,97].

Entropic bias

Another source for noise is entropy bias that is closely linked to phylogenetic effects. By nature, methods detecting signals from correlated mutations rely on a certain degree of covariation between sequence positions [49]. Highly conserved interactions pose a conceptual challenge, as changes from one amino acid to another cannot be detected if sequences do not vary. This results in generally higher coevolution signals from positions with high entropy and underestimated signals for highly conserved interactions [38].

Several heuristics have been proposed to reduce entropy effects, such as Row-Column-Weighting (RCW) [40] or Average Product Correction (APC) [41] (see section 1.4.4).

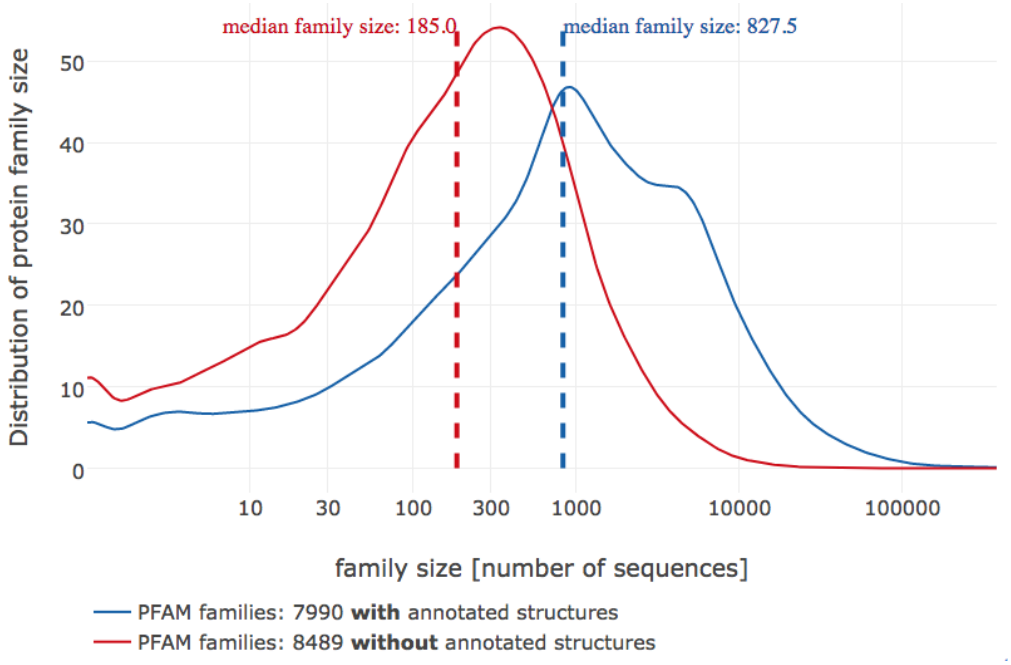


Figure 1.8: Distribution of PFAM family sizes. Less than half of the families in PFAM (7990 compared to 8489 families) do not have an annotated structure. The median family size in number of sequences for families with and without annotated structures is 185 and 827 respectively. Data taken from PFAM 31.0 (March 2017, 16712 entries) [99].

Finite Sampling Effects

Spurious correlations can arise from random statistical noise and blur true coevolution signals especially in low data scenarios. Consequently, false positive predictions attributable to random noise accumulate for protein families comprising low numbers of homologous sequences. This relationship was confirmed in many studies and as a rule of thumb it has been argued that proteins with L residues need at least $5L$ sequences in order to obtain confident predictions that can bet used for protein structure prediction [84,96]. Recently it was shown that precision of predicted contacts saturates for protein families with more than 10^3 diverse sequences and that precision is only dependent on protein length for families with small number of sequences [98].

Interesting targets for contact prediction are protein families without any associated structural information. As can be seen in Figure 1.8, those protein families generally comprise low numbers of homologous sequences with a median of 185 sequences per family and are thus susceptible to finite sampling effects.

With the rapidly increasing size of protein sequence databases (see section 1) the number of protein families with enough sequences for accuarate contact predictions will increase steadily [11,84]. Nevertheless, because of the already mentioned sequencing biases, better and more sensitive statistical models are indespensible to extend the applicability domain of coevolutionary methods.

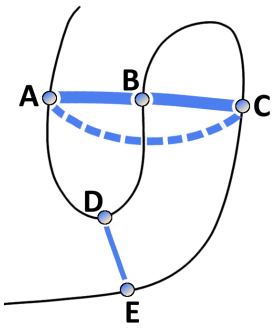


Figure 1.9: Effects of chained covariation obscure signals from true physical interactions. Consider residues A through E with physical interactions between the residue pairs A-B, B-C and D-E. The thickness of the lines between residues reflects the strength of statistical dependencies between the corresponding alignment columns. Strong statistical dependencies between residue pairs (A,B) and (B,C) can induce a strong dependencies between the spatially distant residues A and C. Covariation signals arising from transitive effects can become even stronger than other direct covariation signals and lead to false positive predictions.

Transitive Effects

One important shortcoming of traditional covariance approaches arises from the fact that chains of amino acid interactions are very common in protein structures and lead to direct as well as indirect correlation signals [46,47,49].

The concept of transitve effects is illustrated in Figure 1.9. Considering three residues A, B and C, where A physically interacts with B and B with C. Strong statistical dependencies between pairs (A,B) and (B,C) can induce strong indirect signals for residues A and C, eventhough they are not physically interacting. These indirect correlations can be even larger than signals of other directly interacting pairs (D,E) and thus lead to false predictions [49].

Local statistical methods, being introduced in section 1.3.1, are unable to disentangle these transitive effects as they consider residue pairs independent of one another. In contrast, global statistical models presented in section 1.3.2 learn a joint probability distribution over all residues allowing to dissect direct and indirect correlations [47,49].

Multiple Sequence Alignments

A correct MSA is the essential starting point for coevolution analysis as incorrectly aligned residues will confound the true signal. Highly sensitive and accurate alignment tools such as HHblits generate high quality alignments suitable for contact prediction [100]. However, there are certain subtleties to be kept in mind when generating alignments.

For example, proteins with repeated stretches of amino acids or with regions of low complexity are notoriously hard to align. Especially, repeat proteins have been found to produce many false positive contact predictions [98]. Therefore, MSAs

need to be generated with great care and covariation methods need to be tailored to these specific types of proteins [101,102].

Sensitivity of sequence search is critically dependent on the research question at hand and on the protein family under study. Many diverse sequences in general increase precision of predictions [95,103]. However, deep alignments can capture coevolutionary signals from different subfamilies. If only a specific subfamily is of interest, many false predictions might arise from strong coevolutionary signals specific to another subfamily that constitutes a prominent subset in the alignment. Therefore, a trade-off between specificity and diversity of the alignment is required to reach optimal results [104].

Another intrinsic characteristic of MSAs are repeated stretches of gaps that result from commonly utilized gap-penalty schemes assigning large penalties to insert a gap and lower penalties to gap extensions. Most statistical models treat gaps as the 21st amino acid, thus introducing an imbalance as gaps and amino acids express different behaviours which often results in gap-induced artefacts [91].

Evaluation Strategy

Contact prediction methods are typically evaluated based on a rigid definition of a residue contact. The standard definition of a true contact constitutes an $8\mathring{A}$ C_β distance cutoff as discussed in section 1.5.

However, whether two residues truly interact in a protein structure depends only marginally on the distance between their C_β atoms. More importantly, interactions between side-chains depend on their physico-chemical properties, on their orientation and different environments within proteins [105] (see section 1.1.1). A simple C_β distance threshold not only misses to reflect biological interaction preferences of amino acids but also provides a questionable gold-standard for benchmarking.

Other distance thresholds and definitions for physical contacts (e.g minimal atomic distances or distance between functional groups) have been studied as well. In fact, Duarte and colleagues found that using a C_β distance threshold between $9\mathring{A}$ and $11\mathring{A}$ yields optimal results when predicting the 3D structure from the respective contacts [106]. Anishchenko and colleagues [98] analysed false positive predictions with respect to a minimal atom distance threshold $< 5\mathring{A}$, as they found that this cutoff optimally defines direct physical interactions of residue pairs.

Definitely, choosing different distance cutoffs and reference atoms for defining a true contact changes the evaluation outcome. These crucial specifications need to be considered when evaluating methods and when comparing independent benchmarks that utilize different definitions.

Related to the problem of choosing the right trade-off between sensitivity and specificity when generating alignments is the issue of structural variation within a protein family. Evolutionary couplings are inferred from all family memebers in the MSA and thus might be physical contacts in one family member but not in another. Anishchenko et al. could show that more than 80% of false positives at intermediate distances (minimal heavy atom distance 5-15$\mathring{A}$) are true contacts in at least one homolog structure [98].

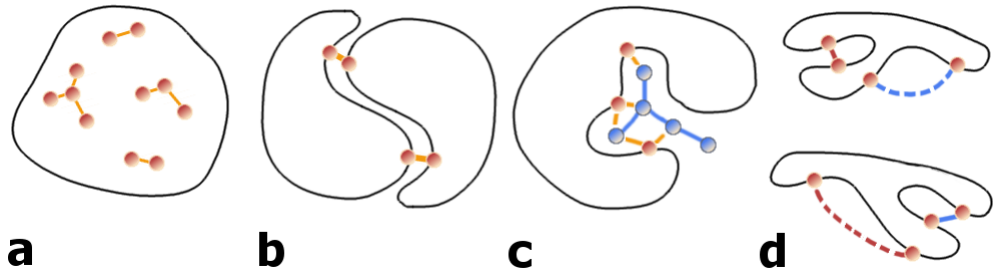


Figure 1.10: Possible sources of coevolutionary signals. **a)** Physical interactions between intra-domain residues. **b)** Interactions across the interface of predominantly homo-oligomeric complexes. **c)** Interactions mediated by ligands or metal atoms. **d)** Transient interactions due to conformational flexibility.

Alternative Sources of Coevolution

Coevolutionary signals can not only arise from intra-domain contacts, but also from other sources, like homo-oligomeric contacts, alternative conformations, ligand-mediated interactions or even contacts over hetero-oligomeric interfaces (see Figure 1.10) [96]. With the objective to predict physical contacts it is therefore necessary to identify and filter these alternative sources of coevolutionary couplings.

Many proteins form homo-oligomers with evolutionary conserved interaction surfaces. Currently it is hard to reliably distinguish intra- and inter-molecular contacts. Anishchenko et al. found that approximately one third of strong co-evolutionary signals between residue pairs at long distances (minimal heavy atom distance $>15\mathring{A}$) can be attributed to interactions across homo-oligomeric interfaces [98]. Several studies specifically analysed co-evolution across homo-oligomeric interfaces for proteins of known structure by filtering for residue pairs with strong couplings at long distances [104,107–110] or used co-evolutionary signals to predict homo-dimeric complexes [111].

It has been proposed that co-evolutionary signals can also arise from ligand or atom mediated interactions between residues or from critical interactions in intermediate folding states [97,112]. Confirming this hypothesis, a study showed that the cumulative strength of couplings for a particular residue can be used to predict functional sites [96,104].

Another important aspect is conformational flexibility. PDB structures used to evaluate coevolution methods represent only rigid snapshots taken in an unnatural crystalline environment. Yet proteins possess huge conformational plasticity and can adopt distinct alternative conformations or adapt shape when interacting with other proteins in an induced fit manner [113]. Several studies demonstrated successfully that coevolutionary signals can capture interactions specific to different distinct conformations [76,104,110,114].

1.7 Developing a Bayesian Model for Contact Prediction

Coevolution methods ad

The most popular and successfull methods for contact prediction optimize the pseudo-log-likelihood of the MSA and use several heuristics to calculate a contact score (see section 1.4.4).

By doing so valuable information in contact matrices is lost. Analyses in section 1 shows what information is contained in coupling matrices and that the signal in coupling matrices varies with C_β distance.

This thesis introcudes a principled Bayesian statistical approach that eradicates these heuristics to fully exploit the information in coupling matrices. Instead of transforming the model parameters $\mathbf{w}$ into heuristic contact scores, one can compute the posterior probability distributions of the distances r_{ij} between C_β atoms of all residues pairs i and j, given the MSA $\mathbf{X}$. The coupling parameters $\mathbf{w}$ are treated as hidden variables that will be integrated out analytically. This approach also allows for extraction of information contained in the particular types of amino acids, since each pair of amino acids will have a different preference to be coupled at certain distances.

TODO Figure ! !

In section 2 introduces max ent model for protein families that will produce the model parameters for the Bayesian model.

In section 3 describes in detail how the posterior distribution of distances can be computed.

Section 4 presents the optimizaton of the coupling prior.

And the Bayesian model will be evalutated in section 5.

The outlook describes an extension of the model to predict inter-residue distances. Development is ongoing.

2

Interpretation of Coupling Matrices

Contact prediction methods learning a *Potts model* for the MSA of a protein familiy, map the inferred 20 x 20 coupling matrices w_{ij} onto scalar values to obtain contact scores for each residue pair as outlined in section 1.4.4. By doing so, the full information contained in coupling matrices is lost:

- the contribution of individual couplings w_{ijab}
- the direction of couplings (positive or negative)
- the correlation between couplings w_{ijab} and w_{ijcd}
- inherent biological meaning

The following analyses give some intuition for the information contained in coupling matrices.

2.1 Single Coupling Values Carry Evidence of Contacts

Given the success of DCA methods, it is clear that the inferred couplings $\mathbf{w}_{ij}$ are good indicators of spatial proximity for residue pairs. As described in section 1.4.4, a contact score for a residue pair is commonly computed as the Frobenius norm over the coupling matrix: $||\mathbf{w}_{ij}||_2 = \sqrt{\sum_{a,b=1}^{20} w_{ijab}}$

The left plot in Figure 2.1 shows the correlation between squared coupling values $(w_{ijab})^2$ and binary contact class (contact=1, non-contact=0) for approximately 100.000 residue pairs per class (for details see methods section 6.3.1). All couplings have a positive class correlation, meaning the stronger the squared coupling value, the more likely a contact can be inferred. Generally, couplings that involve any aliphatic amino acid (I, L, V) or alanine express the strongest class correlation. In contrast, C-C or pairs involving charged residus (R, E, K, D) or tryptophane correlate only weakly with contact class. Interestingly, C-C and couplings involving

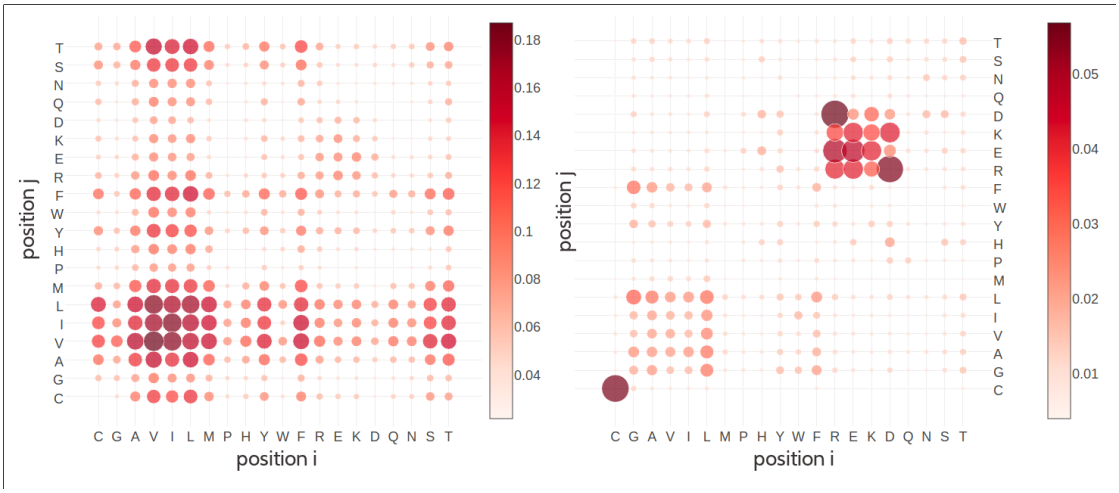


Figure 2.1: **Left** Pearson correlation of squared coupling values $(w_{ijab})^2$ with contact class (contact=1, non-contact=0). **Right** Standard deviation of squared coupling values. Dataset contains 100.000 residue pairs per class (for details see methods section 6.3.1). Contacts are defined as residue pairs with $C_\beta < 8\mathring{A}$ and non-contacts as residue pairs with $C_\beta > 25\mathring{A}$.

charged residues have the highest standard-deviation among all couplings as can be seen in the right plot in Figure 2.1. It can be hypothesized that these couplings considerably contribute to false positive predictions when using the Frobenius norm as a contact score; they can have high squared values (high standard deviation) that do not correlate well with being a contact.

Apparantly, different couplings are of varying importance for contact inference and have distinct characteristics. But looking at the raw coupling values (without squaring), these charateristics become even more pronounced.

The left plot in Figure 2.2 shows the correlation of raw coupling values w_{ijab} with contact class. Interestingly, in contrast to the findings for squared coupling values, couplings for charged residue pairs (R, E, K, D) have the strongest class correlation (positive and negative), whereas aliphatic couplings correlate to a much lesser extent. This implies that absolute (squared) coupling strength for aliphatic couplings is a better indicator for contacts than the raw signed coupling value. On the contrary, the raw signed coupling values for charged residue pairs are much more indicative of a contact than the sheer magnitude of their squared values.

Raw couplings for aromatic pairs or C-C pairs correlate only weakly with contact class. For these pairs neither coupling strength, nor the sign of the coupling value seems to be a good indicator for a contact.

Of course, looking only at correlations can be misleading if there are non-linear patterns in the data, for example higher order dependencies between couplings. For this reason it is advisable to take a more detailed view at coupling matrices and the distributions of their values.

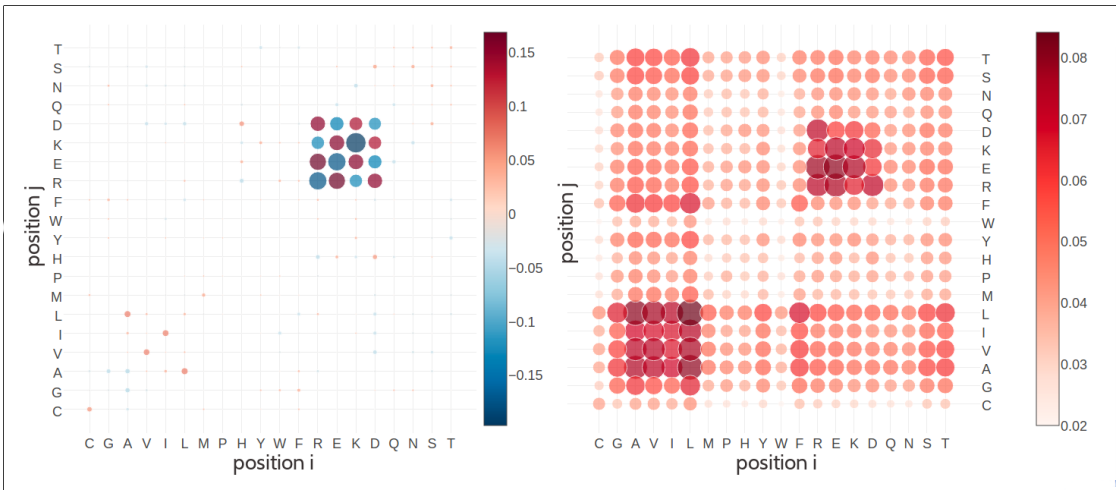


Figure 2.2: **Left** Pearson correlation of raw signed coupling values w_{ijab} with contact class (contact=1, non-contact=0). **Right** Standard deviation of coupling values. Dataset contains 100.000 residue pairs per class (for details see section 6.3.1). Contacts are defined as residue pairs with $C_\beta < 8\mathring{A}$ and non-contacts as residue pairs with $C_\beta > 25\mathring{A}$.

2.2 Physico-Chemical Fingerprints in Coupling Matrices

The correlation analysis of coupling matrices in the last section revealed that certain couplings are more indicative of a contact than others. Individual coupling matrices for a residue pair that is in physcial contact often display striking patterns that agree with the previous findings. Most often, these patterns allow a biological interpreation of the coupling values that reveal details of the interdependency between both residues.

Figure 2.3 visualizes the inferred coupling matrix for a residue pair using the pseudo-likelihood method. Clearly visible is a cluster of strong coupling values for charged and polar residues (E,D,K,R,Q). Positive coupling values can be observed between positively charged residues (R,K) and negatively charged residues (E,D), whereas couplings between equally charged residues have negative values. The coupling matrix perfectly reflects the interaction preference for residues forming salt bridges. Indeed, in the protein structure the first residue (glutamic acid) forms a salt bridge with the second residue (lysine) as can be seen in the left Figure 2.5.

Figure 2.4 visualizes the coupling matrix for a pair of hydrophobic residues. Hydrophobic pairings have strong coupling values but the couplings also reflect a sterical constraint. Alanine as a small hydrophobic residue is favoured at either of both residue positions as it has strong positive couplings with isoleucine, leucine and methionine. But alanine is disfavoured to appear at both positions at the same time as the A-A coupling is negative. Figure 2.5 illustrates the location of the two residues in the protein core. Here, hydrophobic residues are densely packed and the limited space allows for only small hydrophobic residues.

Many more biological interpretable signals can be identified from coupling matri-

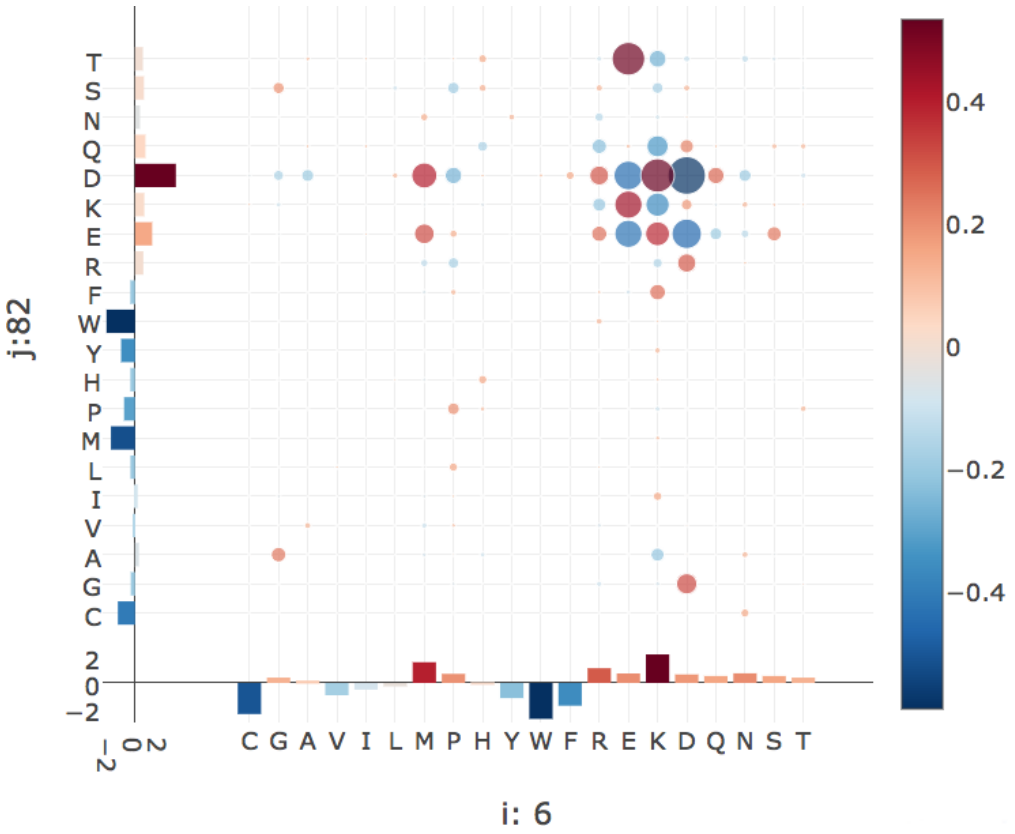


Figure 2.3: Coupling matrix computed with pseudo-likelihood for residues 6 and 82 in protein 1awq chain A. Size of the bubbles represents coupling strength and color represents positive (red) and negative (blue) coupling values. Bars at the x-axis and y-axis represent the corresponding single potentials for both residue positions. Height of the bars stands for potential strength and color for positive (red) and negative (blue) values.

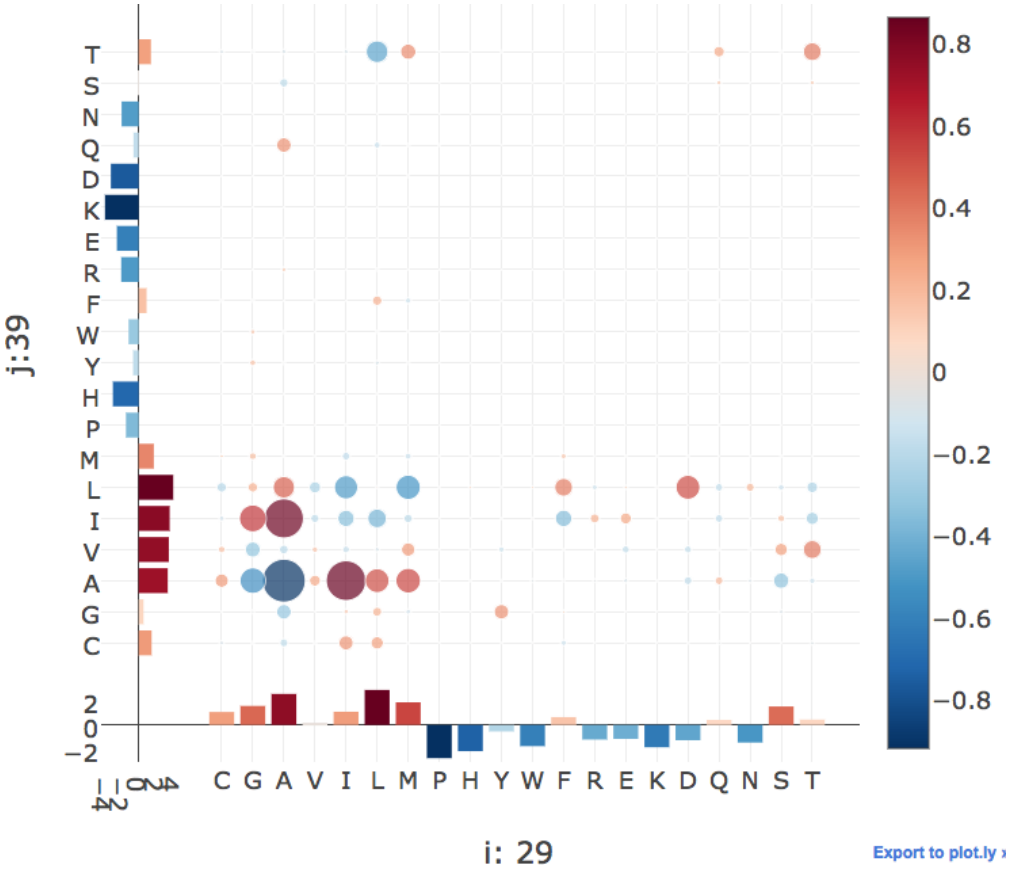


Figure 2.4: Coupling matrix computed with pseudo-likelihood for residues 29 and 39 in protein 1ae9 chain A. Size of the bubbles represents coupling strength and color represents positive (red) and negative (blue) coupling values. Bars at the x-axis and y-axis represent the corresponding single potentials for both residues. Height of the bars stands for potential strength and color for positive (red) and negative (blue) values.

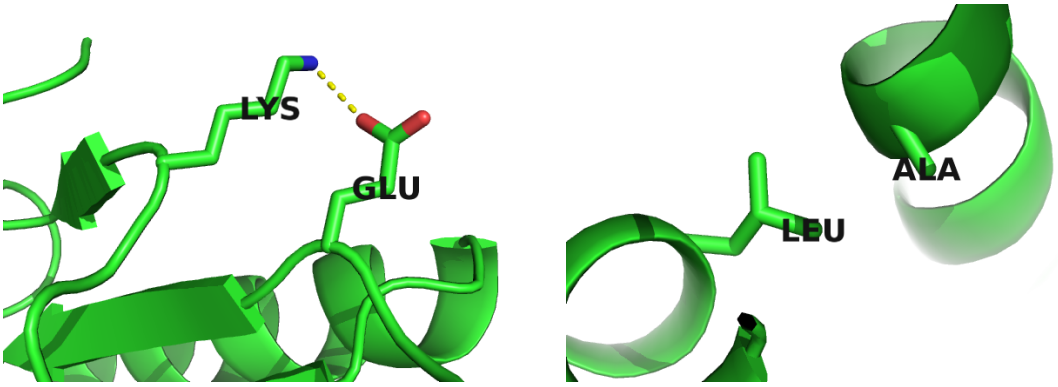


Figure 2.5: Interactions between protein side chains. **Left**: residue 6 (glutamic acid) forms a salt bridge with residue 82 (lysine) in protein 1awq, chain A. **Right**: residue 29 (alanine) and residue 39 (leucine) within the hydrophobic core of protein 1ae9 chain A.

ces, including pi-cation interactions (see Appendix C.1), aromatic-proline interactions (see Appendix C.3), sulfur-aromatic interactions or disulphide bonds (see Appendix C.2).

Coucke and collegues performed a thorough quantitative analysis of coupling matrices selected from confidently predicted residue pairs [115]. They showed that eigenmodes obtained from a spectral analysis of averaged coupling matrices are closely related to physico-chemical properties of amino acid interactions, like electrostaticity, hydrophobicity, steric interactions or disulphide bonds. By looking at specific populations of residues, like buried and exposed residues or residues from specific protein classes (small, mainly α, etc), the eigenmodes of corresponding coupling matrices are found to capture very characteristic interactions for each class, e.g. rare disulfide contacts within small proteins and hydrophilic contacts between exposed residues. Their study confirms the qualitative observations presented above that amino acid interactions can leave characteristic physico-chemical fingerprints in coupling matrices.

2.3 Coupling Profiles Vary with Distance

Analyses in the previous sections showed that certain coupling values correlate more or less strong with contact class and that coupling matrices for contacts express biological meaningfull patterns.

More insights can be obtained by looking at the distribution of distinct coupling values for contacts, non-contacts and arbitrary populations of residue pairs. Figure 2.6 shows the distribution of selected couplings for filtered residue pairs within a C_β distance $< 5\mathring{A}$ (see methods section 6.3.2 for details). The distribution of R-E and E-E coupling values is shifted and skewed towards positive and negative values respectively. This is in accordance with attracting electrostatic interactions between the positively charged side chain of arginine and the negatively charged side chain of gluatamic acid and also with repulsive interactions between the two negatively charged gluatamic acid side chains. Coupling values for cysteine pairs (C-C) have a broad distribution that is skewed towards positive values, reflecting the strong signals obtained from covalent disulphide bonds. The broad distribution for C-C, R-E and E-E agrees with the observation in section 2.1 that these specific coupling values have large standard deviations and that for charged residue pairings the signed coupling value is a strong indicator of a contact.

Hydrophobic pairs like V-I have an almost symmetric coupling distribution, confirming the finding that the direction of coupling is not indicative of a true contact whereas the strength of the coupling is. The hydrophobic effect that determines hydrophobic interactions is not specific or directed. Therefore, hydrophobic interaction partners can commonly be substituted by other hydrophobic residues, which explains the not very pronounced positive coupling signal compared to more specific interactions, e.g ionic interactions. The distribution of aromatic coupling values like F-W is slightly skewed towards negative values, accounting for steric hindrance of their large side chains at small distances.

In an intermediate C_β distance range between $8\mathring{A}$ and $12\mathring{A}$ the distributions for all coupling values are centered close to zero and are less broad. The distributions are

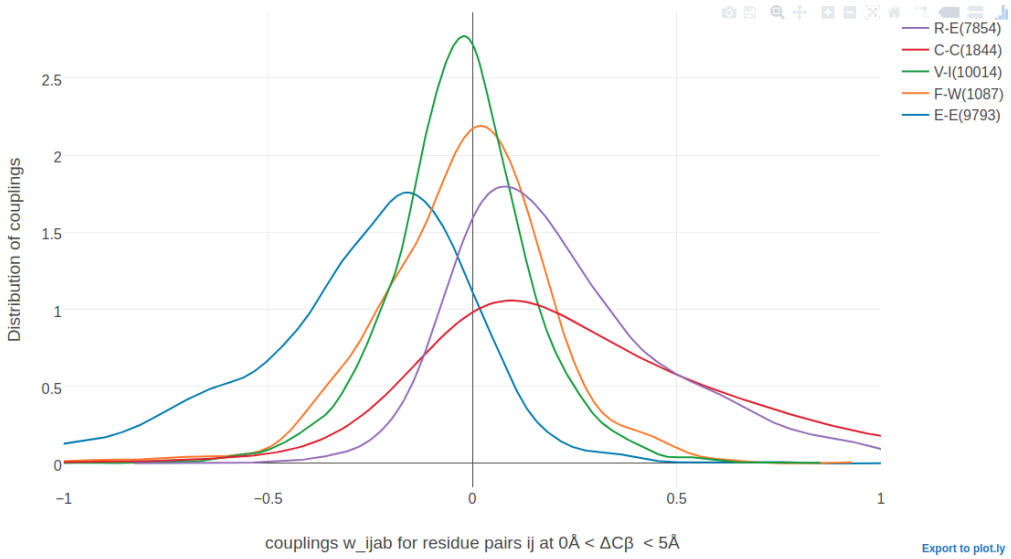


Figure 2.6: Distribution of selected couplings for filtered residue pairs with C_β distance $< 5\mathring{A}$ (see methods section 6.3.2 for details). Number of coupling values used to determine the distribution is given in brackets in the legend. R-E = coupling for pairings of arginine and glutamic acid, C-C = coupling for pairings between cystein residues, V-I = coupling for pairings of valine and isoleucine, F-W = coupling for pairing sof phenylalanine and tryptophane, E-E = coupling for pairings between glutamic acid residues.

still shifted and skewed, but much less pronounced as it has been observed for the distributions at C_β distance $< 5\mathring{A}$. For aromatic pairs like F-W, the distribution of coupling values has very long tails, suggesting rare but strong couplings for aroamtic side chains at this distance.

Figure 2.8 shows the distribution of selected couplings for residue pairs far apart in the protein structure (C_β distance $> 20\mathring{A}$).
The distribution for all couplings is centered at zero and has small variance. Only for C-C coupling values, the distribution has a long tail for positve values, presumably arising from the fact that the maximum entropy model cannot distuinguish highly conserved signals of multiple disulphide bonds within a protein. This observation also agrees with the previous finding in section 2.1 that C-C coupling values, albeit having large standard-deviations, correlate only weakly with contact class. The same arguments apply to couplings of aromatic pairs that have a comparably broad distribution and do not correlate strongly with contact class. The strong coevolution signals for aromatic pairs even at high distance ranges might be a result of insufficient disentangling of transitive effects, as aromatic residues are known to form network-like structures in the protein core that stabilize protein structure (see Figure C.7 in Appendix)[13].

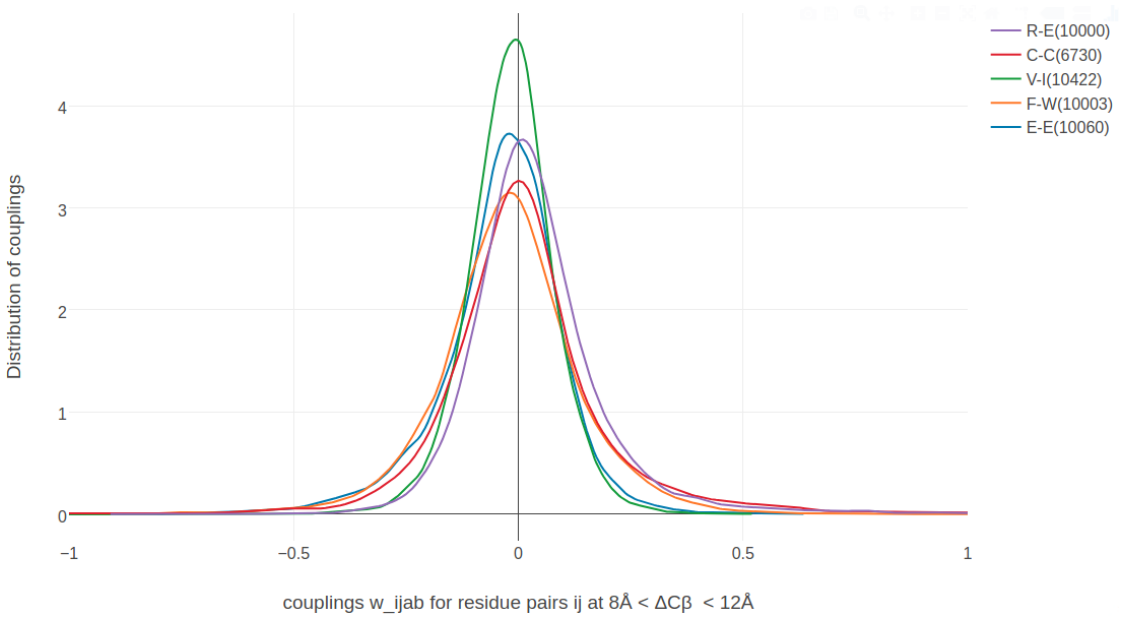


Figure 2.7: Distribution of selected couplings for filtered residue pairs with C_β distances between $8\mathring{A}$ and $12\mathring{A}$ (see methods section 6.3.2 for details). Number of coupling values used to determine the distribution is given in brackets in the legend. R-E = coupling for pairings of arginine and glutamic acid, C-C = coupling for pairings between cystein residues, V-I = coupling for pairings of valine and isoleucine, F-W = coupling for pairing sof phenylalanine and tryptophane, E-E = coupling for pairings between glutamic acid residues.

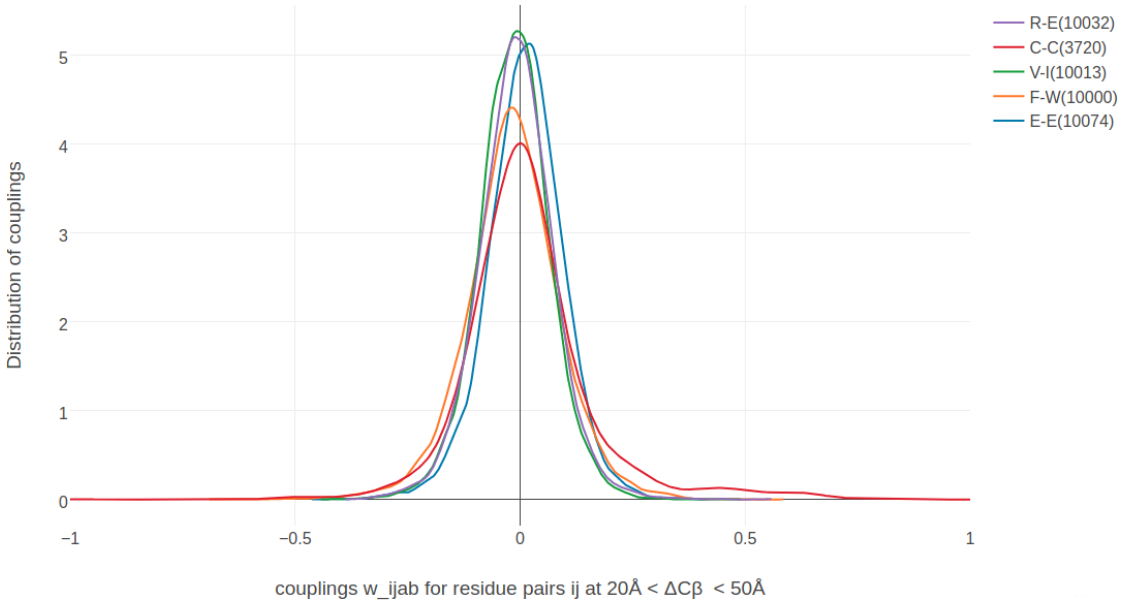
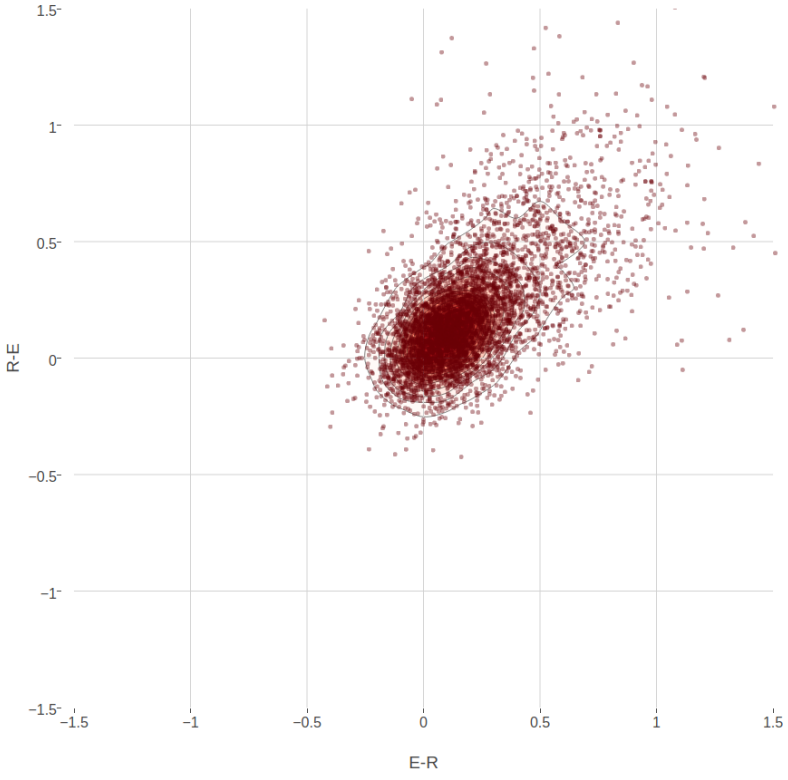


Figure 2.8: Distribution of selected couplings for filtered residue pairs with C_β distances between $20\mathring{A}$ and $50\mathring{A}$ (see methods section 6.3.2 for details). Number of coupling values used to determine the distribution is given in brackets in the legend. R-E = coupling for pairings of arginine and glutamic acid, C-C = coupling for pairings between cystein residues, V-I = coupling for pairings of valine and isoleucine, F-W = coupling for pairing sof phenylalanine and tryptophane, E-E = coupling for pairings between glutamic acid residues.

2.4 Higher Order Dependencies Between Couplings

The analyses in the previous sections focused on single coupling values picked from the 20×20-dimensional coupling matrices $\mathbf{w}_{ij}$. As mentioned before, analysing only single dimension might be misleading and further insights might be concealed in a higher order relationships. Unfortunately, it is not possible to reasonably visualize the high dimensional coupling matrices.

But taking a look at two dimensional coupling scatter plots already reveals some further insights and confirms the previous observations that couplings reflect biological relevant amino acid interactions. Figure **??** and 2.9 illustrate the distribution of couplings at C_β distances less than $8\mathring{A}$ between the attractive ionic pairings of E-R and R-E and between the ionic pairing R-E and the repulsive pair of equally charged residues E-E, respectively. Whereas coupling values for R-E and E-R are positively correlated, coupling values for R-E and E-E are negatively correlated.



\begin{figure}

Figure 2.10 shows distributions between couplings for hydrophobic pairings that are almost symmetric and broadly centered around zero. Coupling distributions for residue pairs that are not physically interacting ($C_\beta >> 8\mathring{A}$) resemble the distribution for hydrophobic pairings in that there is no correlation, but at high distance the distributions are much tighter centered around zero.

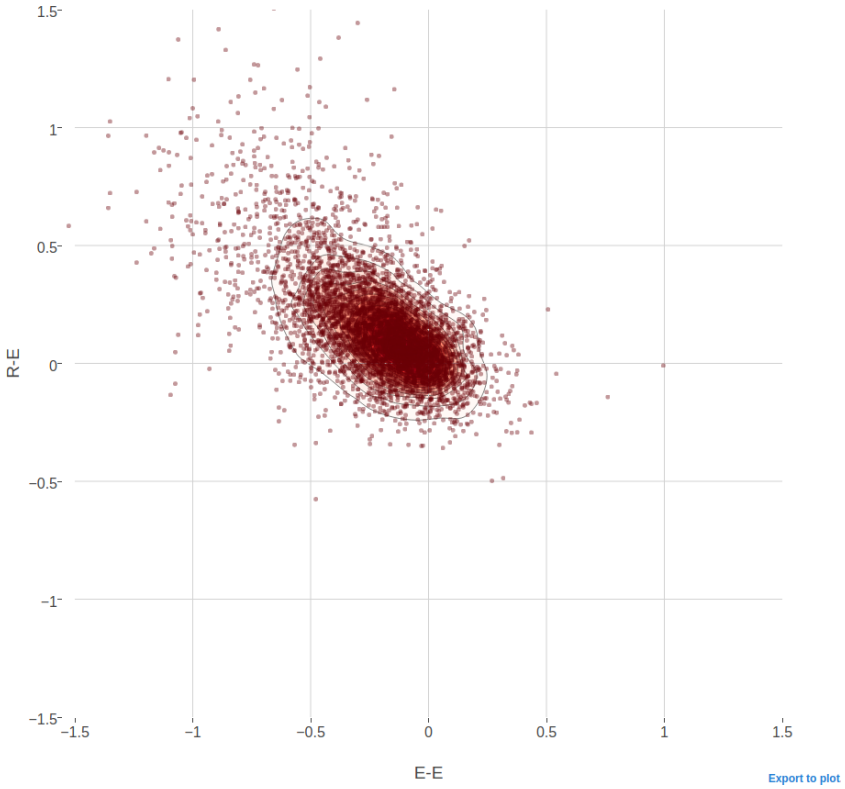


Figure 2.9: Two-dimensional distribution of coupling values R-E and E-E for approximately 10000 residue pairs with $\Delta C_\beta < 8\mathring{A}$. The distribution is almost symmetric and the coupling values are negatively correlated. Residue pairs have been filtered for sequence separation, percentage of gaps and evidence in alignment as explained in methods section 6.3.2.

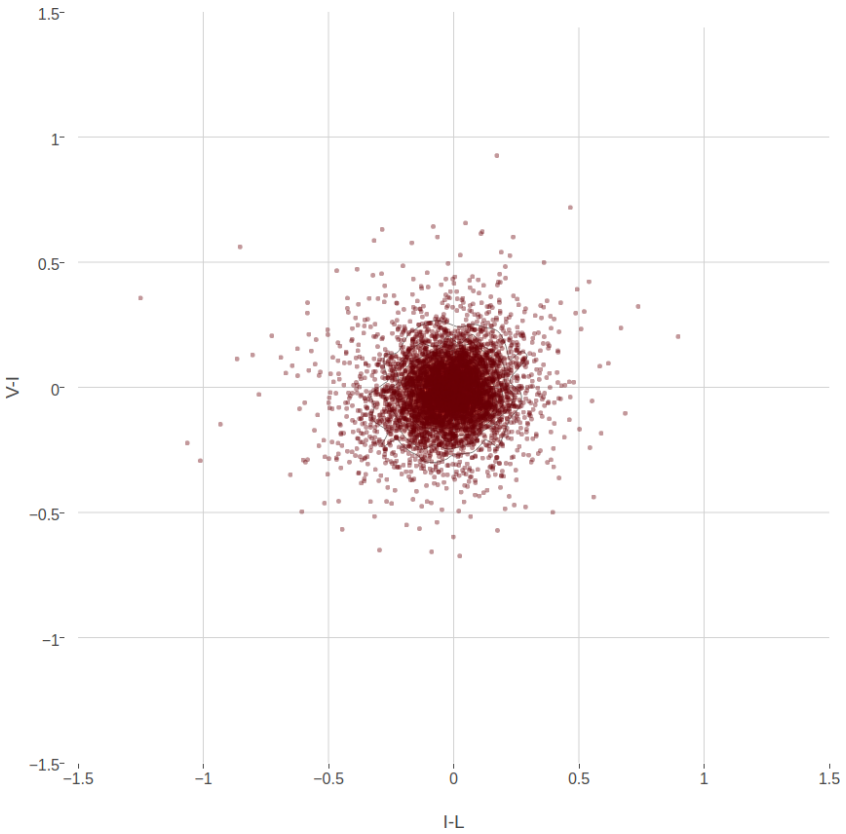


Figure 2.10: Two-dimensional distribution of coupling values R-E and E-E for approximately 10000 residue pairs with $\Delta C_\beta < 8\mathring{A}$. The coupling values are symmetrically distributed around zero without visible correlation. Residue pairs have been filtered for sequence separation, percentage of gaps and evidence in alignment as explained in methods section 6.3.2.

3

Optimizing the Full-Likelihood

Section 1.4 introduced the *Potts model* for contact prediction that is able to distinguish between directly and indirectly coupled residue pairs by jointly modelling the probabilty of a protein sequence over all residues. Maximum-likelihood inference of the model parameters is numerically challenging due to the exponential complexity of the partition function that normalizes the probability distribution. Several approximate inference techniques for the full likelihood have been developed trying to sidestep the exact computation of the partition function. At this point in time, pseudo-likelihood is the most successful approximate solution with regard to the specific problem of predicting residue-residue contacts. It has been shown that the pseudo-likelihood is a consistent estimator to the full likelihood in the limit of large amounts of data, however, it is unclear whether it represents a good approximation when there is only little data, in other words for small protein families that are typical of contact prediction.

Computing the gradient of the likelihood analytically is infeasible, because computing $p(x_i = a, x_j = b|\mathbf{v}, \mathbf{w}) = \sum_{y_1,\ldots,y_L=1}^{20} p(y_1, \ldots, y_L|\mathbf{v}, \mathbf{w}) I(y_i = a, y_j = b)$ would require summing over 20^L sequences $(y_1, \ldots, y_L)$. Several approaches have been used to get around this problem as described in section **??**. The most popular one for protein contact prediction is to optimize the pseudo likelihood instead (see section 1.4.3.1). Its gradient involves a sum over just the 20 amino acids instead of over all possible sequences of length L.

While the function value of the full likelihood cannot efficiently be computed, it is possible to approximate the gradient of the full likelihood with an approach called *contrastive divergence* that makes use of MCMC sampling techniques [116].

In the next sections I am going to specify the exact model that differs from the *Potts model* as it is used for pseudo-likelihood inference as explained in section @ref(pseudo-likelihood}) in some small details. Then I will discuss how the gradient of the likelihood can be approximated with *contrastive divergence* and present the results for this optimization stratedgy.

3.1 The Likelihood of the Sequences as a Potts Model

The N sequences of the MSA $\mathbf{X}$ are denoted as $\mathbf{x}_1, ..., \mathbf{x}_N$. Each sequence $\mathbf{x}_n = (\mathbf{x}_{n1}, ..., \mathbf{x}_{nL})$ is a string of L letters from an alphabet indexed by $\{0, ..., 20\}$, where 0 stands for a gap and $\{1, ..., 20\}$ stand for the 20 types of amino acids.

As already described in detail in section 1.4, the likelihood of the sequences in the MSA of the protein family is modelled with a *Potts Model*:

$$\begin{aligned} p(\mathbf{X}|\mathbf{v}, \mathbf{w}) &= \prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{v}, \mathbf{w}) \\ &= \prod_{n=1}^{N} \frac{1}{Z(\mathbf{v}, \mathbf{w})} \exp\left(\sum_{i=1}^{L} v_i(x_{ni}) \sum_{1 \le i < j \le L} w_{ij}(x_{ni}, x_{nj})\right) \end{aligned} \tag{3.1}$$

The coefficients v_{ia} and w_{ijab} are referred to as single potentials and couplings, respectively that describe the tendency of an amino acid a (and b) to (co-)occur at the respective positions in the MSA. $Z(\mathbf{v}, \mathbf{w})$ is the partition function that normalizes the probability distribution $p(\mathbf{x}_n|\mathbf{v}, \mathbf{w})$:

$$Z(\mathbf{v}, \mathbf{w}) = \sum_{y_1, ..., y_L = 1}^{20} \exp\left(\sum_{i=1}^{L} v_i(y_i) \sum_{1 \le i < j \le L} w_{ij}(y_i, y_j)\right) \tag{3.2}$$

3.2 Treating Gaps as Missing Information

Treating gaps explicitly as 0'th letter of the alphabet will lead to couplings between columns that are not in physical contact. To see why, imagine a hypothetical alignment consisting of two sets of sequences as it is illustrated in Figure 3.1. The first set has sequences covering only the left half of columns in the MSA, while the second set has sequences covering only the right half of columns. The two blocks could correspond to protein domains that were aligned to a single query sequence.

Now consider couplings between a pair of columns i, j with i from the left half and j from the right half. Since no sequence (except the single query sequence) overlaps both domains, the empirical amino acid pair frequencies $q(x_i = a, x_j = b)$ will vanish for all $a, b \in \{1, ..., L\}$.

The gradient of the log likelihood for couplings is

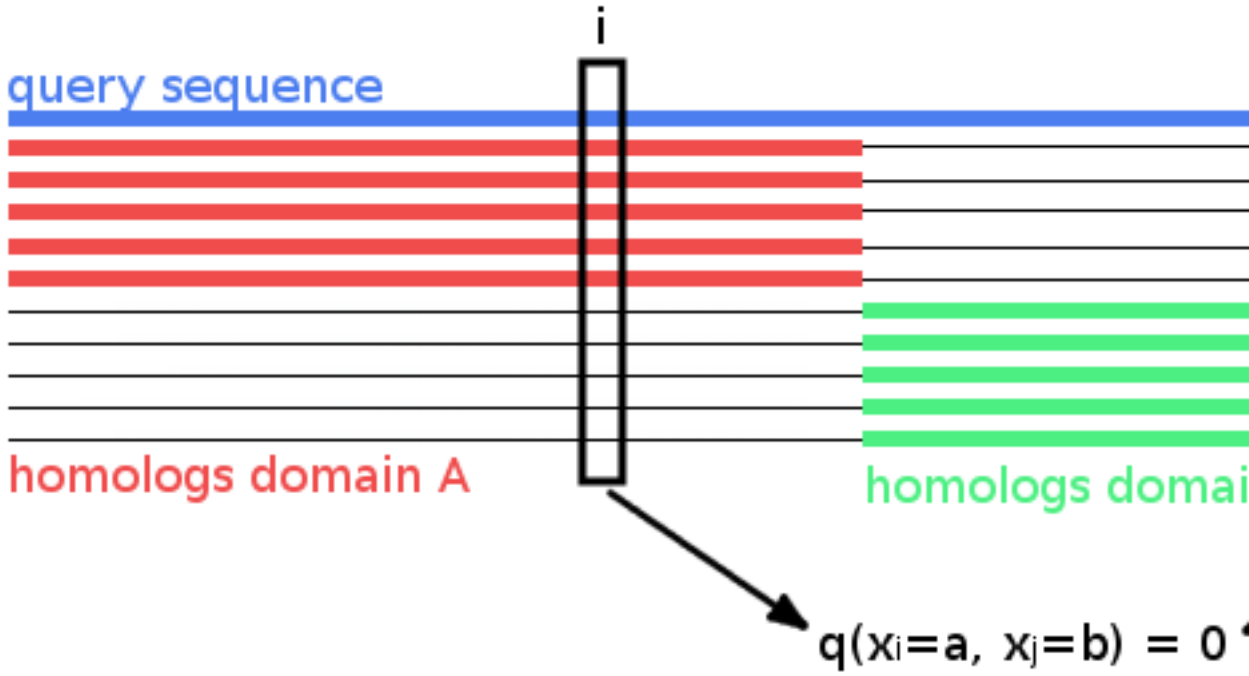


Figure 3.1: Hypothetical MSA consisting of two sets of sequences: the first set has sequences covering only the left half of columns, while the second set has sequences covering only the right half of columns. The two blocks could correspond to protein domains that were aligned to a single query sequence. Empirical amino acid pair frequencies $q(x_i = a, x_j = b)$ will vanish for positions i from the left half and j from the right half of the alignment.

$$\frac{\partial LL}{\partial w_{ijab}} = \sum_{n=1}^{N} I(x_{ni} = a, x_{nj} = b) - N \frac{\partial}{\partial w_{ijab}} \log Z(\mathbf{v}, \mathbf{w}) \tag{3.3}$$

$$= \sum_{n=1}^{N} I(x_{ni}{=}a, x_{nj}{=}b) \tag{3.4}$$

$$- N \sum_{y_1,\ldots,y_L=1}^{20} \frac{\exp\left(\sum_{i=1}^{L} v_i(y_i) + \sum_{1\le i<j\le L} w_{ij}(y_i, y_j)\right)}{Z(\mathbf{v}, \mathbf{w})} I(y_i{=}a, y_j{=}b) \tag{3.5}$$

$$= Nq(x_i{=}a, x_j{=}b) - N \sum_{y_1,\ldots,y_L=1}^{20} p(y_1, \ldots, y_L|\mathbf{v}, \mathbf{w})\, I(y_i{=}a, y_j{=}b) \tag{3.6}$$

$$= Nq(x_i{=}a, x_j{=}b) - Np(x_i{=}a, x_j{=}b|\mathbf{v}, \mathbf{w}) \tag{3.7}$$

Note that the empirical frequencies $q(x_i{=}a, x_j{=}b)$ are equal to the model probabilities $p(x_i{=}a, x_j{=}b|\mathbf{v}, \mathbf{w})$ at the maximum of the likelihood when the gradient vanishes. Therefore, $p(x_i{=}a, x_j{=}b|\mathbf{v}, \mathbf{w})$ would have to be zero in the optimum when the empirical amino acid frequencies $q(x_i = a, x_j = b)$ vanish for pairs of columns as described above. However, $p(x_i{=}a, x_j{=}b|\mathbf{v}, \mathbf{w})$ can only become zero, when the exponential term is zero, which would only be possible if w_{ijab} goes to ∞. This is clearly undesirable, as physical contacts will be deduced from the size of the couplings.

The solution is to treat gaps as missing information. This means that the normalisation of $p(\mathbf{x}_n|\mathbf{v}, \mathbf{w})$ should not run over all positions $i \in \{1, ..., L\}$ but only over those i that are not gaps in $\mathbf{x}_n$. Therefore, the set of sequences S_n used for normalization of $p(\mathbf{x}_n|\mathbf{v}, \mathbf{w})$ in the partition function will be defined as:

$$S_n := \{(y_1, ..., y_L) : 0 \le y_i \le 20 \wedge (y_i{=}0 \text{ iff } x_{ni}{=}0)\} \tag{3.8}$$

and the partition function becomes:

$$Z_n(\mathbf{v}, \mathbf{w}) = \sum_{\mathbf{y}\in S_n} \exp\left(\sum_{i=1}^{L} v_i(y_i) \sum_{1\le i<j\le L} w_{ij}(y_i, y_j)\right) \tag{3.9}$$

To ensure that the gaps in x_n do not contribute anything to the sums, the parameters associated with a gap will be fixed to 0:

$$v_i(0) = \mathbf{w}_{ij}(0, b) = \mathbf{w}_{ij}(a, 0) = 0\ ,$$

for all $i, j \in \{1, ..., L\}$ and $a, b \in \{0, ..., 20\}$. Furthermore, the empirical amino acid frequencies q_{ia} and q_{ijab} need to be redefined such that they are normalised over $\{1, ..., 20\}$:

$$N_i := \sum_{n=1}^{N} I(x_{ni} \neq 0) \qquad q_{ia} = q(x_i = a) := \frac{1}{N_i} \sum_{n=1}^{N} I(x_{ni} = a) \tag{3.10}$$

$$N_{ij} := \sum_{n=1}^{N} I(x_{ni} \neq 0, x_{nj} \neq 0) \quad q_{ijab} = q(x_i = a, x_j = b) := \frac{1}{N_{ij}} \sum_{n=1}^{N} I(x_{ni} = a, x_{nj} = b) \tag{3.11}$$

With this definition, empirical amino acid frequencies are normalized without gaps, so that

$$\sum_{a=1}^{20} q_{ia} = 1 \ , \ \sum_{a,b=1}^{20} q_{ijab} = 1. \tag{3.12}$$

3.3 The Regularized Log Likelihood and its Gradient

As with pseudo-likelihood based methods, Gaussian priors $\mathcal{N}(\mathbf{v}|\mathbf{v}^*, \lambda_v^{-1}\mathbf{I})$ and $\mathcal{N}(\mathbf{w}|\mathbf{0}, \lambda_w^{-1}\mathbf{I})$ will be used to constrain the parameters $\mathbf{v}$ and $\mathbf{w}$ and to fix the Gauge (see section 1.4.3.1). The choice of v^* will be discussed in section 3.4. By including the logarithm of this prior into the log likelihood using the gap treatment described in the last section, the regularised likelihood is obtained,

$$LL_{\text{reg}}(\mathbf{v}, \mathbf{w}) = \log \left[p(\mathbf{X}|\mathbf{v}, \mathbf{w}) \, \mathcal{N}(\mathbf{v}|\mathbf{v}^*, \lambda_v^{-1}\mathbf{I}) \, \mathcal{N}(\mathbf{w}|\mathbf{0}, \lambda_w^{-1}\mathbf{I}) \right] \tag{3.13}$$

or explicitely,

$$LL_{\text{reg}}(\mathbf{v}, \mathbf{w}) = \sum_{n=1}^{N} \left[\sum_{i=1}^{L} v_i(x_{ni}) + \sum_{1 \leq i < j \leq L} w_{ij}(x_{ni}, x_{nj}) - \log Z_n(\mathbf{v}, \mathbf{w}) \right] \tag{3.14}$$

$$- \frac{\lambda_v}{2} \sum_{i=1}^{L} \sum_{a=1}^{20} (v_{ia} - v_{ia}^*)^2 - \frac{\lambda_w}{2} \sum_{1 \leq i < j \leq L} \sum_{a,b=1}^{20} w_{ijab}^2. \tag{3.15}$$

The gradient of the regularized log likelihood has single components

$$\frac{\partial LL_{\text{reg}}}{\partial v_{ia}} = \sum_{n=1}^{N} I(x_{ni} = a) - \sum_{n=1}^{N} \frac{\partial}{\partial v_{ia}} \log Z_n(\mathbf{v}, \mathbf{w}) - \lambda_v (v_{ia} - v_{ia}^*) \tag{3.16}$$

$$= N_i q(x_i = a) \tag{3.17}$$

$$- \sum_{n=1}^{N} \sum_{\mathbf{y} \in S_n} \frac{\exp\left(\sum_{i=1}^{L} v_i(y_i) + \sum_{1 \leq i < j \leq L}^{L} w_{ij}(y_i, y_j) \right)}{Z_n(\mathbf{v}, \mathbf{w})} I(y_i = a) \tag{3.18}$$

$$- \lambda_v (v_{ia} - v_{ia}^*) \tag{3.19}$$

and pair components

$$\frac{\partial LL_{\text{reg}}}{\partial w_{ijab}} = \sum_{n=1}^{N} I(x_{ni}=a, x_{nj}=b) - \sum_{n=1}^{N} \frac{\partial}{\partial w_{ijab}} \log Z_n(\mathbf{v}, \mathbf{w}) - \lambda_w w_{ijab} \tag{3.20}$$

$$= N_{ij} q(x_i=a, x_j=b) \tag{3.21}$$

$$- \sum_{n=1}^{N} \sum_{\mathbf{y} \in S_n} \frac{\exp\left(\sum_{i=1}^{L} v_i(y_i) + \sum_{1 \le i < j \le L}^{L} w_{ij}(y_i, y_j)\right)}{Z_n(\mathbf{v}, \mathbf{w})} I(y_i=a, y_j=b) \tag{3.22}$$

$$- \lambda_w w_{ijab} \tag{3.23}$$

Note that (without regulariation $\lambda_v = \lambda_w = 0$) the empirical frequencies $q(x_i=a)$ and $q(x_i=a, x_j=b)$ are equal to the model probabilities at the maximum of the likelihood when the gradient becomes zero.

3.4 The prior on v

Most previous approaches chose a prior around the origin, $p(\mathbf{v}) = \mathcal{N}(\mathbf{v}|\mathbf{0}, \lambda_v \mathbf{I})$. This choice has an obvious draw-back. Taking the sum over $b = 1, \ldots, 20$ at the optimum of the gradient of couplings in eq. (3.31), yields

$$0 = N_{ij}\, q(x_i=a, x_j \neq 0) - N_{ij}\, p(x_i=a|\mathbf{v}, \mathbf{w}) - \lambda_w \sum_{b=1}^{20} w_{ijab}. \tag{3.24}$$

Incidentally, by taking the sum over a it follows that,

$$\sum_{a,b=1}^{20} w_{ijab} = 0. \tag{3.25}$$

At the optimum the gradient with respect to v_{ia} vanishes and $p(x_i = a|\mathbf{v}, \mathbf{w}) = q(x_i = a) - \lambda_v(v_{ia} - v^*_{ia})/N_i$ can be substituted into equation (3.24), yielding

$$0 = N_{ij}\, q(x_i=a, x_j \neq 0) - N_{ij}\, q(x_i = a) + \frac{N_{ij}}{N_i} \lambda_v (v_{ia} - v^*_{ia}) - \lambda_w \sum_{b=1}^{20} w_{ijab}\ , \tag{3.26}$$

for all $i, j \in \{1, \ldots, L\}$ and all $a \in \{1, \ldots, 20\}$. Considering a MSA without gaps, it can be shown how the choice $\mathbf{v}^* = \mathbf{0}$ leads to undesirable results. The first two terms $N_{ij}\, q(x_i=a, x_j \neq 0) - N_{ij}\, q(x_i = a)$ cancel out, leaving

$$0 = \lambda_v (v_{ia} - v^*_{ia}) - \lambda_w \sum_{b=1}^{20} w_{ijab}. \tag{3.27}$$

Consider a column i that is not coupled to any other and assume that amino acid a was frequent in column i and therefore v_{ia} would be large and positive. Then according to eq. (3.27), for any other column j the 20 coefficients w_{ijab} for $b \in \{1, \ldots, 20\}$ would have to take up the bill and deviate from zero!

This unwanted behaviour can be corrected by instead choosing a Gaussian prior centered around $\mathbf{v}^*$ obeying

$$\frac{\exp(v^*_{ia})}{\sum_{a'=1}^{20} \exp(v^*_{ia'})} = q(x_i = a). \tag{3.28}$$

This choice ensures that if no columns are coupled, i.e. $p(\mathbf{x}|\mathbf{v}, \mathbf{w}) = \prod_{i=1}^{L} p(x_i)$, $\mathbf{v} = \mathbf{v}^*$ and $\mathbf{w} = \mathbf{0}$ gives the correct probability model for the sequences in the MSA. Furthermore imposing the restraint $\sum_{a=1}^{20} v_{ia} = 0$ to fix the gauge of the v_{ia} (i.e. to remove the indeterminacy), yields

$$v^*_{ia} = \log q(x_i = a) - \frac{1}{20} \sum_{a'=1}^{20} \log q(x_i = a'). \tag{3.29}$$

For this choice, $v_{ia} - v^*_{ia}$ will be approximately zero and will certainly be much smaller than v_{ia}, hence the sum over coupling coefficients in eq. (3.27) will be close to zero, as it should be.

Another way to understand the choice of $\mathbf{v}^*$ in eq. (3.29) as opposed to $\mathbf{v}^* = \mathbf{0}$ is by noting that in that case $q(x_i = a) \approx p(x_i = a|\mathbf{v}^*, \mathbf{w}^*)$. Therefore, if $q(x_i = a, x_j = b) = q(x_i = a)\, q(x_j = b)$ it follows that $p(x_i = a, x_j = b|\mathbf{v}, \mathbf{w}) \approx q(x_i = a, x_j = b) = p(x_i = a|\mathbf{v}^*, \mathbf{w}^*)\, p(x_j = b|\mathbf{v}^*, \mathbf{w}^*)$, i.e. we would correctly conclude that $w_{ijab} = 0$ and (i, a) and (j, b) are not coupled.

Regarding (3.31): Note that the couplings between columns i and j in the hypothetical MSA presented in the last section 3.2 will now vanish since $N_{ij} = 0$ and the gradient with respect to w_{ijab} is equal to $-\lambda_w w_{ijab}$.

3.5 Optimizing the Full Likelihood with Contrastive Divergence

If the proportion of gap positions in $\mathbf{X}$ is small (e.g. $< 5\%$, also compare percentage of gaps in dataset in Appendix Figure B.2), the sums over $\mathbf{y} \in S_n$ in eqs. (3.19) and (3.23) can be approximated by $p(x_i = a|\mathbf{v}, \mathbf{w})I(x_{ni} \neq 0)$ and $p(x_i = a, x_j = b|\mathbf{v}, \mathbf{w})I(x_{ni} \neq 0, x_{nj} \neq 0)$, respectively, and the partial derivatives become

$$\frac{\partial LL_{\text{reg}}}{\partial v_{ia}} = N_i q(x_i = a) - N_i\, p(x_i = a|\mathbf{v}, \mathbf{w}) - \lambda_v (v_{ia} - v^*_{ia}) \tag{3.30}$$

$$\frac{\partial LL_{\text{reg}}}{\partial w_{ijab}} = N_{ij} q(x_i = a, x_j = b) - N_{ij}\, p(x_i = a, x_j = b|\mathbf{v}, \mathbf{w}) - \lambda_w w_{ijab} \tag{3.31}$$

The gradients consist of the empirical single and pairwise amino acid counts, which are constant and can be computed once from the alignment and the model probability terms that cannot be computed analytically.

MCMC algorithms are typically used in Bayesian statistics to generate samples from probability distributions that involve the computation of complex integrals and therefore cannot be computed analytically. Samples are generated from an distribution as the current state of a running Markov chain. The equilibrium statistics identical to true prob distribution statistics if MCMC is run long enough Lapedes et al. applied such a scheme in 1999 but running the Markov Chain until reaching its stationary distribution is only feasible for small proteins.

Hinton suggests CD as an approximation to MCMC methods. - represents a general framework that is applicable to all kinds of likelihoods evenhtough originally proposed for products of experts models - approach that is applicable to all kinds of ML problems and that makes use of another class of MCMC algorithms: Gibbs sampling. Instead of starting a Markov Chain from a random point, it is initialized from a data sample. It is then evolved for only one Gibbs step and the newly generated samples are used to compute estimate marginal distribution from simple statistics. N Markov chains will be initialized with the N sequences from the MSA and N new samples will be generated by a single step of Gibbs sampling from each of the N sequences. Single and pairwise amino acid counts can be computed from the newly generated sample that correspond to an estimate of the marginal probabilies from the model.

Gibbs sampling works only when conditional probabilites of the target distribution can be exactly computed.
This is the case and the conditionals look like this:

One step of Gibbs sampling samples a new amino acid according to the conditional probability for each position in the sequence (randomly selected). When enough sampling steps are made, samples can be considered independent of one another.

- gradient is really noisy but hinton showed empirically and later it was shown proven that CD gradients becomes zero when full likelihood is at optimum
- The intuition is that one obtains a rough estimate of the gradient even though the sampler has not reached the equilibrium distribution Even though the approximation for the second term is very bad, it can be seen that this approximate gradient will become zero approximately where the true gradient of the likelihood also becomes zero. To see this, imagine $(\mathbf{v}^*, \mathbf{w}^*)$ is the maximum of the likelihood. Then, starting from the sequences in the MSA, the Gibbs sampling step should not lead away from the empirical distribution, because the parameters $(\mathbf{v}^*, \mathbf{w}^*)$ already describe the empirical distribution correctly. This equality of the two maxima is accurate to the extent that the empirical distribution with its finite number of sequences N can represent the true distribution given by parameters $(\mathbf{v}^*, \mathbf{w}^*)$. Therefore, the larger N, the better CD will optimise into the maximum of the true likelihood.
- It can be shown that CD using one step of Gibbs sampling for on only one variable for fully visible Boltzman machines is exactly equivalent to optimising the pseudo likelihood. In general CD can be considered an approximation to pLL [117,118].

- it was shown that CD10 works better than CD 1, but of course more computational complex
- PCD is a variation of CD in that the markov chain is NOT reinitialized at every iteration
- tieleman showed improved convergence properties

For PCD, the Markov chains are not restarted from the N sequences in the MSA every time a new gradient is computed. Instead the Markov chains are evolved between successive gradient computations without resetting them. This ensures that, as we approach the maximum $(\mathbf{v}^*, \mathbf{w}^*)$, we acquire more and more samples from the distribution corresponding to parameters $(\mathbf{v}, \mathbf{w})$ near the optimum. Hence our approximation to the gradient of the likelihood gets better the longer we sample, independent of the number of sequences N in the MSA.

Dr Stefan Seemayer provided a Python implementation of CCMpred that was extended to optimize the full-likelihood of the MRF.

CD is about the difference between the original data set and a perturbed data set perturbed data set : The contrasting data set needs to represent A data sample characteristic of the current PARAMETERS –> Gibbs Sampling starting from data Note: as contrasting dataset towards true_parameters, the elements of the gradient converge to the gradient of the max log likelihood – At the limit of the Markov chain, the CD converges to the actual MLE

4

A Bayesian Statistical Model for Residue-Residue Contact Prediction

All methods so far predict contacts by finding the one solution of parameters v_{ia} and w_{ijab} that maximizes a regularized version of the log likelihood of the MSA and in a second step transforming the MAP estimates of the couplings $\mathbf{w}^*$ into heuristic contact scores (see Introduction 1.4.3.1). Apart from the heuristic transformation that omits meaningful information comprised in the coupling matrices $\mathbf{w}_{ij}$ as discussed in section 2, using the MAP estimate of the parameters instead of the true distribution has the decisive disadvantage of concealing the uncertainty of the estimates.

The next sections present the derivation of a principled Bayesian statistical approach for contact prediction eradicating these deficiencies. The model provides estimates of the probability distributions of the distances r_{ij} between C_β atoms of all residues pairs i and j, given the MSA $\mathbf{X}$. The parameters $(\mathbf{v}, \mathbf{w})$ of the MRF model describing the probability distribution of the sequences in the MSA are treated as hidden parameters that can be integrated out using an approximation to the posterior distribution of couplings $\mathbf{w}$. This approach also allows to explictely model the distance-dependence of coupling coeffcients $\mathbf{w}_{ij}$ as a mixture of Gaussians with distance-dependent mixture weights and thus can even learn correlations between couplings.

4.1 Computing the Posterior Distribution of Distances $p(\mathbf{r}|\mathbf{X})$

The joint probability of distances and MRF model parameters $(\mathbf{v}, \mathbf{w})$ given the MSA $\mathbf{X}$ and a set of sequence derived features ϕ (described in detail in section 5), can be written as a hierarchical Bayesian model of the form:

$$p(\mathbf{r}, \mathbf{v}, \mathbf{w}|\mathbf{X}, \phi) \propto p(\mathbf{X}|\mathbf{v}, \mathbf{w}) p(\mathbf{v}, \mathbf{w}|\mathbf{r})\, p(\mathbf{r}|\phi)\,. \tag{4.1}$$

The ultimate goal is to compute the posterior probability of the distances, $p(\mathbf{r}|\mathbf{X}, \phi)$, that can be obtained by treating the parameters $(\mathbf{v}, \mathbf{w})$ as hidden variables and marginalizing over these parameters,

$$p(\mathbf{r}|\mathbf{X}, \phi) \propto p(\mathbf{X}|\mathbf{r}) p(\mathbf{r}|\phi) \tag{4.2}$$

$$p(\mathbf{X}|\mathbf{r}) = \int\int p(\mathbf{X}|\mathbf{v}, \mathbf{w})\, p(\mathbf{v}, \mathbf{w}|\mathbf{r})\, d\mathbf{v}\, d\mathbf{w}\,. \tag{4.3}$$

The single potentials $\mathbf{v}$ will be fixed at their best estimate $\mathbf{v}^*$ (see section 3.4) by using a very tight prior $p(\mathbf{v}) = \mathcal{N}(\mathbf{v}|\mathbf{v}^*, \lambda_v^{-1}\mathbf{I}) \to \delta(\mathbf{v} - \mathbf{v}*)$ for $\lambda_v \to \infty$ that acts as a delta function. This allows the replacement of the intergral over $\mathbf{v}$ with the value of the integrand at its mode $\mathbf{v}^*$.

Computing the integral over $\mathbf{w}$ can be achieved by factorizing the integrand into factors over (i, j) and performing each integration over the coupling coefficients $\mathbf{w}_{ij}$ for (i, j) separately.

For that account, the prior over $\mathbf{w}$ will be modelled as a product over independent contributions over $\mathbf{w}_{ij}$ with $\mathbf{w}_{ij}$ depending only on the distance r_{ij}, which is described in detail in the next section 4.2. The prior over MRF model parameters then yields,

$$p(\mathbf{v}, \mathbf{w}|\mathbf{r}) = \mathcal{N}(\mathbf{v}|\mathbf{v}^*, \lambda_v^{-1}\mathbf{I}) \prod_{1 \le i < j \le L} p(\mathbf{w}_{ij}|r_{ij})\,. \tag{4.4}$$

Furthermore, section 4.3 proposes an approximation to the regularised likelihood, $p(\mathbf{X}|\mathbf{v}, \mathbf{w})\, p(\mathbf{v}, \mathbf{w})$, with a Gaussian distribution that facilitates the analytical solution of the integral in eq. (4.3) and is covered in section 4.4.

Finally, the marginals $p(r_{ij}|\mathbf{X}, \phi) = \int p(\mathbf{r}|\mathbf{X}, \phi) d\mathbf{r}_{\setminus ij}$, where $\mathbf{r}_{\setminus ij}$ is the vector containing all coordinates of $\mathbf{r}$ except r_{ij} will be computed in 4.5.

4.2 Modelling the prior over couplings with dependence on r_{ij}

The prior over couplings $p(\mathbf{w}_{ij}|r_{ij})$ will be modelled as a mixture of $K+1$ 400-dimensional Gaussians, with means $\mu_k \in \mathbb{R}^{400}$, precision matrices $\mathbf{\Lambda}_k \in \mathbb{R}^{400\times 400}$, and distance-dependent, normalised weights $g_k(r_{ij})$,

$$p(\mathbf{w}_{ij}|r_{ij}) = \sum_{k=0}^{K} g_k(r_{ij})\, \mathcal{N}(\mathbf{w}_{ij}|\mu_k, \mathbf{\Lambda}_k^{-1})\,. \tag{4.5}$$

The mixture weights $g_k(r_{ij})$ in eq. (4.5) are modelled as softmax:

$$g_k(r_{ij}) = \frac{\exp \gamma_k(r_{ij})}{\sum_{k'=0}^{K} \exp \gamma_{k'}(r_{ij})} \tag{4.6}$$

The functions $g_k(r_{ij})$ remain invariant when adding an offset to all $\gamma_k(r_{ij})$. This degeneracy can be removed by setting $\gamma_0(r_{ij}) = 1$.

4.3 Gaussian approximation to the posterior of couplings

From sampling experiments done by Markus Gruber we know that the regularized pseudo-log-likelihood for realistic examples of protein MSAs obeys the equipartition theorem. The equipartition theorem states that in a harmonic potential (where third and higher order derivatives around the energy minimum vanish) the mean potential energy per degree of freedom (i.e. per eigendirection of the Hessian of the potential) is equal to $k_B T/2$, which is of course equal to the mean kinetic energy per degree of freedom. Hence we have a strong indication that in realistic examples the pseudo log likelihood is well approximated by a harmonic potential. We assume here that this will also be true for the regularized log likelihood.

The posterior distribution of couplings $\mathbf{w}$ is given by

$$p(\mathbf{w}|\mathbf{X}, \mathbf{v}^*) = p(\mathbf{X}|\mathbf{v}^*, \mathbf{w}) \mathcal{N}(\mathbf{w}|\mathbf{0}, \lambda_w^{-1}\mathbf{I}) \tag{4.7}$$

where the single potentials $\mathbf{v}$ are set to the target vector $\mathbf{v}^*$ as discussed in section 4.1.

The posterior distribution can be approximated with a so called "Laplace Approximation"[75] as follows. By performing a second order Taylor expansion around the mode $\mathbf{w}^*$ of the log posterior it can be written as

$$\begin{aligned} \log p(\mathbf{w}|\mathbf{X}, \mathbf{v}^*) \overset{!}{\approx}\ & \log p(\mathbf{w}^*|\mathbf{X}, \mathbf{v}^*) && (4.8)\\ & + \nabla_{\mathbf{w}} \log p(\mathbf{w}|\mathbf{X}, \mathbf{v}^*)|_{\mathbf{w}^*}(\mathbf{w} - \mathbf{w}^*) && (4.9)\\ & - \frac{1}{2}(\mathbf{w} - \mathbf{w}^*)^{\mathrm{T}} \mathbf{H} (\mathbf{w} - \mathbf{w}^*) \ . && (4.10) \end{aligned}$$

where $\mathbf{H}$ signifies the *negative* Hessian matrix with respect to the components of $\mathbf{w}$,

$$(\mathbf{H})_{klcd,ijab} = - \left. \frac{\partial^2 \log p(\mathbf{w}|\mathbf{X}, \mathbf{v}^*)}{\partial \mathbf{w}_{klcd} \, \partial w_{ijab}} \right|_{(\mathbf{w}^*)} . \tag{4.11}$$

The mode $\mathbf{w}^*$ will be determined with the CD approach described in detail in section 3. Since the gradient vanishes at the mode maximum, $\nabla_{\mathbf{w}} \log p(\mathbf{w}|\mathbf{X}, \mathbf{v}^*)|_{\mathbf{w}^*} = 0$, the second order approximation can be written as

$$\log p(\mathbf{w}|\mathbf{X},\mathbf{v}^*) \approx \log p(\mathbf{w}^*|\mathbf{X},\mathbf{v}^*) - \frac{1}{2}(\mathbf{w}-\mathbf{w}^*)^{\mathrm{T}}\,\mathbf{H}\,(\mathbf{w}-\mathbf{w}^*)\,. \tag{4.12}$$

Hence, the posterior of couplings can be approximated with a Gaussian

$$\begin{aligned} p(\mathbf{w}|\mathbf{X},\mathbf{v}^*) &\approx p(\mathbf{w}^*|\mathbf{X},\mathbf{v}^*)\exp\left(-\frac{1}{2}(\mathbf{w}-\mathbf{w}^*)^{\mathrm{T}}\mathbf{H}(\mathbf{w}-\mathbf{w}^*)\right) \\ &= p(\mathbf{w}^*|\mathbf{X},\mathbf{v}^*)\frac{(2\pi)^{\frac{D}{2}}}{|\mathbf{H}|^{\frac{D}{2}}} \times \mathcal{N}(\mathbf{w}|\mathbf{w}^*,\mathbf{H}^{-1}) \qquad (4.13) \\ &\propto \mathcal{N}(\mathbf{w}|\mathbf{w}^*,\mathbf{H}^{-1})\,, \qquad (4.14) \end{aligned}$$

with proportionality constant that depends only on the data and with a precision matrix equal to the negative Hessian matrix. The surprisingly easy computation of the Hessian can be found in Methods section 6.5.1.

4.3.1 Iterative improvement of Laplace approximation

The quality of the Gaussian approximation to the posterior distribution of couplings $p(\mathbf{w}|\mathbf{X},\mathbf{v}^*)$ depends on two points,

1. how well is the posterior distribution of couplings approximated by a Gaussian
2. how closely does the mode of the posterior distribution of couplings lie near the mode of the integrand in equation **??**.

The second point can be addressed quite effectively in the following way.

(see Murphy page 658 eq. 18.137 and eq 18.138)

Supppose the optimal prior parameters $(\tilde{\mu_k}, \tilde{\mathbf{\Lambda}_k})$ have been trained as described in Methods section 6.5.3, using the standard isotropic regularisation prior $\mathcal{N}(\mathbf{w}_{ij}|\mathbf{0},\lambda_w^{-1}\mathbf{I})$. An improved regularisation prior $\mathcal{N}(\mathbf{w}_{ij}|\mu(r_{ij}),\mathbf{\Sigma}(r_{ij}))$ can then be selected using the knowledge of the true, optimised prior, by matching the mean and variance of the improved regularisation with those of the true prior from the first optimisation:

$$\begin{aligned} \mu(r_{ij}) &= \mathrm{E}_{p(\mathbf{w}_{ij}|r_{ij},\tilde{\mu},\tilde{\Lambda})}\left[\mathbf{w}_{ij}\right] \qquad (4.15) \\ &= \int \mathbf{w}_{ij}\, p(\mathbf{w}_{ij}|r_{ij},\tilde{\mu},\tilde{\Lambda}) d\mathbf{w} \qquad (4.16) \\ &= \int \mathbf{w}_{ij} \sum_{k=0}^{K} g_k(r_{ij})\,\mathcal{N}(\mathbf{w}_{ij}|\tilde{\mu_k},\tilde{\Lambda}_k^{-1}) d\mathbf{w} \qquad (4.17) \\ &= \sum_{k=0}^{K} g_k(r_{ij}) \int \mathbf{w}_{ij}\,\mathcal{N}(\mathbf{w}_{ij}|\tilde{\mu_k},\tilde{\Lambda}_k^{-1}) d\mathbf{w} \qquad (4.18) \\ \mu(r_{ij}) &= \sum_{k=0}^{K} g_k(r_{ij})\,\tilde{\mu_k} \qquad (4.19) \end{aligned}$$

and similarly,

$$\mathbf{\Sigma}(r_{ij}) = \mathrm{var}_{p(\mathbf{w}_{ij}|r_{ij},\tilde{\mu},\tilde{\Lambda})}\left[\mathbf{w}_{ij}\right] \tag{4.20}$$

$$= \int (\mathbf{w}_{ij} - \mu(r_{ij}))(\mathbf{w}_{ij} - \mu(r_{ij}))^{\mathrm{T}}\, p(\mathbf{w}_{ij}|r_{ij},\tilde{\mu},\tilde{\Lambda}) d\mathbf{w} \tag{4.21}$$

$$= \sum_{k=0}^{K} g_k(r_{ij}) \int (\mathbf{w}_{ij} - \mu(r_{ij}))(\mathbf{w}_{ij} - \mu(r_{ij}))^{\mathrm{T}}\, \mathcal{N}(\mathbf{w}_{ij}|\tilde{\mu_k}, \tilde{\mathbf{\Lambda}_k}^{-1}) d\mathbf{w} \tag{4.22}$$

$$= \sum_{k=0}^{K} g_k(r_{ij}) \int (\mathbf{w}_{ij} - \mu(r_{ij}) + \tilde{\mu_k})(\mathbf{w}_{ij} - \mu(r_{ij}) + \tilde{\mu_k})^{\mathrm{T}}\, \mathcal{N}(\mathbf{w}_{ij}|\mathbf{0}, \tilde{\mathbf{\Lambda}_k}^{-1}) d\mathbf{w} \tag{4.23}$$

$$\mathbf{\Sigma}(r_{ij}) = \sum_{k=0}^{K} g_k(r_{ij}) \left(\tilde{\mathbf{\Lambda}_k}^{-1} + (\mu(r_{ij}) - \tilde{\mu_k})(\mu(r_{ij}) - \tilde{\mu_k})^{\mathrm{T}}\right) . \tag{4.24}$$

We can now run a second optimisation with better regularisation prior, in which the $\tilde{\mu}$ and $\tilde{\Lambda}$ are fixed and will not be optimised. Instead we optimise the marginal likelihood as a function of μ_k and $\mathbf{\Lambda}_k$. Since the new regularisation prior will be very close to the mode of the integrand in the marginal likelihood, our approximation for the second iteration has improved in comparison to the first iteration. In principle, a third iteration can be done in which our regularisation prior derived from the prior that was found by optimisation in the second iteration. However this is unlikely to further improve the predictions.

4.4 Computing the likelihood function of distances $p(\mathbf{X}|\mathbf{r})$

In order to compute the likelihood function of the distances, one needs to solve the integral over $(\mathbf{v}, \mathbf{w})$,

$$p(\mathbf{X}|\mathbf{r}) = \int\int p(\mathbf{X}|\mathbf{v},\mathbf{w})\, p(\mathbf{v},\mathbf{w}|\mathbf{r})\, d\mathbf{v}\, d\mathbf{w} . \tag{4.25}$$

Inserting the prior over parameters $p(\mathbf{v},\mathbf{w}|\mathbf{r})$ from eq. (4.4) into the previous equation and performing the integral over $\mathbf{v}$, as discussed earlier in section 4.1, yields

$$p(\mathbf{X}|\mathbf{r}) = \int \left(\int p(\mathbf{X}|\mathbf{v},\mathbf{w})\, \mathcal{N}(\mathbf{v}|\mathbf{v}^*, \lambda_v^{-1}\mathbf{I})\, d\mathbf{v}\right) \prod_{1\le i<j\le L} p(\mathbf{w}_{ij}|r_{ij})\, d\mathbf{w} \tag{4.26}$$

$$p(\mathbf{X}|\mathbf{r}) = \int p(\mathbf{X}|\mathbf{v}^*,\mathbf{w}) \prod_{1\le i<j\le L} p(\mathbf{w}_{ij}|r_{ij})\, d\mathbf{w} \tag{4.27}$$

Next, the likelihood will be multiplied with the regularisation prior and the distance-dependent prior will be divided by the regularisation prior again:

$$p(\mathbf{X}|\mathbf{r}) \;=\; \int p(\mathbf{X}|\mathbf{v}^*,\mathbf{w})\,\mathcal{N}(\mathbf{w}|\mathbf{0},\lambda_w^{-1}\mathbf{I}) \prod_{1\le i<j\le L} \frac{p(\mathbf{w}_{ij}|r_{ij})}{\mathcal{N}(\mathbf{w}_{ij}|\mathbf{0},\lambda_w^{-1}\mathbf{I})}\,d\mathbf{w}\,. \quad (4.28)$$

Now the crucial advantage of our likelihood regularisation is borne out: We can chose the strength of the regularisation prior, λ_w, such that the mode $\mathbf{w}^*$ of the regularised likelihood is near to the mode of the integrand in the last integral. The regularisation prior $\mathcal{N}(\mathbf{w}_{ij}|\mathbf{0},\lambda_w^{-1}\mathbf{I})$ is then a simpler, approximate version of the real, distance-dependent prior $\prod_{1\le i<j\le L} p(\mathbf{w}_{ij}|r_{ij})$. This allows us to approximate the regularised likelihood with a Gaussian distribution (eq. (4.14)), because this approximation will be fairly accurate in the region around its mode, which is near the region around the mode of the integrand and this again is in the region that contributes most to the integral:

$$p(\mathbf{X}|\mathbf{r}) \;\propto\; \int \mathcal{N}(\mathbf{w}|\mathbf{w}^*,\mathbf{H}^{-1}) \prod_{1\le i<j\le L} \frac{p(\mathbf{w}_{ij}|r_{ij})}{\mathcal{N}(\mathbf{w}_{ij}|\mathbf{0},\lambda_w^{-1}\mathbf{I})}d\mathbf{w}\,. \quad (4.29)$$

The matrix $\mathbf{H}$ has dimensions $(L^2\times 20^2)\times(L^2\times 20^2)$. Computing it is obviously infeasible, even if there was a way to compute $p(x_i=a,x_j=b|\mathbf{v}^*,\mathbf{w}^*)$ efficiently. In Methods section **??** is shown that in practice, the off-diagonal block matrices with $(i,j)\neq(k,l)$ are negligible in comparison to the diagonal block matrices. For the purpose of computing the integral in eq. (4.29), it is therefore a good approximation to simply set the off-diagonal block matrices (case 3 in (6.26)) to zero!

The first term in the integrand of eq. (4.29) now factorizes over (i,j),

$$\mathcal{N}(\mathbf{w}|\mathbf{w}^*,\mathbf{H}^{-1}) \approx \prod_{1\le i<j\le L} \mathcal{N}(\mathbf{w}_{ij}|\mathbf{w}_{ij}^*,\mathbf{H}_{ij}^{-1}), \quad (4.30)$$

with the diagonal block matrices are $(\mathbf{H}_{ij})_{ab,cd} := (\mathbf{H})_{ijab,ijcd}$.

Now the product over all residue indices can be moved in front of the integral and each integral can be performed over $\mathbf{w}_{ij}$ separately,

$$p(\mathbf{X}|\mathbf{r}) \;\propto\; \int \prod_{1\le i<j\le L} \mathcal{N}(\mathbf{w}_{ij}|\mathbf{w}_{ij}^*,\mathbf{H}_{ij}^{-1}) \prod_{1\le i<j\le L} \frac{p(\mathbf{w}_{ij}|r_{ij})}{\mathcal{N}(\mathbf{w}_{ij}|\mathbf{0},\lambda_w^{-1}\mathbf{I})}d\mathbf{w} \quad (4.31)$$

$$p(\mathbf{X}|\mathbf{r}) \;\propto\; \int \prod_{1\le i<j\le L} \left(\mathcal{N}(\mathbf{w}_{ij}|\mathbf{w}_{ij}^*,\mathbf{H}_{ij}^{-1}) \frac{p(\mathbf{w}_{ij}|r_{ij})}{\mathcal{N}(\mathbf{w}_{ij}|\mathbf{0},\lambda_w^{-1}\mathbf{I})}\right) d\mathbf{w} \quad (4.32)$$

$$p(\mathbf{X}|\mathbf{r}) \;\propto\; \prod_{1\le i<j\le L} \int \mathcal{N}(\mathbf{w}_{ij}|\mathbf{w}_{ij}^*,\mathbf{H}_{ij}^{-1}) \frac{p(\mathbf{w}_{ij}|r_{ij})}{\mathcal{N}(\mathbf{w}_{ij}|\mathbf{0},\lambda_w^{-1}\mathbf{I})}d\mathbf{w}_{ij} \quad (4.33)$$

Inserting the distance-dependent coupling prior defined in eq. (4.5) yields

$$p(\mathbf{X}|\mathbf{r}) \propto \prod_{1\le i<j\le L} \int \mathcal{N}(\mathbf{w}_{ij}|\mathbf{w}^*_{ij}, \mathbf{H}^{-1}_{ij}) \frac{\sum_{k=0}^{K} g_k(r_{ij}) \mathcal{N}(\mathbf{w}_{ij}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{w}_{ij}|\mathbf{0}, \lambda_w^{-1}\mathbf{I})} d\mathbf{w}_{ij} \quad (4.34)$$

$$p(\mathbf{X}|\mathbf{r}) \propto \prod_{1\le i<j\le L} \sum_{k=0}^{K} g_k(r_{ij}) \int \frac{\mathcal{N}(\mathbf{w}_{ij}|\mathbf{w}^*_{ij}, \mathbf{H}^{-1}_{ij})}{\mathcal{N}(\mathbf{w}_{ij}|\mathbf{0}, \lambda_w^{-1}\mathbf{I})} \mathcal{N}(\mathbf{w}_{ij}|\mu_k, \mathbf{\Lambda}_k^{-1}) d\mathbf{w}_{ij} \quad (4.35)$$

The integral can be carried out using the following formula:

$$\int d\mathbf{x} \frac{\mathcal{N}(\mathbf{x}|\mu_1, \mathbf{\Lambda}_1^{-1})}{\mathcal{N}(\mathbf{x}|\mathbf{0}, \mathbf{\Lambda}3^{-1})} \mathcal{N}(\mathbf{x}|\mu_2, \mathbf{\Lambda}_2^{-1}) = \frac{\mathcal{N}(\mathbf{0}|\mu_1, \mathbf{\Lambda}_1^{-1})\mathcal{N}(\mathbf{0}|\mu_2, \mathbf{\Lambda}_2^{-1})}{\mathcal{N}(\mathbf{0}|\mathbf{0}, \mathbf{\Lambda}_3^{-1})\mathcal{N}(\mathbf{0}|\mu_{12}, \mathbf{\Lambda}_{123}^{-1})} \quad (4.36)$$

with

$$\mathbf{\Lambda}_{123} := \mathbf{\Lambda}_1 - \mathbf{\Lambda}_3 + \mathbf{\Lambda}_2 \quad (4.37)$$

$$\mu_{12} := \mathbf{\Lambda}_{123}^{-1}(\mathbf{\Lambda}_1\mu_1 + \mathbf{\Lambda}_2\mu_2). \quad (4.38)$$

We define

$$\mathbf{\Lambda}_{ij,k} := \mathbf{H}_{ij} - \lambda_w\mathbf{I} + \mathbf{\Lambda}_k \quad (4.39)$$

$$\mu_{ij,k} := \mathbf{\Lambda}_{ij,k}^{-1}(\mathbf{H}_{ij}\mathbf{w}^*_{ij} + \mathbf{\Lambda}_k\mu_k)\,. \quad (4.40)$$

and obtain

$$p(\mathbf{X}|\mathbf{r}) \propto \prod_{1\le i<j\le L} \sum_{k=0}^{K} g_k(r_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})}\,. \quad (4.41)$$

$\mathcal{N}(\mathbf{0}|\mathbf{0}, \lambda_w^{-1}\mathbf{I})$ and $\mathcal{N}(\mathbf{0}|\mathbf{w}^*_{ij}, \mathbf{H}^{-1}_{ij})$ are constants that depend only on $\mathbf{X}$ and λ_w and can be omitted.

4.5 The posterior probability distribution for r_{ij}

The posterior distribution for r_{ij} can be computed by marginalizing over all other distances, which are summarized in the vector $\mathbf{r}_{\setminus ij}$:

$$p(r_{ij}|\mathbf{X}, \phi) = \int d\mathbf{r}_{\setminus ij}\, p(\mathbf{r}|\mathbf{X}, \phi) \quad (4.42)$$

$$\propto \int d\mathbf{r}_{\setminus ij}\, p(\mathbf{X}|\mathbf{r})\, p(\mathbf{r}|\phi) \quad (4.43)$$

$$\propto \int d\mathbf{r}_{\setminus ij} \prod_{i'<j'} \sum_{k=0}^{K} g_k(r_{i'j'}) \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} \prod_{i'<j'} p(r_{i'j'}|\phi_{i'j'}) \quad (4.44)$$

and, by pulling out of the integral over $\mathbf{r}_{\setminus ij}$ the term depending only on r_{ij},

$$p(r_{ij}|\mathbf{X},\phi) \propto p(r_{ij}|\phi_{ij}) \sum_{k=0}^{K} g_k(r_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu_k,\mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k},\mathbf{\Lambda}_{ij,k}^{-1})} \tag{4.45}$$

$$\times \prod_{i'<j',(i',j')\neq(i,j)} \int dr_{i'j'}\, p(r_{i'j'}|\phi_{i'j'}) \sum_{k=0}^{K} g_k(r_{i'j'}) \frac{\mathcal{N}(\mathbf{0}|\mu_k,\mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k},\mathbf{\Lambda}_{ij,k}^{-1})} \tag{4.46}$$

Since the second factor involving the integrals over $r_{i'j'}$ is a constant with respect to r_{ij}, we find

$$p(r_{ij}|\mathbf{X},\phi) \propto p(r_{ij}|\phi_{ij}) \sum_{k=0}^{K} g_k(r_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu_k,\mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k},\mathbf{\Lambda}_{ij,k}^{-1})}\,. \tag{4.47}$$

5

Contact Prior

The wealth of successful meta-predictors presented in section 1.3.3 highlights the importance to exploit other sources of information apart from coevolution statistics. Much information about residue interactions is typically contained in features of 1D properties at positions i and j predicted from local sequence profiles, such as secondary structure, solvent accessibility or contact number, and in features of predicted 2D properties such as the contact prediction scores for (i, j) from a profile-based method.

For example, predictions of secondary structure elements and solvent accessibility are used by almost all modern machine learning predictors, such as MetaPsicov [67], NeBCon [70], EPSILON-CP [69], PconsC3 [65]. Other frequently used sequence derived features include pairwise contact potentials, sequence separation and conservation measures such as column entropy [67,70,119].

In the following sections I present a random forest classifier that uses sequence derived features to distinguish contacts from non-contacts. Methods section 6.7.1 lists all features used to train the classifier including the aforementioned standard features as well as some novel features.

The probabilistic predictions of the random forest model can be introduced directly as prior information into the Bayesian statistical model presented in the last section **??** to improve the overall prediction accuracy in terms of posterior probabilities. Furthermore, contact scores from coevolution methods can be added as an additional feature to the random forest model in order to elucidate how much the combined information improves prediction accuracy over the single methods.

5.1 Random Forest Classifiers

Random Forests are supervised machine learning methods that belong to the class of ensemble methods [120–122]. They are easy to implement, fast to train and can handle large numbers of features due to implicit feature selection [123].

Ensemble methods combine the predictions of several independent base estimators with the goal to improve generalizability over a single estimator. Random forests are ensembles of decision trees where randomness is introduced in two ways:

1. every tree is build on a random sample that is of the same size but drawn with replacement from the training set (i.e., a bootstrap sample)
2. every split of a node is performed on a random subset of features

Predictions from all trees are averaged to obtain the final prediction.

A single decision tree, especially when it is grown very deep is highly susceptible to noise in the training set and therefore prone to overfitting which results in poor generalization ability. As a consequence of randomness and averaging over many decision trees, the variance of a random forest predictor decreases and therefore the risk of overfitting.

Random forests are capable of regression and classification tasks. For classification, predictions for new data are obtained by running a data sample down every tree in the forest and then either apply majority voting over single class votes or averaging the probabilistic class predictions. Probabilistic class predictions of single trees are computed as the fraction of training set samples of the same class in a leaf whereas the single class vote refers to the majority class in a leaf.

Typically, *Gini impurity*, which is a computationally efficient approximation to the entropy, is used as a split criterion to estimate the quality of a split of a node. It measures the degree of purity in a data set regarding class labels as $GI = (1 - \sum_{k=1}^{K} p_k^2)$, where p_k is the proportion of class k in the data set. The feature f with the highest *decrease in Gini impurity* ΔGI_f over the two resulting child node subsets will be used to split the data set at the given node N,

$$\Delta GI_f(N_{\text{parent}}) = GI_f(N_{\text{parent}}) - p_{\text{left}} GI(N_{\text{left}}) - p_{\text{right}} GI(N_{\text{left}})$$

where p_{left} and p_{right} refers to the fraction of samples ending up in the left and right child node respectively [123].

Summing the *decrease in Gini impurity* for a feature f over all trees whenever f was used for a split yields the *Gini importance* measure, which can be used as an estimate of general feature relevance. Random forests therefore are popular methods for feature selection and it is common practice to remove the least important features from a data set to reduce the complexity of the model. However, feature importance measured with respect to *Gini importance* needs to be interpreted with care. The random forest model cannot distinguish between correlated features and it will choose any of the correlated features for a split, thereby reducing the importance of the other features and introducing bias. Furthermore, it has been found that feature selection based on *Gini importance* is biased towards selecting features with more categories as they will be chosen more often for splits and therefore tend to obtain higher scores [124].

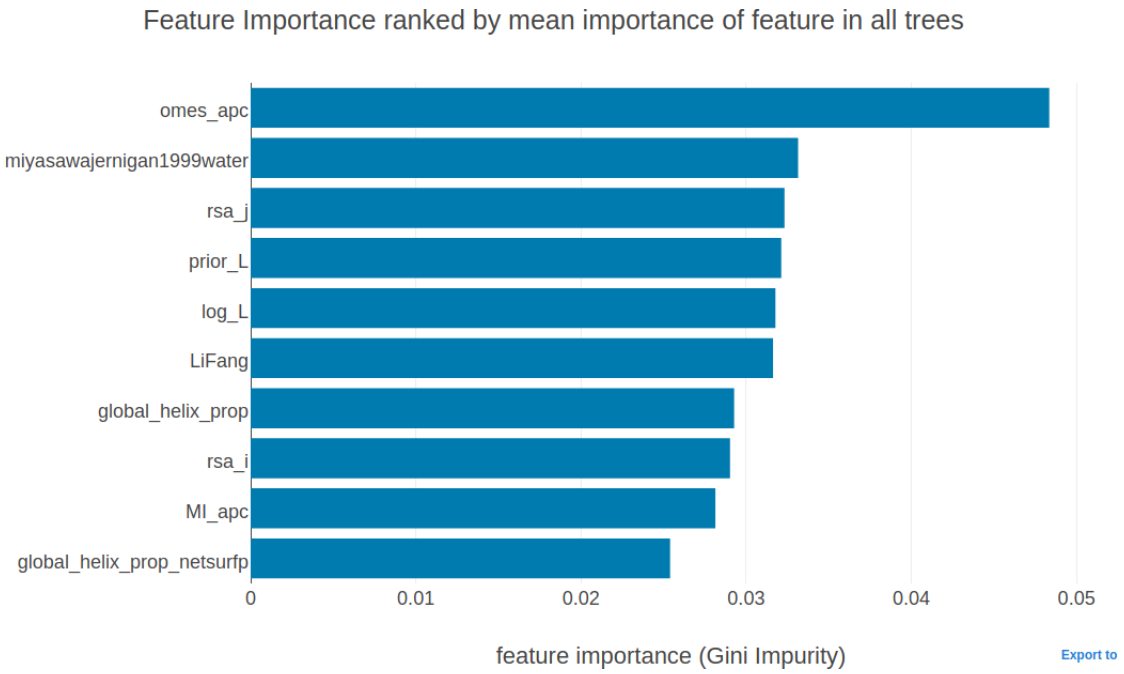


Figure 5.1: Top ten features ranked according to *Gini importance*. omes_apc = OMES contact score with APC, MI_apc = mutual information score with APC, prior_L = contact prior based on protein length as described in methods section 6.7.1.1, log_L = logarithm of protein length, miyasawajernigan1999water = mean pairwise contact potential based on quasi-chemical potential by Miyazawa & Jernigan (1999) [125], rsa_i amd rsa_j = solvent accessibility prediction for residues i and j with NetsurfP [126], LiFang = mean pairwise contact potential as used by Li & Fang [52].

5.2 Evaluating Random Forest Model as Contact Predictor

I trained a random forest classifier on the feature set described in methods section 6.7.1 and optimized model hyperparameters as well as some data set specific settings (e.g window size and class ratios) with 5-fold cross-validation as described in methods section 6.7.2.

Ranking features by *Gini importance* reveals that both local statistical contact scores, OMES and mutual information (see **??**local-methods)), constitute the most important features (see Figure 5.1). Further important features include the log protein length and the contact prior based on protein length. Solvent accessibility predictions also rank highly as well as some of the mean potential scores.

Many features receive only very low *Gini importance* scores. In order to reduce the complexity of the model that comprises 225 features, it is convenient to evaluate model performance on subsets of highly important features only (see methods section 6.7.3). As can be seen in Figure 5.2, most models trained on subsets of the feature space perform nearly identical to the model trained on the complete feature set. When using only the 26 most relevant features, performance of the model drops compared to using at least 75 features.

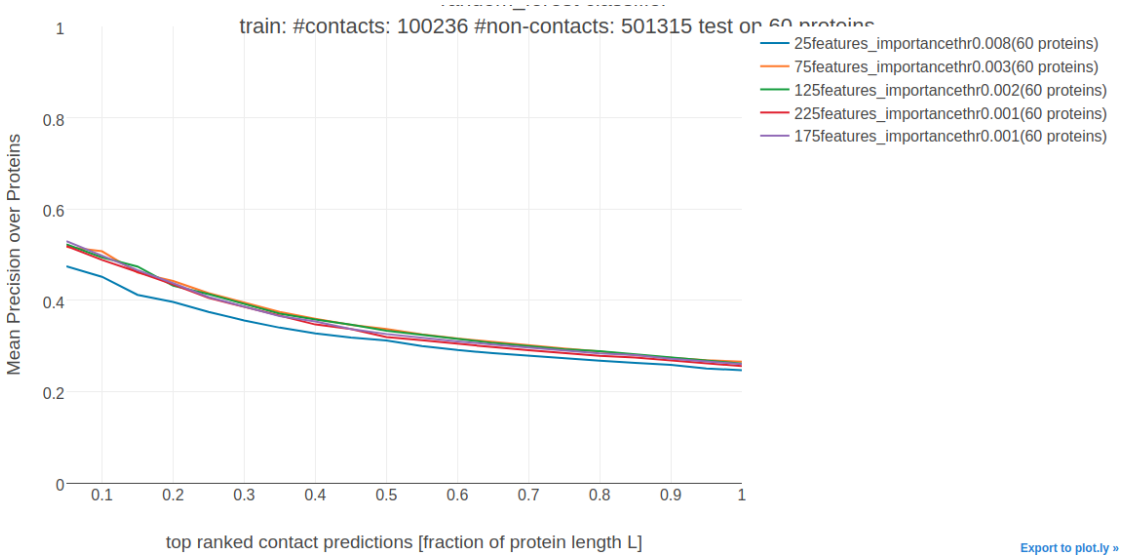


Figure 5.2: Average precision of random forest models trained on subsets of sequence derived features. Subsets of features have been selected as described in methods section 6.7.3.

The model using the 75 most important features according to *Gini importance* has been selected for further analysis. Figure 6.7.4 compares performance of various models. The random forest model trained on the 75 most important features (green line) has a mean precision of 0.33 for the top 5L contacts compared to a precision of 0.47 for the conventional L2norm + APC contact score based on pseudo-likelihood optimization (red line). The performance of the two local statistical contact scores, OMES+APC and MI+APC, which also constitute the two most important features of the random forest model, are shown as blue and yellow lines in Figure 6.7.4. The random forest model improves approximately ten percentage points in precision over both methods.

When analysing performance with respect to alignment size it is clear that the random forest model outperforms the pseudo-likelihood score for small alignments. Both, OMES and MI also perform weak on small alignments, leading to the conclusion that the remaining sequence derived features are highly relevant when the alignment contains only few sequences. This finding is expected, as it is well known that models trained on simple sequence features perform almost independent of alignment size [69].

Mean precision for top ranked contact predictions over 286 proteins splitted into four equally sized subsets according to Neff. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. Subsets are defined according to quantiles of Neff values. Upper left: Subset of proteins with Neff $<$ Q1. Upper right: Subset of proteins with Q1 $<=$ Neff $<$ Q2. Lower left: Subset of proteins with Q2 $<=$ Neff $<$ Q3. Lower right: Subset of proteins with Q3 $<=$ Neff $<$ Q4. Methods are the same as in Figure 6.3.

In order to evaluate how much the sequence derived features can improve contact prediction over the coevolutionary contact scores, the coevolutionary contact score can simply be included as an additional feature into the Random Forest model.

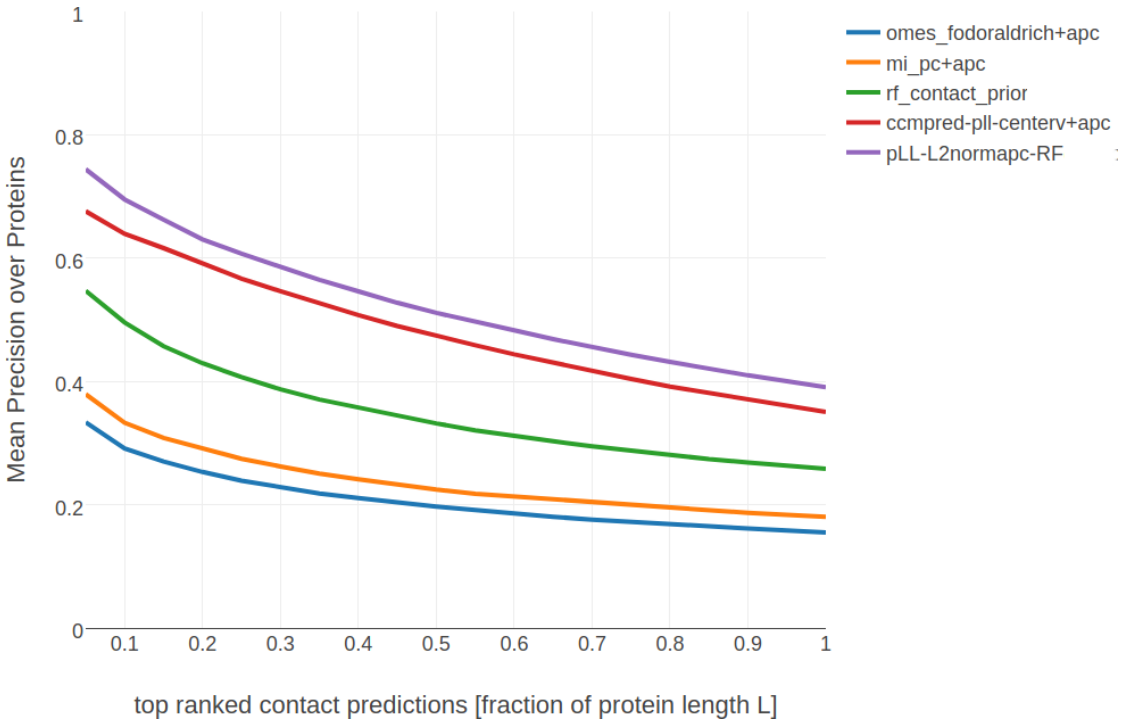


Figure 5.3: Mean precision for top ranked contacts on a test set of 761 proteins. **omes_fodoraldrich+apc** = OMES score with APC as described in section 6.7.1.3. **mi_pc + APC** = mutual information with APC as described in section 6.7.1.3. **rf_contact_prior** = random forest model using only sequence derived features. **pLL-L2normapc-RF** = random forest model using sequence derived features and pseudo-likelihood contact score (L2norm + APC). **ccmpred-pll-centerv+apc** = conventional pseudo-likelihood contact score (L2norm + APC)

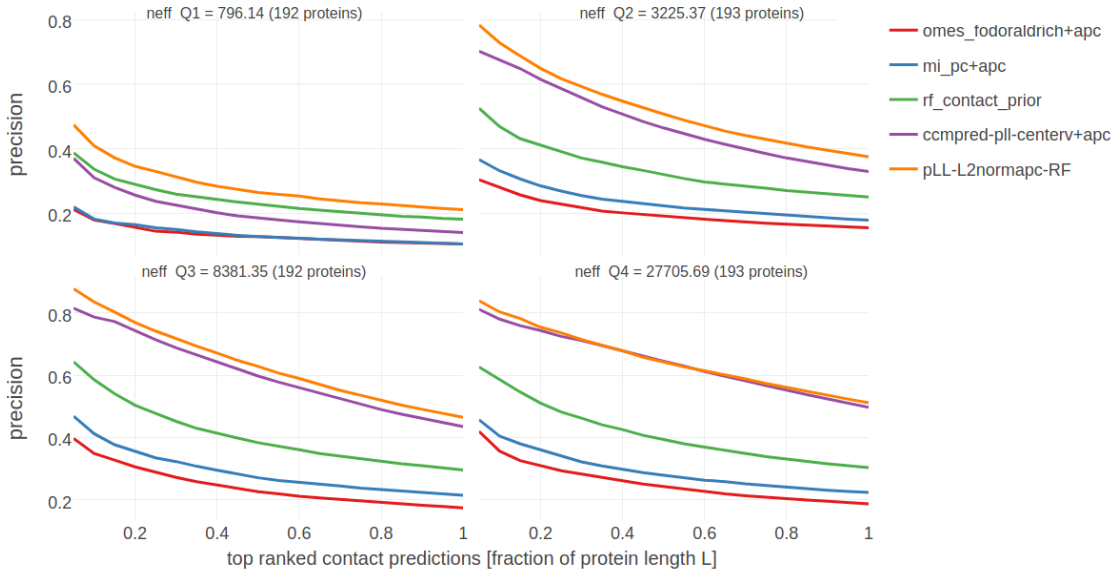


Figure 5.4: Mean precision for top ranked contacts on a test set of 761 proteins splitted into four equally sized subsets according to Neff. Subsets are defined according to quantiles of Neff values. Upper left: Subset of proteins with Neff < Q1. Upper right: Subset of proteins with Q1 <= Neff < Q2. Lower left: Subset of proteins with Q2 <= Neff < Q3. Lower right: Subset of proteins with Q3 <= Neff < Q4. **omes_fodoraldrich+apc** = OMES score with APC as described in section 6.7.1.3. **mi_pc + APC** = mutual information with APC as described in section 6.7.1.3. **rf_contact_prior** = random forest model using only sequence derived features. **pLL-L2normapc-RF** = random forest model using sequence derived features and pseudo-likelihood contact score (APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$). **ccmpred-pll-centerv+apc** = APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$ computed with pseudo-likelihood.

The model was trained as described in methods section 6.7.4 using the additional pseudo-likelihood score feature. As expected, the pseudo-likelihood score comprises the most important feature in the model as can be seen in Figure 6.16. As can be seen in Figure 6.17, using only the 26 most important features improves the predictive power of the model even further.

Finally, the comparison of the extended random forest model ("pLL-L2normapc-RF") to the pseudo-likelihood score in Figure 5.3 reveals that the additional coevolutionary feature indeed improves performance over the score without prior information. Especially for small alignments, the improvment is substantial as can be seen in Figure 5.4. In contrast, the improvement on large alignments is small, as the gain from simple sequence features compared to the much more powerful coevolution signals is neglectable.

6 Methods

all you need to know

6.1 Dataset

A protein dataset has been constructed from the CATH (v4.1) [127] database for classification of protein domains. All CATH domains from classes 1(mainly α), 2(mainly β), 3($\alpha + \beta$) have been selected and filtered for internal redundancy at the sequence level using the `pdbfilter` script from the HH-suite[100] with an E-value cutoff=0.1. The dataset has been split into ten subsets aiming at the best possible balance between CATH classes 1,2,3 in the subsets. All domains from a given CATH topology (=fold) go into the same subsets, so that any two subsets are non-redundant at the fold level. Some overrepresented folds (e.g. Rossman Fold) have been subsampled ensuring that in every subset each class contains at max 50% domains of the same fold. Consequently, a fold is not allowed to dominate a subset or even a class in a subset. In total there are 6741 domains in the dataset.

Multiple sequence alignments were built from the CATH domain sequences (COMBS) using HHblits [100] with parameters to maximize the detection of homologous sequences:

```
hhblits -maxfilt 100000 -realign_max 100000 -B 100000 -Z 100000 -n 5
-e 0.1 -all hhfilter -id 90 -neff 15 -qsc -30
```

The COMBS sequences are derived from the SEQRES records of the PDB file and sometimes contain extra residues that are not resolved in the structure. Therefore, residues in PDB files have been renumbered to match the COMBS sequences. The process of renumbering residues in PDB files yielded ambigious solutions for 293 proteins, that were removed from the dataset. Another filtering step was applied to remove 80 proteins that do not hold the following properties:

- more than 10 sequences in the multiple sequence alignment ($N > 10$)
- protein length between 30 and 600 residues ($30 \leq L \leq 600$)

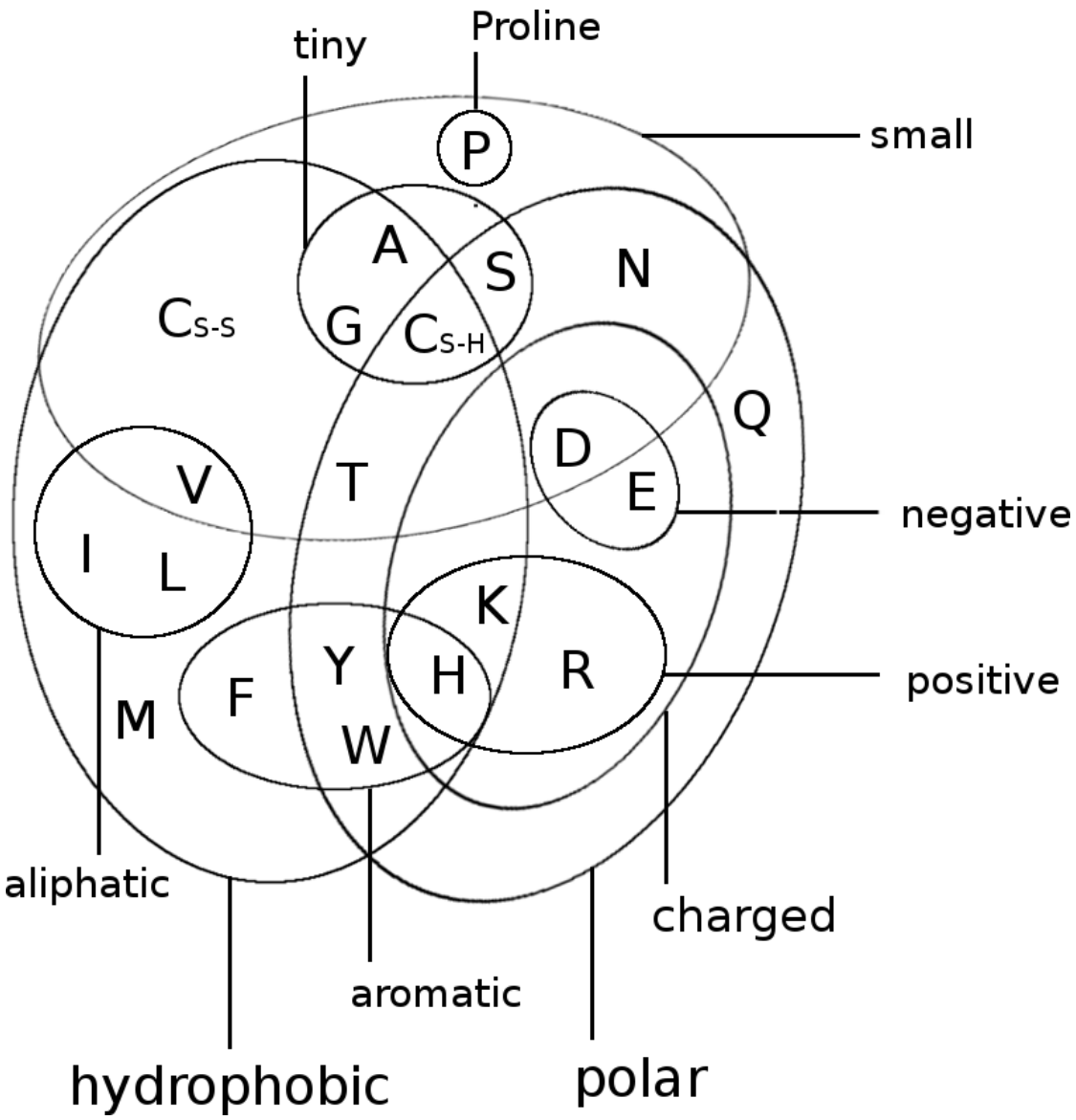


Figure 6.1: Distribution of CATH classes (1=mainly α, 2=mainly β, 3=$\alpha - \beta$) in the dataset and the ten subsets.

- less than 80% gaps in the multiple sequence alignment (percent gaps < 0.8)
- at least one residue-pair in contact at $C_\beta < 8\mathbf{r}A$ and minimum sequence separation of 6 positions

The final dataset is comprised of **6368** proteins with almost evenly distributed CATH classes over the ten subsets (Figure 6.1).

6.2 Optimizing Pseudo-Likelihood

Dr Stefan Seemayer has reimplementated the open-source software CCMpred [82] in Python. Based on a fork of his private github repository I continued development and extended the software, which is now called CCMpredPy. It will soon be

available at https://github.com/soedinglab/CCMpredPy. All computations in this thesis are performed with CCMpredPy unless stated otherwise.

6.2.1 Pseudo-Likelihood Objective Function and its Gradients

CCMpred optimizes the regularized negative pseudo-log-likelihood using conjugate gradients optimizer.

The negative pseudo-log-likelihood, abbreviated $\setminus\!\sqrt{\updownarrow\updownarrow}$, is defined as:

$$\setminus\!\sqrt{\updownarrow\updownarrow}(\mathbf{X}|\mathbf{v},\mathbf{w}) = -\sum_{n=1}^{N}\sum_{i=1}^{L}\left(v_i(x_i^{(n)}) + \sum_{\substack{j=1\\j\neq i}}^{L} w_{ij}(x_i^{(n)}, x_j^{(n)}) - \log Z_i^{(n)}\right) \tag{6.1}$$

The normalization term Z_i sums over all assignments to one position i in sequence:

$$Z_i^{(n)} = \sum_{a=1}^{20}\exp\left(v_i(a) + \sum_{\substack{j=1\\j\neq i}}^{L} w_{ij}(a, x_j^{(n)})\right) \tag{6.2}$$

6.2.2 Differences between CCMpred and CCMpredpy

CCMpredPy differs from CCMpred [82] which is available at https://github.com/soedinglab/CCMpred in several details:

- Initialization of potentials $\mathbf{v}$ and $\mathbf{w}$
  - CCMpred initializes single potentials $\mathbf{v}_i(a) = \log f_i(a) - \log f_i(a = \text{"}-\text{"})$ with $f_i(a)$ being the frequency of amino acid a at position i and $a = \text{"}-\text{"}$ representing a gap. A single pseudo-count has been added before computing the frequencies. Pair potentials $\mathbf{w}$ are intialized at 0.
  - CCMpredPy initializes single potentials $\mathbf{v}$ with the ML estimate of single potentials (see section **??**) using amino acid frequencies computed as described in section 6.2.4. Pair potentials $\mathbf{w}$ are initialized at 0.
- Regularization
  - CCMpred uses a Gaussian regularization prior centered at zero for both single and pair potentials. The regularization coefficient for single potentials $\lambda_v = 0.01$ and for pair potentials $\lambda_w = 0.2 * (L-1)$ with L being protein length.
  - CCMpredPy uses a Gaussian regularization prior centered at zero for the pair potentials. For the single potentials the Gaussian regularization prior is centered at the ML estimate of single potentials (see section **??**)

using amino acid frequencies computed as described in section 6.2.4. The regularization coefficient for single potentials $\lambda_v = 10$ and for pair potentials $\lambda_w = 0.2 * (L - 1)$ with L being protein length.

Default settings for CCMpredPy have been chosen to best reproduce CCMpred results. A benchmark over a subset of approximately 3000 proteins confirms that performance measured as PPV for both methods is almost identical (see Figure 6.2).

The benchmark in Figure 6.2 as well as all contacts predicted with CCMpred and CCMPredPy (using pseudo-likelihood) in my thesis have been computed using the following flags:

Flags used with CCMpredPy (using pseudo-likelihood objective function):

```
--maxit 250                          # Compute a maximum of MAXIT operations
--center-v                           # Use a Gaussian prior for single potentials c
--reg-l2-lambda-single 10            # regularization coefficient for single potent
--reg-l2-lambda-pair-factor 0.2      # regularization coefficient for pairwise pote
--pc-uniform                         # use uniform pseudocounts (1/21 for 20 amino
--pc-count 1                         # defining pseudo count admixture coefficient
--epsilon 1e-5                       # convergence criterion for minimum decrease i
--ofn-pll                            # using pseudo-likelihood as objective functio
--alg-cg                             # using conjugate gradient to optimize objecti
```

Flags used with CCMpred:

```
-n 250    # NUMITER:  Compute a maximum of NUMITER operations
-l 0.2    # LFACTOR:  Set pairwise regularization coefficients to LFACTOR * (L-1
-w 0.8    # IDTHRES:  Set sequence reweighting identity threshold to IDTHRES
-e 1e-5   # EPSILON:  Set convergence criterion for minimum decrease in the last
```

6.2.3 Sequence Reweighting

As discussed in section 1.6, sequences in a MSA do not represent independent draws from a probabilistic model. To reduce the effects of overrepresented sequences, typically a simple weighting strategy is applied that assigns a weight to each sequence that is the inverse of the number of similar sequences according to an identity threshold [81]. It has been found that reweighting improves contact prediction performance [45,76,97] significantly but results are robust against the choice of the identity threshold in a range between 0.7 and 0.9 [76]. We chose an identity threshold of 0.8.

Every sequence x_n of length L in an alignment with N sequences has an associated weight $w_n = 1/m_n$, where m_n represents the number of similar sequences:

$$w_n = \frac{1}{m_n}, m_n = \sum_{m=1}^{N} I\left(ID(x_n, x_m) \geq 0.8\right) ID(x_n, x_m) = \frac{1}{L} \sum_{i=1}^{L} I(x_n^i = x_m^i) \quad (6.3)$$

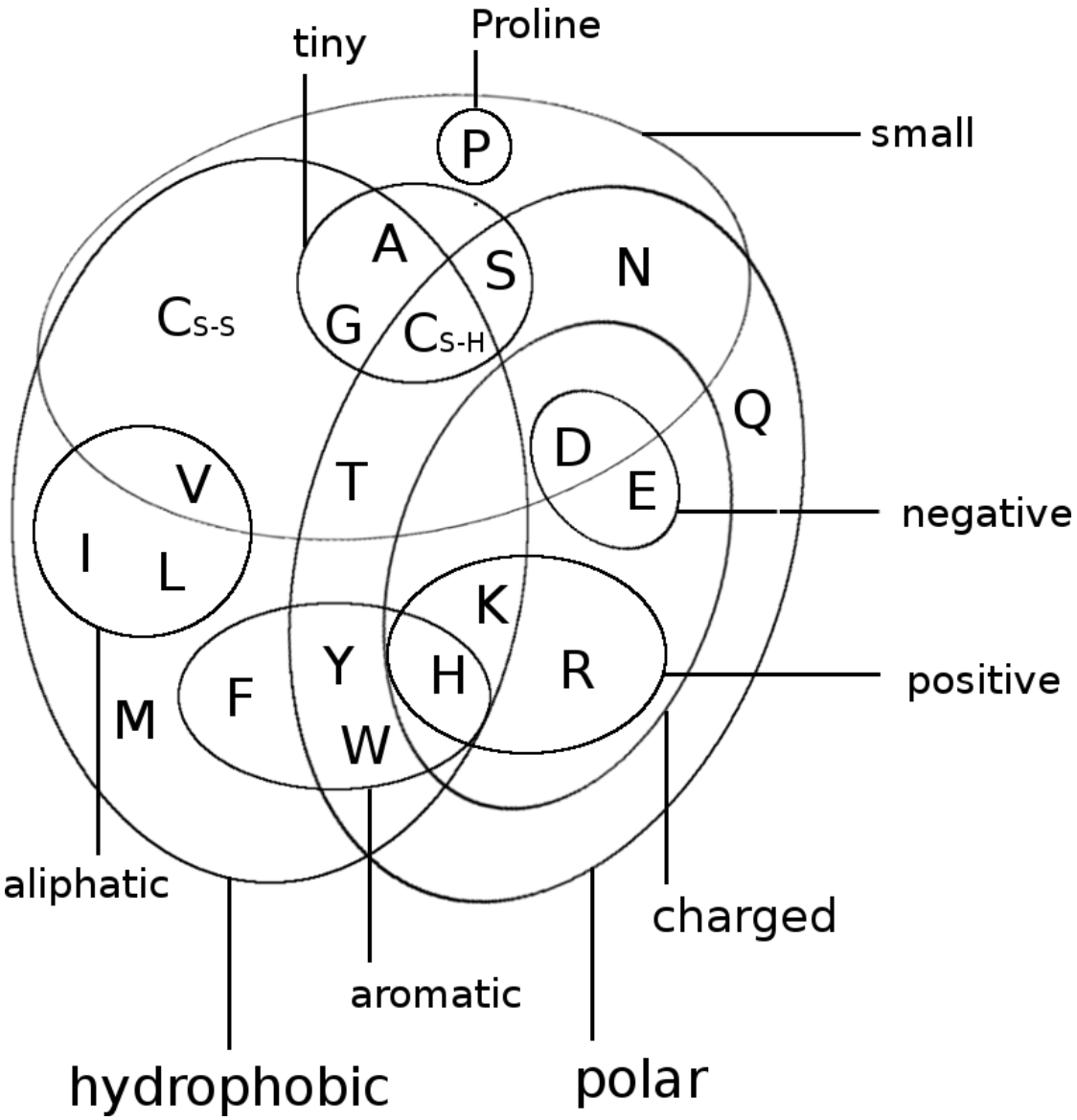


Figure 6.2: Benchmark for CCMpred and CCMpredPy on a dataset of 3124 proteins. ccmpred-vanilla+apc: CCMpred [82] with APC. ccmpred-pll-centerv+apc: CCMpredPy with APC. Specific flags that have been used to run both methods are described in detail in the text (see section 6.2.2).

The number of effective sequences $\mathbf{N}_{\text{eff}}$ of an alignment is then the number of sequence clusters computed as:

$$N_{\text{eff}} = \sum_{n=1}^{N} w_n \tag{6.4}$$

TODO: Plot Performance for Seq weighting

6.2.4 Computing Amino Acid Frequencies

Single and pairwise amino acid frequencies are computed from the alignment by weighting amino acid counts (see section 6.2.3) and adding pseudocounts for numerical stability.

Let $a, b \in \{1, \ldots, 20\}$ be amino acids, $q(x_i = a), q(x_i = a, x_j = b)$ and $q_0(x_i = a), q_0(x_i = a, x_j = b)$ be the empirical single and pair frequencies with and without pseudocounts, respectively. We define

$$q(x_i{=}a) := (1-\tau)\; q_0(x_i{=}a) + \tau \tilde{q}(x_i{=}a) \tag{6.5}$$

$$q(x_i{=}a, x_j{=}b) := (1-\tau)^2\; [q_0(x_i{=}a, x_j{=}b) - q_0(x_i{=}a) q_0(x_j{=}b)] + \tag{6.6}$$

$$q(x_i{=}a)\; q(x_j{=}b) \tag{6.7}$$

with $\tilde{q}(x_i{=}a) := f(a)$ being background amino acid frequencies and $\tau \in [0, 1]$ is a pseudocount admixture coefficient, which is a function of the diversity of the multiple sequence alignment:

$$\tau = \frac{N_{\text{pc}}}{(N_{\text{eff}} + N_{\text{pc}})} \tag{6.8}$$

where $N_{pc} > 0$.

The formula for $q(x_i{=}a, x_j{=}b)$ in the second line in eq (6.7) was chosen such that for $\tau = 0$ we obtain $q(x_i = a, x_j = b) = q_0(x_i = a, x_j = b)$, and furthermore $q(x_i = a, x_j{=}b) = q(x_i{=}a)q(x_j{=}b)$ exactly if $q_0(x_i{=}a, x_j{=}b) = q_0(x_i{=}a)q_0(x_j{=}b)$.

6.3 Analysis of Coupling Matrices

6.3.1 Correlation of Couplings with Contact Class

Approximately 100000 residue pairs have been filtered for contacts and non-contacts respectively according to the following criteria:

- consider only residue pairs separated by at least 10 positions in sequence
- minimal diversity ($= \frac{\sqrt{N}}{L}$) of alignment = 0.3
- minimal number of non-gapped sequences = 1000
- C_β distance threshold for contact: $< 8\mathbf{r}A$
- C_β distance threshold for noncontact: $> 25\mathbf{r}A$

6.3.2 Coupling Distribution Plots

For one-dimensional coupling distribution plots the residue pairs and respective pseudo-log-likelihood coupling values w_{ijab} have been selected as follows:

- consider only residue pairs separated by at least 10 positions in sequence
- discard residues that have more than 30% gaps in the alignment
- discard residue pairs that have insufficient evidence in the alignment: $N_{ij} \cdot q_i(a) \cdot q_j(b) < 100$ with:
  - N_{ij} is the number of sequences with neither a gap at position i nor at position j
  - $q_i(a)$ and $q_j(b)$ are the frequencies of amino acids a and b at positions i and j (computed as described in section 6.2.4)

These criteria ensure that uninformative couplings are neglected, e.g. sequence neighbors albeit being contacts according to the C_β contact definition cannot be assumed to express biological meaningful coupling patterns, or couplings for amino acid pairings that do not have statistical relevant counts in the alignment.

The same criteria have been applied for selecting couplings for the two-dimensional distribution plots with the difference that evidence for a single coupling term has to be $N_{ij} \cdot q_i(a) \cdot q_j(b) < 80$.

6.4 Optimizing the Full-Likelihood

Given the likelihood gradient estimates obtained with CD, the full likelihood can now be minimized using a gradient descent optimization algorithm. Second order optimization algorithms that make use of the (approximate) partial second derivates cannot be applied here, as the computation of the second derivatives of the full likelihood is too complex.

The following sections will describe the hyperparameter tuning for the stochastic gradient descent optimization as well as tuning different aspects of the Gibbs Sampler used to approximate the gradient with CD. The performance will be evaluated as the mean precision of the top ranked contact predictions over a benchmark set of 300 proteins, that is a subset of the data set described in methods section 6.1. The reference method for all new developments is the pseudo-likelihood method that uses the APC corrected L2norm as a contact score as explained in section 1.4.4. Pseudo-likelihood couplings are computed with the tool CCMpredPy that is introduced in methods section 6.2.2. Contact scores for couplings optimized with CD are computed in the same way as for the pseudo-likelihood.

6.4.1 Hyperparameter Optimization for Stochastic Gradient Descent

Gradient descent algorithms in general minimize an objective function by iteratively updating the function parameters in the opposite direction of the gradient of

the objective function with respect to the parameters. Stochastic gradient descent SGD is a variant thereof that uses only a subsample of the data at each step of the optimization procedure to estimate the gradient. Consequently, the gradient estimates are noisy resulting in parameter updates with high variance and strong fluctuations of the objective function. These fluctuation enable stochastic gradient descent to escape local minima but also complicate finding the exact minimum of the objective function. By slowly decreasing the step size of the parameter updates at every iteration, stochastic gradient descent most likely will converge to the global minimum for convex objective functions [128,129]. However, choosing an optimal step size for parameter updates, referred to as learning rate, as well as finding the optimal rate of decay offers a challenge and needs manual tuning [130].

When optimizing CD with SGD, the stochasticity is not so much introduced by looking only at subsamples of the data but rather by the stochastic nature of the Gibbs sampling process to approximate the gradient. The coupling parameters $\mathbf{w}$ will be updated in each iteration t by taking a step of size α along the direction of the negative gradient of the full log likelihood $-\nabla_w LL_{\text{reg}}(\mathbf{v}, \mathbf{w})$ that has been approximated with contrastive divergence,

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \cdot \nabla_w LL_{\text{reg}}(\mathbf{v}, \mathbf{w}) \; . \tag{6.9}$$

In order to get a first intuition of the optimization problem, I used an initial learning rate $\alpha_0 \in \{1\text{e}{-4}, 5\text{e}{-4}, 1\text{e}{-3}, 5\text{e}{-3}\}$ with a standard learning rate annealing schedule $\alpha = \frac{\alpha_0}{1+\gamma \cdot t}$, with $\gamma = 0.1$ being the decay rate and t the current iteration [129]. Typically, the algorithm has converged and is stopped whenever the difference of subsequent function evaluations is less than a small value ϵ. However, as it is not feasible to evaluate the full likelihood function at each iteration, because of the exponential complexity of the partition function, it is necessary to define a different stopping criteron. Therefore the optimization will stop whenever the gradient norm changes less than a small $\epsilon = 1\text{e}{-8}$ as in [131] or when a maximum of 5000 iterations has been reached. Furthermore, parameters are initialized with the pseudo-likelihood optimum, assuming that it is already close to the full likelihood optimum.

Figure 6.3 shows the mean precision against top ranked contacts computed from pseudo-likelihood couplings and CD couplings optimized with four different learning rates. Overall, mean precision for CD contacts is comparable to pseudo-likelihood contacts. Using smaller learning rates with the stochastic gradient descent optimizer results in higher mean precision.

When evaluating the learning rate settings with respect to alignment size (see Figure 6.4) it seems that higher learning rates do not work well for proteins with large alignments. The magnitude of the gradient scales with the number of sequences in the alignment because the gradient is computed as a difference of amino acid counts between observed and sampled sequences (see section 3.5).
And since alignments with many sequences produce high amino acid counts, their gradients will generally be higher compared to alignments with few sequences, especially at the beginning of the optimization procedure when the difference in amino acid counts between sampled and observed sequences is still big. Analysis of individual proteins with large alignments that perform bad in the benchmark

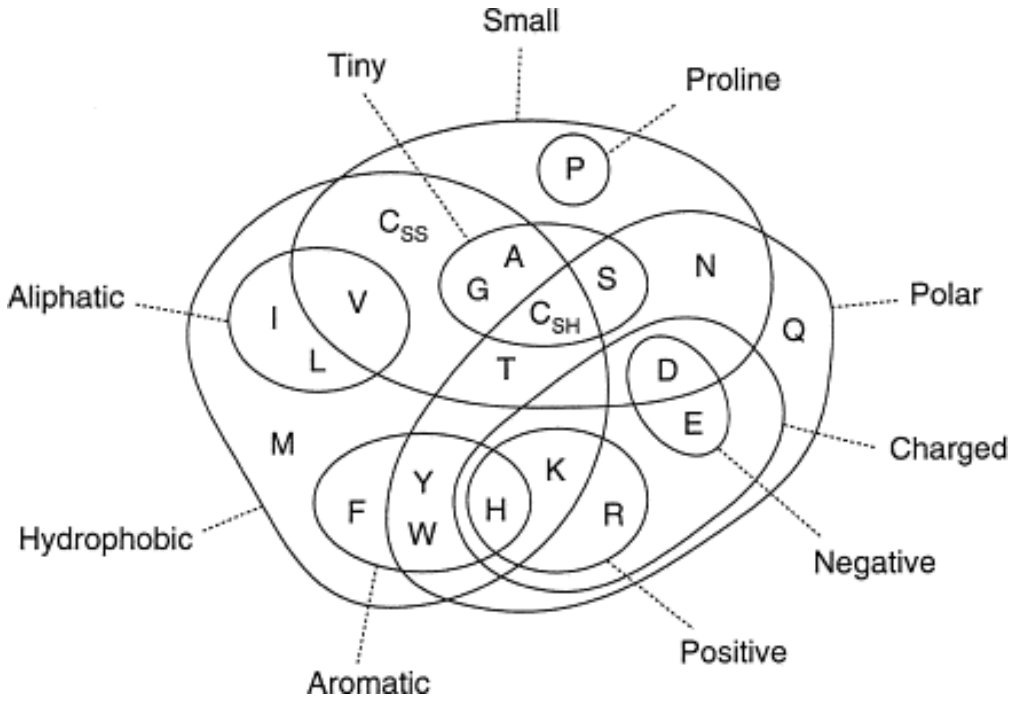


Figure 6.3: Mean precision for top ranked contact predictions over 286 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. pseudo-likelihood: Contact scores computed from pseudo-likelihood. The other methods derive contact scores from couplings computed from CD using stochastic gradient descent with different initial learning rates α_0 as specified in the legend.

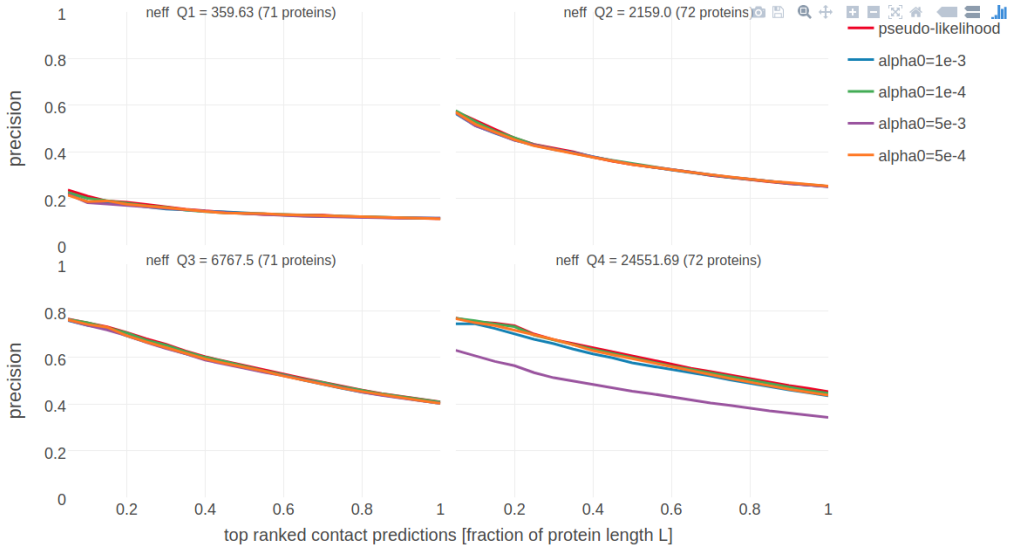


Figure 6.4: Mean precision for top ranked contact predictions over 286 proteins splitted into four equally sized subsets according to Neff. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. Subsets are defined according to quantiles of Neff values. Upper left: Subset of proteins with Neff $<$ Q1. Upper right: Subset of proteins with Q1 $<=$ Neff $<$ Q2. Lower left: Subset of proteins with Q2 $<=$ Neff $<$ Q3. Lower right: Subset of proteins with Q3 $<=$ Neff $<$ Q4. Methods are the same as in Figure 6.3.

reveals that high learning rates lead the parameters to diverge as can be seen in appendix D.1.

Furthermore, the smaller the learning rate, the more iterations are necessary for the optimization ot converge. With a learning rate of 1e$-$4 no protein converged with less than the maximum number of 5000 iterations (see Figure in Appendix D.1). It is therefore necessary to find a more appropriate learning rate schedule and decay rate.

I evaluated the following learning rate schedules and decay rates for an initial learning rate $\alpha_0 = 1\text{e}{-4}$ which gave highest precision in the previous benchmark but slow convergence:

- default learning rate schedule $\alpha = \frac{\alpha_0}{1+\gamma \cdot t}$ with $\gamma \in \{1\text{e}{-2}, 1\text{e}{-1}, 1\}$
- square root learning rate schedule $\alpha = \frac{\alpha_0}{\sqrt{1+\gamma \cdot t}}$ with $\gamma \in \{1\text{e}{-2}, 1\text{e}{-1}, 1, 10\}$
- sigmoidal learning rate schedule $\alpha_{t+1} = \frac{\alpha_t}{1+\gamma \cdot t}$ with $\gamma \in \{1\text{e}{-5}, 1\text{e}{-4}, 1\text{e}{-3}, 1\text{e}{-2}\}$

The decay schedules with different decay rates are visualized in Figure 6.5

Only the sigmoidal learning rate schedule achieves precision comparable to the pseudo-likelihood score while improving convergence speed measured by the number of iterations. Appendix D.3 shows benchmarks as well as the distribution over the number of iterations until convergence for all learning rate schedules that have been evaluated. Whereas none of the proteins except one did converge within 5000 iterations using the default learning rate schedule with decay rate $\gamma = 1\text{e}{-1}$ (blue box plot in Figure 6.6), all proteins converged within 2000 or 1000 iterations using

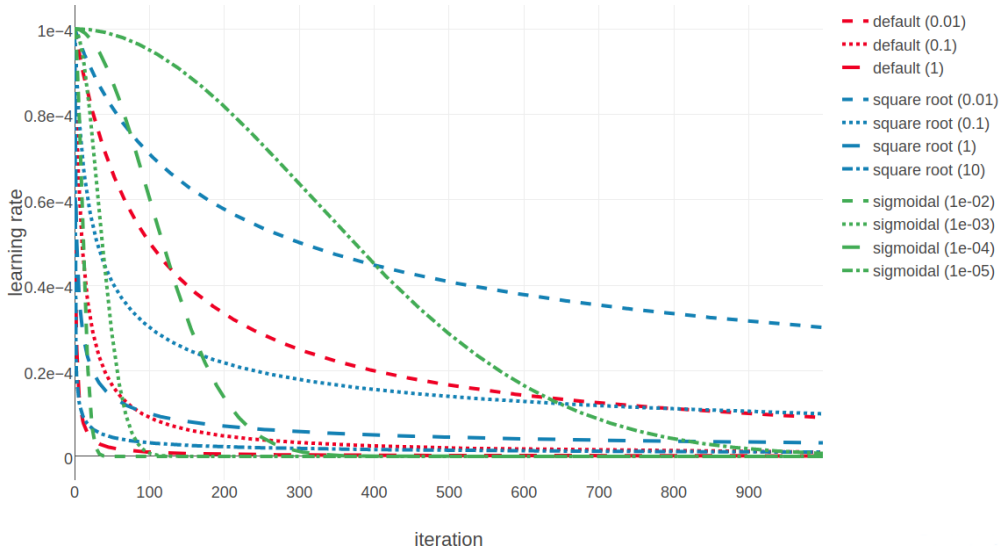


Figure 6.5: Value of learning rate against the number of iterations for different learning rate schedules. Red legend group represents the default learning rate schedule $\alpha = \alpha_0/(1 + \gamma \cdot t)$. Blue legend represents the sigmoidal learning rate schedule $\alpha_{t+1} = \alpha_t/(1 + \gamma \cdot t)$ with γ. Green legend represents the square root learning rate schedule $\alpha = \alpha_0/\sqrt{1 + \gamma \cdot t}$. The iteration number is given by t. Initial learning rate α_0 is set to 1e-4 and γ is the decay rate and its value is given in brackets in the legend.

the sigmoidal learning rate schedule with decay rates $\gamma = 1e{-}5$ and $\gamma = 1e{-}4$, respectively (orange and green box plot respectively in Figure 6.6).

Because the gradient of the full likelihood scales with number of sequences in the MSA, I defined the initial learning rate α_0 and the decay rate γ for the sigmoidal learning rate schedule as functions of Neff. Aiming to obtain values for the initial learning rate α_0 and the decay rate γ around 1e-4, as these values yield good performance and fast convergence, I defined $\alpha_0 = \gamma = \frac{3e{-}4}{\log N_{\text{eff}}}$. Assuming small Neff ≈ 50 this yields $\alpha_0 = \gamma \approx 1.7e{-}4$ and for big Neff ≈ 20000 this yields $\alpha_0 = \gamma \approx 7e{-}5$. The distribution of the number of iterations until convergence is displayed as red box plot in Figure 6.6, and is in the range of the sigmoidal learning rate schedule with decay rates $\gamma = 1e{-}5$ and $\gamma = 1e{-}4$, as expected.

All following analyses are conducted using the sigmoidal learning rate schedule with initial learning rate and decay rate defined as functions of Neff as $\alpha_0 = \gamma = \frac{3e{-}4}{\log N_{\text{eff}}}$.

(ref:caption-distribution-num-it-for-best-learning-rate-schedules) Distribution of the number of iterations until convergence for gradient descent optimizations of the full likelihood for different learning rate schedules. Initial learning rate α_0 is fixed to 1e-4 and maximum number of iterations is set to 5000. Blue box plot displays default learning rate schedule with decay rate $\gamma = 1e{-}1$. Orange and green boxplot represent sigmoidal learning rate schedule with decay rates $\gamma = 1e{-}5$ and $\gamma = 1e{-}4$, respectively. Red boxplot displays sigmoidal learning rate schedule with α_0 and decay rate γ defined as functions of Neff.

Interestingly, and in contradiction to the benchmark results, using higher initial

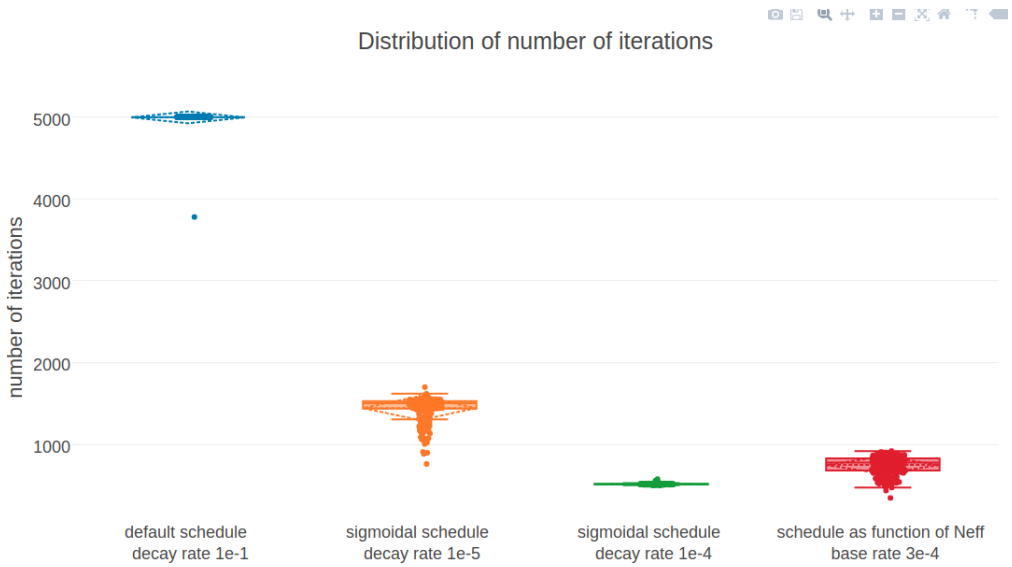


Figure 6.6: (ref:caption-distribution-num-it-for-best-learning-rate-schedules)

learning rates usually leads the parameters to diverge further from the pseudo-likelihood initializations than using smaller and faster decaying learning rates. Figure **??** illustrates this effect for protein *1mkc*. Determining the learning and decay rate with respect to Neff, as just described, yields $\alpha_0 = \gamma \approx 6.6e-5$. Optimization converges after $\approx$ 600 iterations with an L2-norm over couplings $||\mathbf{w}||_2 = 15.28$. When using a larger learning rate $\alpha_0 = 1e-3$ the optimization has converged after $\approx$ 560 iterations with an L2-norm over couplings $||\mathbf{w}||_2 = 12.66$.

6.4.2 Optimizing Regularization Coefficients for Contrastive Divergence

Gaussian priors are put on the single potentials $\mathbf{v}$ and the couplings $\mathbf{w}$ when optimizing the full likelihood with CD just as it is done for pseudo-likelihood optimization (compare section 1.4.3.1). A difference compared to pseudo-likelihood optimization that uses zero centered priors, is the centering of the Gaussian prior for the single potentials $\mathbf{v}$ at v^* as described in section 3.4. The hyperparameter tuning for stochastic gradient descent described in the last section applied the default pseudo-likelihood regularization coefficients $\lambda_v = 10$ and $\lambda_w = 0.2 \cdot (L - 1)$. The regularization coefficient λ_w for couplings $\mathbf{w}$ is defined with respect to protein length L owing to the fact that the number of possible contacts in a protein increases quadratically with L whereas the number of observed contacts only increases linearly as can be seen in Figure 6.7.

It is possible that CD achieves optimal performance using stronger or weaker regularization coefficients compared to pseudo-likelihood. Therefore, I evaluated performance of contrastive divergence using different regularization coefficients $\lambda_w \in \{1e{-}2, 1e{-}1, 0.2, 1\} \cdot (L - 1)$ while leaving the regularization for single potentials at the default value $\lambda_v = 10$. Furthermore, I analysed whether precision is impacted by only optimizing the couplings $\mathbf{w}$ (with default regularization) while fixing the single potentials v_i to their best estimates v_i^* as described in section 3.4.

Figure 6.7: Number of contacts ($C_\beta < 8\mathring{A}$) with respect to protein length and sequence separation has a linear relationship.

As can be seen in Figure 6.8, using strong regularization for the couplings $\lambda_w = (L-1)$ results in a drop of mean precision. Using weaker regularization or fixing the single potentials hardly has an impact on precision with using $\lambda_w = 1e{-}2(L-1)$ yielding slightly improved precision for the top ranked contacts.

6.4.3 Optimizing the Sampling Scheme for Contrastive Divergence

I analysed whether choosing a different number of sequences for the approximation of the gradient via Gibbs sampling can improve performance. Randomly selecting only a subset N' of the N observed sequences corresponds to the stochastic gradient descent idea of a minibatch and introduces additional stochasticity over the random Gibbs sampling process. Using $N' < N$ sequences for Gibbs sampling has the further advantage of decreasing the runtime at each iteration. Note, that the reference counts from the observed sequences $N_{ij}q(x_i = a, x_j = b)$ that are part of the gradient calculation will be kept constant. Therefore it is necessary, however to rescale the amino acid counts from the sampled sequences in a way such that the total sample counts match the total observed counts.

I evaluated two different schemes for randomly selecting $N' = xL$ sequences from the N given sequences of the alignment at every iteration:

- **without** replacement (enforcing $N' = \min(N, xL)$)
- **with** replacement

with $x \in \{1, 5, 10, 50\}$.

As can be seen in the Figure **??**, the choice of minibatch size which corresponds to the number sequences that are selected to approximate the gradient, has no influence on precision.

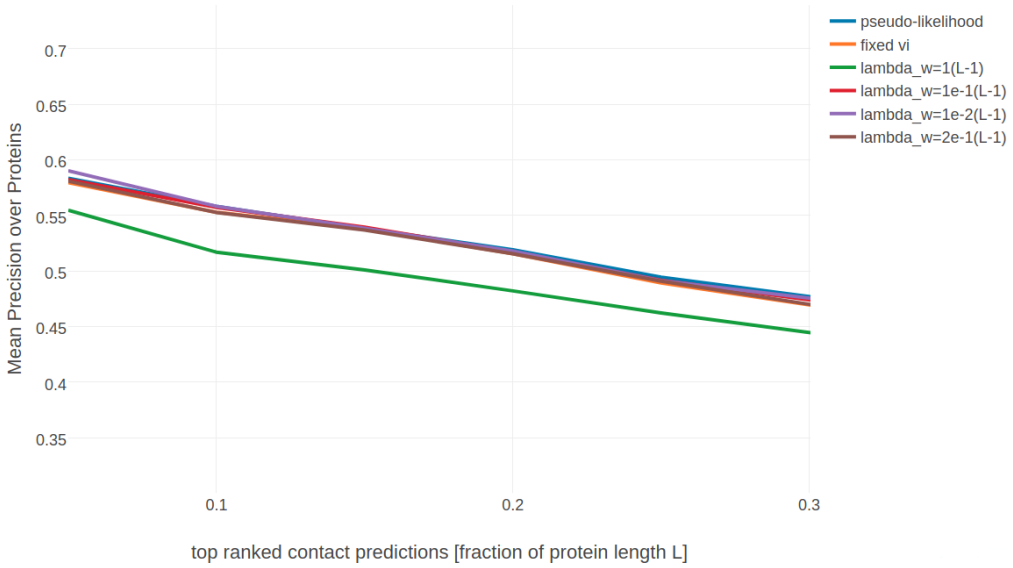


Figure 6.8: Performance of contrastive divergence optimization of the full likelihood with different regularization settings compared to pseudo-likelihood (blue) for 280 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. Default regularization coefficients as used with pseudo-likelihood are $\lambda_v = 10$ and $\lambda_w = 0.2(L-1)$. "fixed vi" (orange) uses CD to optimize only couplings with default regularization while keeping the single potentials v_i fixed at their MLE optimum v_i^*. The other optimization runs with CD (green, red, purple, brown) use default regularization for the single potentials and a regularization coefficient for the couplings according to legend description.

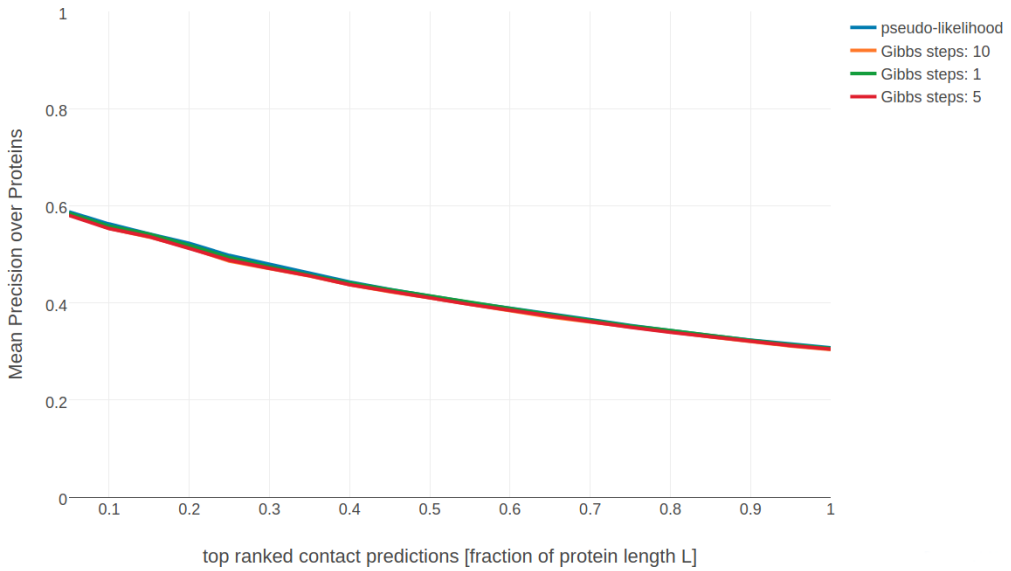


Figure 6.9: Performance of contrastive divergence optimization of the full likelihood with different number of Gibbs steps compared to pseudo-likelihood (blue) for 287 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. pseudo-likelihood: contact scores computed from pseudo-likelihood. The other methods derive contact scores from couplings computed from CD with different number of Gibbs sampling steps.

PLOT PEFORMANCE SMAPLING SIZE

This results is somewhat unexpected, because using more samples to approximate the gradient should result in a better gradient approximation and thus in a better performance. Indeed, the magnitude of the gradient norms decreases when more sequences are used for sampling as can be seen in Figure **??**. However, this does apparantly not translate into better parameter values.

PLOT GRADIENT NORMS

The default CD algorithm as described by Hinton in 2002 applies only one full step of Gibbs sampling on each data sample to generate a sampled data set that will be used to approximate the gradient [116]. One full step of Gibbs sampling corresponds to sampling each position in a protein sequence according to the conditional probabilities computed from the current model probabilties as described in **??**. The sampled sequences obtained after only one step of Gibbs sampling will be very similar to the input sequences. It has been shown that sampling with $n > 1$ steps gives more precise results but increases computational cost per gradient evaluation [132,133].

In the following I analysed the impact on performance when Gibbs sampling each sequence with 1, 5 and 10 full steps. As can be seen, there is hardly an impact on precision while having much longer runtimes (by a factor or 5 and 10).

Another variant of CD that has been suggested by Tieleman in 2008 is *PCD*[133] that does not reset the Markov Chains at every iteration. The reason being that when using small learning rates, the model changes only slightly between iterations and the true data distribution can be better approximated However, subsequent samples of *PCD* will be highly correlated creating a kind of momentum effect.

Furthermore is has been found that *PCD* should be used with smaller learning rates and higher minibatch sizes.

As PCD might require samller update steps and larger minibatches, I analysed the performance of PCD for the default settings of CD and additionally for smaller learning and decay rates and larger minibatches. Note that one Markov chain is kept for every sequence of the input alignment. At each iteration a subset $N' < N$ of the Markov chains is randomly selected (without replacement) and used to for another round of Gibss sampling at the current iteration.

PLOT PCD for different LEARNIGN RATEWS and SAMPLE SIZES

6.5 Bayesian Model for Residue-Resdiue Contact Prediction

6.5.1 Efficiently Computing the negative Hessian of the regularized log-likelihood

Surprisingly, the elements of the Hessian at the mode $\mathbf{w}^*$ are easy to compute. Let $i, j, k, l \in \{1, \ldots, L\}$ be columns in the MSA and let $a, b, c, d \in \{1, \ldots, 20\}$ represent amino acids.

The partial derivative $\partial / \partial \mathbf{w}_{klcd}$ of the second term in the gradient of the couplings in eq. (3.23) is

$$\frac{\partial^2 LL_{\text{reg}}(\mathbf{v}^*, \mathbf{w})}{\partial w_{klcd}\, \partial w_{ijab}} = -\sum_{n=1}^{N} \sum_{\mathbf{y} \in S_n} \frac{\partial \left(\frac{\exp\left(\sum_{i=1}^{L} v_i(y_i) + \sum_{1 \le i < j \le L}^{L} w_{ij}(y_i, y_j)\right)}{Z_n(\mathbf{v}, \mathbf{w})} \right)}{\partial w_{klcd}} I(y_i = a, y_j = b) \tag{6.10}$$

$$-\lambda_w \delta_{ijab,klcd}\,, \tag{6.11}$$

where $\delta_{ijab,klcd} = I(ijab = klcd)$ is the Kronecker delta. Applying the product rule, we find

$$\frac{\partial^2 LL_{\text{reg}}(\mathbf{v}^*, \mathbf{w})}{\partial w_{klcd}\, \partial w_{ijab}} = -\sum_{n=1}^{N} \sum_{\mathbf{y} \in S_n} \frac{\exp\left(\sum_{i=1}^{L} v_i(y_i) + \sum_{1 \le i < j \le L}^{L} w_{ij}(y_i, y_j)\right)}{Z_n(\mathbf{v}, \mathbf{w})} I(y_i = a, y_j = b) \tag{6.12}$$

$$\times \left[\frac{\partial}{\partial w_{klcd}} \left(\sum_{i=1}^{L} v_i(y_i) + \sum_{1 \le i < j \le L} w_{ij}(y_i, y_j) \right) - \frac{1}{Z_n(\mathbf{v}, \mathbf{w})} \frac{\partial Z_n(\mathbf{v}, \mathbf{w})}{\partial w_{klcd}} \right] \tag{6.13}$$

$$- \lambda_w \delta_{ijab,klcd} \tag{6.14}$$

$$\frac{\partial^2 LL_{\text{reg}}(\mathbf{v}^*, \mathbf{w})}{\partial w_{klcd}\, \partial w_{ijab}} = -\sum_{n=1}^{N} \sum_{\mathbf{y} \in S_n} \frac{\exp\left(\sum_{i=1}^{L} v_i(y_i) + \sum_{1 \le i < j \le L}^{L} w_{ij}(y_i, y_j)\right)}{Z_n(\mathbf{v}, \mathbf{w})} I(y_i = a, y_j = b) \tag{6.15}$$

$$\times \left[I(y_k = c, y_l = d) - \frac{\partial}{\partial w_{klcd}} \log Z_n(\mathbf{v}, \mathbf{w}) \right] \tag{6.16}$$

$$- \lambda_w \delta_{ijab,klcd}\,. \tag{6.17}$$

We simplify this expression using

$$p(\mathbf{y}|\mathbf{v},\mathbf{w}) = \frac{\exp\left(\sum_{i=1}^{L} v_i(y_i) + \sum_{1\le i<j\le L} w_{ij}(y_i,y_j)\right)}{Z_n(\mathbf{v},\mathbf{w})}, \tag{6.18}$$

yielding

$$\begin{aligned}
\frac{\partial^2 LL_{\mathrm{reg}}(\mathbf{v}^*,\mathbf{w})}{\partial w_{klcd}\,\partial w_{ijab}} &= -\sum_{n=1}^{N}\sum_{\mathbf{y}\in S_n} p(\mathbf{y}|\mathbf{v},\mathbf{w})\, I(y_i{=}a, y_j{=}b, y_k{=}c, y_l{=}d) && (6.19)\\
&+ \sum_{n=1}^{N}\sum_{\mathbf{y}\in \mathcal{S}_n} p(\mathbf{y}|\mathbf{v},\mathbf{w})\, I(y_i{=}a, y_j{=}b) \sum_{\mathbf{y}\in S_n} p(\mathbf{y}|\mathbf{v},\mathbf{w}) I(y_k{=}c, y_l{=}d) && (6.20)\\
&- \lambda_w \delta_{ijab,klcd}\,. && (6.21)
\end{aligned}$$

If $\mathbf{X}$ does not contain too many gaps, this expression can be approximated by

$$\begin{aligned}
\frac{\partial^2 LL_{\mathrm{reg}}(\mathbf{v}^*,\mathbf{w})}{\partial w_{klcd}\,\partial w_{ijab}} &= -N_{ijkl}\, p(x_i{=}a, x_j{=}b, x_k{=}c, x_l{=}d|\mathbf{v},\mathbf{w})\\
&\quad + N_{ijkl}\, p(x_i{=}a, x_j{=}b|\mathbf{v},\mathbf{w})\, p(x_k{=}c, x_l{=}d|\mathbf{v},\mathbf{w}) - \lambda_w \delta_{ijab,klcd} && (6.22)
\end{aligned}$$

where N_{ijkl} is the number of sequences that have a residue in i, j, k and l.

Looking at three cases separately:

- case 1: $(k,l) = (i,j)$ and $(c,d) = (a,b)$
- case 2: $(k,l) = (i,j)$ and $(c,d) \neq (a,b)$
- case 3: $(k,l) \neq (i,j)$ and $(c,d) \neq (a,b)$,

the elements of $\mathbf{H}$, which are the negative second partial derivatives of $LL_{\mathrm{reg}}(\mathbf{v}^*,\mathbf{w})$ with respect to the components of $\mathbf{w}$, are

$$\begin{aligned}
\text{case 1}: (\mathbf{H})_{ijab,ijab} &= N_{ij}\, p(x_i{=}a, x_j{=}b|\mathbf{v}^*,\mathbf{w}^*)\,(1 - p(x_i{=}a, x_j{=}b|\mathbf{v}^*,\mathbf{w}^*)) && (6.23)\\
&\quad + \lambda_w && (6.24)\\
\text{case 2}: (\mathbf{H})_{ijcd,ijab} &= -N_{ij}\, p(x_i{=}a, x_j{=}b|\mathbf{v}^*,\mathbf{w}^*)\, p(x_i{=}c, x_j{=}d|\mathbf{v}^*,\mathbf{w}^*) && (6.25)\\
\text{case 3}: (\mathbf{H})_{klcd,ijab} &= N_{ijkl}\, p(x_i{=}a, x_j{=}b, x_k{=}c, x_l{=}d|\mathbf{v}^*,\mathbf{w}^*)\\
&\quad - N_{ijkl}\, p(x_i{=}a, x_j{=}b|\mathbf{v}^*,\mathbf{w}^*)\, p(x_k{=}c, x_l{=}d|\mathbf{v}^*,\mathbf{w}^*) && (6.26)
\end{aligned}$$

We know from eq. (3.31) that at the mode $\mathbf{w}^*$ the model probabilities match the empirical frequencies up to a small regularization term,

$$p(x_i{=}a, x_j{=}b|\mathbf{v}^*,\mathbf{w}^*) = q(x_i{=}a, x_j{=}b) - \frac{\lambda_w}{N_{ij}} w^*_{ijab}\,, \tag{6.27}$$

and therefore the negative Hessian elements in cases 1 and 2 can be expressed as

$$(\mathbf{H})_{ijab,ijab} = N_{ij}\left(q(x_i{=}a, x_j{=}b) - \frac{\lambda_w}{N_{ij}} w^*_{ijab}\right)\left(1 - q(x_i{=}a, x_j{=}b) + \frac{\lambda_w}{N_{ij}} w^*_{ijab}\right) \tag{6.28}$$

$$+ \lambda_w \tag{6.29}$$

$$(\mathbf{H})_{ijcd,ijab} = -N_{ij}\left(q(x_i{=}a, x_j{=}b) - \frac{\lambda_w}{N_{ij}} w^*_{ijab}\right)\left(q(x_i{=}c, x_j{=}d) - \frac{\lambda_w}{N_{ij}} w^*_{ijcd}\right). \tag{6.30}$$

In order to write the previous eq. (6.30) in matrix form, the *regularised* empirical frequencies $\mathbf{q\prime}_{ij}$ will be defined as

$$(\mathbf{q\prime}_{ij})_{ab} = q'_{ijab} := q(x_i{=}a, x_j{=}b) - \lambda_w w^*_{ijab}/N_{ij}\,, \tag{6.31}$$

and the 400×400 diagonal matrix $\mathbf{Q}_{ij}$ will be defined as

$$\mathbf{Q}_{ij} := \mathrm{diag}(\mathbf{q\prime}_{ij})\,. \tag{6.32}$$

Now eq. (6.30) can be written in matrix form

$$\mathbf{H}_{ij} = N_{ij}\left(\mathbf{Q}_{ij} - \mathbf{q\prime}_{ij}\mathbf{q\prime}_{ij}^{\mathrm{T}}\right) + \lambda_w \mathbf{I}\,. \tag{6.33}$$

6.5.2 Efficiently Computing the Inverse of Matrix $\mathbf{\Lambda}_{ij,k}$

It is possible to efficiently invert the matrix $\mathbf{\Lambda}_{ij,k} = \mathbf{H}_{ij} - \lambda_w \mathbf{I} + \Lambda_k$, that is introduced in 4.2 where $\mathbf{H}_{ij}$ is the 400×400 diagonal block submatrix $(\mathbf{H}_{ij})_{ab,cd} := (\mathbf{H})_{ijab,ijcd}$ and Λ_k is an invertible diagonal precision matrix that is introduced in section **??**.

Equation (6.33) can be used to write $\mathbf{\Lambda}_{ij,k}$ in matrix form as

$$\mathbf{\Lambda}_{ij,k} = \mathbf{H}_{ij} - \lambda_w \mathbf{I} + \mathbf{\Lambda}_k = N_{ij}\mathbf{Q}_{ij} - N_{ij}\mathbf{q\prime}_{ij}\mathbf{q\prime}_{ij}^{\mathrm{T}} + \mathbf{\Lambda}_k\,. \tag{6.34}$$

Owing to eqs. (3.12) and (3.25), $\sum_{a,b=1}^{20} q'_{ijab} = 1$. The previous equation (6.34) facilitates the calculation of the inverse of this matrix using the *Woodbury identity* for matrices

$$(\mathbf{A} + \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{D} + \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1}\,. \tag{6.35}$$

by setting

$$\mathbf{A} = N_{ij}\mathbf{Q}_{ij} + \mathbf{\Lambda}_k \tag{6.36}$$
$$\mathbf{B} = \mathbf{q}\prime_{ij} \tag{6.37}$$
$$\mathbf{C} = \mathbf{q}\prime_{ij}^{\mathrm{T}} \tag{6.38}$$
$$\mathbf{D} = -N_{ij}^{-1} \tag{6.39}$$
$$\tag{6.40}$$

$$(\mathbf{H}_{ij} - \lambda_w \mathbf{I} + \mathbf{\Lambda}_k)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{q}\prime_{ij}\left(-N_{ij}^{-1} + \mathbf{q}\prime_{ij}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{q}\prime_{ij}\right)^{-1}\mathbf{q}\prime_{ij}^{\mathrm{T}}\mathbf{A}^{-1} \tag{6.41}$$
$$= \mathbf{A}^{-1} + \frac{(\mathbf{A}^{-1}\mathbf{q}\prime_{ij})(\mathbf{A}^{-1}\mathbf{q}\prime_{ij})^{\mathrm{T}}}{N_{ij}^{-1} - \mathbf{q}\prime_{ij}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{q}\prime_{ij}}\,. \tag{6.42}$$

Note that $\mathbf{A}$ is diagonal as $\mathbf{Q}_{ij}$ and $\mathbf{\Lambda}_k$ are diagonal matrices: $\mathbf{A} = \mathrm{diag}(N_{ij}q'_{ijab} + (\mathbf{\Lambda}_k)_{ab,ab})$. Moreover, $\mathbf{A}$ has only positive diagonal elements, because $\mathbf{\Lambda}_k$ is invertible and has only positive diagonal elements and because $q'_{ijab} = p(x_i = a, x_j = b|\mathbf{v}^*, \mathbf{w}^*) \geq 0$.

Therefore $\mathbf{A}$ is invertible: $\mathbf{A}^{-1} = \mathrm{diag}(N_{ij}q'_{ijab} + (\mathbf{\Lambda}_k)_{ab,ab})^{-1}$.

Because $\sum_{a,b=1}^{20} q'_{ijab} = 1$, the denominator of the second term is

$$N_{ij}^{-1} - \sum_{a,b=1}^{20} \frac{q'^2_{ijab}}{N_{ij}q'_{ijab} + (\mathbf{\Lambda}_k)_{ab,ab}} > N_{ij}^{-1} - \sum_{a,b=1}^{20} \frac{q'^2_{ijab}}{N_{ij}q'_{ijab}} = 0 \tag{6.43}$$

and therefore the inverse of $\mathbf{\Lambda}_{ij,k}$ in eq. (6.42) is well defined.

The log determinant of $\mathbf{\Lambda}_{ij,k}$ is necessary to compute the ratio of Gaussians (see equation (**??**)) and can be computed using the matrix determinant lemma:

$$\det(\mathbf{A} + \mathbf{u}\mathbf{v}^{\mathrm{T}}) = (1 + \mathbf{v}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{u})\det(\mathbf{A}) \tag{6.44}$$

Setting $\mathbf{A} = N_{ij}\mathbf{Q}_{ij} + \mathbf{\Lambda}_k$ and $\mathbf{v} = \mathbf{q}\prime_{ij}$ and $\mathbf{u} = -N_{ij}\mathbf{q}\prime_{ij}$ yields

$$\det(\mathbf{\Lambda}_{ij,k}) = \det(\mathbf{H}_{ij} - \lambda_w\mathbf{I} + \mathbf{\Lambda}_k) = (1 - N_{ij}\mathbf{q}\prime_{ij}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{q}\prime_{ij})\det(\mathbf{A})\,. \tag{6.45}$$

$\mathbf{A}$ is diagonal and has only positive diagonal elements so that $\log(\det(\mathbf{A})) = \sum \log(\mathrm{diag}(\mathbf{A}))$.

6.5.3 Training the Hyperparameters μ_k, $\mathbf{\Lambda}_k$ and γ_k

The model parameters $\mu = (\mu_1, \ldots, \mu_K)$, $\mathbf{\Lambda} = (\mathbf{\Lambda}_1, \ldots, \mathbf{\Lambda}_K)$ and $\gamma = (\gamma_1, \ldots, \gamma_K)$ will be trained by maximizing the logarithm of the full likelihood over a set of training MSAs $\mathbf{X}^1, \ldots, \mathbf{X}^N$ and associated structures with distance vectors $\mathbf{r}^1, \ldots, \mathbf{r}^N$ plus a regularizer $R(\mu, \mathbf{\Lambda})$:

$$LL(\mu, \mathbf{\Lambda}, \gamma) + R(\mu, \mathbf{\Lambda}) = \sum_{n=1}^{N} \log p(\mathbf{X}^n | \mathbf{r}^n, \mu, \mathbf{\Lambda}, \gamma) + R(\mu, \mathbf{\Lambda}) \to \max . \quad (6.46)$$

The regulariser penalizes values of μ_k and $\mathbf{\Lambda}_k$ that deviate too far from zero:

$$R(\mu, \mathbf{\Lambda}) = -\frac{1}{2\sigma_\mu^2} \sum_{k=1}^{K} \sum_{ab=1}^{400} \mu_{k,ab}^2 - \frac{1}{2\sigma_{\text{diag}}^2} \sum_{k=1}^{K} \sum_{ab=1}^{400} \Lambda_{k,ab,ab}^2 \quad (6.47)$$

Reasonable values are $\sigma_\mu = 0.1$, $\sigma_{\text{diag}} = 100$.

The log likelihood can be optimized using LBFG-S-B[**???**], which requires the computation of the gradient of the log likelihood. For simplicity of notation, the following calculations consider the contribution of the log likelihood for just one protein, which allows to drop the index n in r_{ij}^n, $(\mathbf{w}_{ij}^n)^*$ and $\mathbf{H}_{ij}^n$.

From eq. (4.41) the log likelihood for a single protein is

$$LL(\mu, \mathbf{\Lambda}, \gamma_k) = \sum_{1 \le i < j \le L} \log \sum_{k=0}^{K} g_k(r_{ij}) \frac{\mathcal{N}(\mathbf{0} | \mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0} | \mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} + R(\mu, \mathbf{\Lambda}) + \text{const.} . \quad (6.48)$$

6.5.4 The gradient of the log likelihood with respect to μ

By applying the formula $df(x)/dx = f(x)\, d \log f(x)/dx$ to compute the gradient of eq. (6.48) (neglecting the regularization term) with respect to $\mu_{k,ab}$, one obtains

$$\frac{\partial}{\partial \mu_{k,ab}} LL(\mu, \mathbf{\Lambda}, \gamma_k) = \sum_{1 \le i < j \le L} \frac{g_k(r_{ij}) \frac{\mathcal{N}(\mathbf{0} | \mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0} | \mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} \frac{\partial}{\partial \mu_{k,ab}} \log \left(\frac{\mathcal{N}(\mathbf{0} | \mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0} | \mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} \right)}{\sum_{k'=0}^{K} g_{k'}(r_{ij}) \frac{\mathcal{N}(\mathbf{0} | \mu'_k, \mathbf{\Lambda}'^{-1}_k)}{\mathcal{N}(\mathbf{0} | \mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})}} . \quad (6.49)$$

To simplify this expression, we define the responsibility of component k for the posterior distribution of $\mathbf{w}_{ij}$, the probability that $\mathbf{w}_{ij}$ has been generated by component k:

$$p(k|ij) = \frac{g_k(r_{ij}) \frac{\mathcal{N}(\mathbf{0} | \mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0} | \mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})}}{\sum_{k'=0}^{K} g_{k'}(r_{ij}) \frac{\mathcal{N}(\mathbf{0} | \mu'_k, \mathbf{\Lambda}'^{-1}_k)}{\mathcal{N}(\mathbf{0} | \mu'_{ij,k}, \mathbf{\Lambda}'^{-1}_{ij,k})}} . \quad (6.50)$$

By substituting the definition for responsibility, (6.49) simplifies

$$\frac{\partial}{\partial \mu_{k,ab}} LL(\mu, \mathbf{\Lambda}, \gamma_k) = \sum_{1 \le i < j \le L} p(k|ij) \frac{\partial}{\partial \mu_{k,ab}} \log \left(\frac{\mathcal{N}(\mathbf{0} | \mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0} | \mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} \right) , \quad (6.51)$$

and analogously for partial derivatives with respect to $\Lambda_{k,ab,cd}$.

The partial derivative inside the sum can be written

$$\frac{\partial}{\partial \mu_{k,ab}} \log \left(\frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} \right) = \frac{1}{2} \frac{\partial}{\partial \mu_{k,ab}} \left(\log |\mathbf{\Lambda}_k| - \mu_k^{\mathrm{T}} \mathbf{\Lambda}_k \mu_k - \log |\mathbf{\Lambda}_{ij,k}| + \mu_{ij,k}^{\mathrm{T}} \mathbf{\Lambda}_{ij,k} \mu_{ij,k} \right) . \tag{6.52}$$

Using the following formula for a matrix $\mathbf{A}$, a real variable x and a vector $\mathbf{y}$ that depends on x,

$$\frac{\partial}{\partial x} \left(\mathbf{y}^{\mathrm{T}} \mathbf{A} \mathbf{y} \right) = \frac{\partial \mathbf{y}^{\mathrm{T}}}{\partial x} \mathbf{A} \mathbf{y} + \mathbf{y}^{\mathrm{T}} \mathbf{A} \frac{\partial \mathbf{y}}{\partial x} = \mathbf{y}^{\mathrm{T}} (\mathbf{A} + \mathbf{A}^{\mathrm{T}}) \frac{\partial \mathbf{y}}{\partial x} \tag{6.53}$$

the partial derivative therefore becomes

$$\frac{\partial}{\partial \mu_{k,ab}} \log \left(\frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} \right) = \left(-\mu_k^{\mathrm{T}} \mathbf{\Lambda}_k \mathbf{e}_{ab} + \mu_{ij,k}^{\mathrm{T}} \mathbf{\Lambda}_{ij,k} \mathbf{\Lambda}_{ij,k}^{-1} \mathbf{\Lambda}_k \mathbf{e}_{ab} \right) \tag{6.54}$$

$$= \mathbf{e}_{ab}^{\mathrm{T}} \mathbf{\Lambda}_k (\mu_{ij,k} - \mu_k) . \tag{6.55}$$

Finally, the gradient of the log likelihood with respect to μ becomes

$$\nabla_{\mu_k} LL(\mu, \mathbf{\Lambda}, \gamma_k) = \sum_{1 \leq i < j \leq L} p(k|ij) \, \mathbf{\Lambda}_k \left(\mu_{ij,k} - \mu_k \right) . \tag{6.56}$$

6.5.5 The gradient of the log likelihood with respect to $\mathbf{\Lambda}_k$

Analogously to eq. (6.51) one first needs to solve

$$\frac{\partial}{\partial \Lambda_{k,ab,cd}} \log \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} = \tag{6.57}$$

$$\frac{1}{2} \frac{\partial}{\partial \Lambda_{k,ab,cd}} \left(\log |\mathbf{\Lambda}_k| - \mu_k^{\mathrm{T}} \mathbf{\Lambda}_k \mu_k - \log |\mathbf{\Lambda}_{ij,k}| + \mu_{ij,k}^{\mathrm{T}} \mathbf{\Lambda}_{ij,k} \mu_{ij,k} \right) , \tag{6.58}$$

by applying eq. (6.53) as before as well as the formulas

$$\frac{\partial}{\partial x} \log |\mathbf{A}| = \mathrm{Tr} \left(\mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial x} \right) , \tag{6.59}$$

$$\frac{\partial \mathbf{A}^{-1}}{\partial x} = -\mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial x} \mathbf{A}^{-1} . \tag{6.60}$$

This yields

$$\frac{\partial}{\partial \Lambda_{k,ab,cd}} \log |\mathbf{\Lambda}_k| = \mathrm{Tr}\left(\mathbf{\Lambda}_k^{-1} \frac{\partial \mathbf{\Lambda}_k}{\partial \Lambda_{k,ab,cd}}\right) = \mathrm{Tr}\left(\mathbf{\Lambda}_k^{-1} \mathbf{e}_{ab} \mathbf{e}_{cd}^{\mathrm{T}}\right) = \Lambda^{-1}_{k,cd,ab} \tag{6.61}$$

$$\frac{\partial}{\partial \Lambda_{k,ab,cd}} \log |\mathbf{\Lambda}_{ij,k}| = \mathrm{Tr}\left(\mathbf{\Lambda}_{ij,k}^{-1} \frac{\partial (\mathbf{H}_{ij} - \lambda_w \mathbf{I} + \mathbf{\Lambda}_k)}{\partial \Lambda_{k,ab,cd}}\right) = \Lambda^{-1}_{ij,k,cd,ab} \tag{6.62}$$

$$\frac{\partial (\mu_k^{\mathrm{T}} \mathbf{\Lambda}_k \mu_k)}{\partial \Lambda_{k,ab,cd}} = \mu_k^{\mathrm{T}} \mathbf{e}_{ab} \mathbf{e}_{cd}^{\mathrm{T}} \mu_k = \mathbf{e}_{ab}^{\mathrm{T}} \mu_k \mu_k^{\mathrm{T}} \mathbf{e}_{cd} = (\mu_k \mu_k^{\mathrm{T}})_{ab,cd} \tag{6.63}$$

$$\begin{aligned}
\frac{\partial (\mu_{ij,k}^{\mathrm{T}} \mathbf{\Lambda}_{ij,k} \mu_{ij,k})}{\partial \Lambda_{k,ab,cd}} &= \mu_{ij,k}^{\mathrm{T}} \frac{\partial \mathbf{\Lambda}_{ij,k}}{\partial \Lambda_{k,ab,cd}} \mu_{ij,k} + 2\mu_{ij,k}^{\mathrm{T}} \mathbf{\Lambda}_{ij,k} \frac{\partial \mathbf{\Lambda}^{-1}_{ij,k}}{\partial \Lambda_{k,ab,cd}} (\mathbf{H}_{ij} \mathbf{w}_{ij}^* + \mathbf{\Lambda}_k \mu_k) + 2\mu_{ij,k}^{\mathrm{T}} \frac{\partial \mathbf{\Lambda}_k}{\partial \Lambda_{k,ab,cd}} \mu_k \\
&= (\mu_{ij,k} \mu_{ij,k}^{\mathrm{T}} + 2\mu_{ij,k} \mu_k^{\mathrm{T}})_{ab,cd} - 2\mu_{ij,k}^{\mathrm{T}} \mathbf{\Lambda}_{ij,k} \mathbf{\Lambda}_{ij,k}^{-1} \frac{\partial \mathbf{\Lambda}_{ij,k}}{\partial \Lambda_{k,ab,cd}} \mathbf{\Lambda}_{ij,k}^{-1} (\mathbf{H}_{ij} \mathbf{w}_{ij}^* + \mathbf{\Lambda}_k \mu_k) \qquad (6.64) \\
&= (\mu_{ij,k} \mu_{ij,k}^{\mathrm{T}} + 2\mu_{ij,k} \mu_k^{\mathrm{T}})_{ab,cd} - 2\mu_{ij,k}^{\mathrm{T}} \frac{\partial \mathbf{\Lambda}_{ij,k}}{\partial \Lambda_{k,ab,cd}} \mu_{ij,k} \qquad (6.65) \\
&= (-\mu_{ij,k} \mu_{ij,k}^{\mathrm{T}} + 2\mu_{ij,k} \mu_k^{\mathrm{T}})_{ab,cd} \,. \qquad (6.66)
\end{aligned}$$

Inserting these results into eq. (6.58) yields

$$\frac{\partial}{\partial \Lambda_{k,ab,cd}} \log \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} = \frac{1}{2} \left(\mathbf{\Lambda}_k^{-1} - \mathbf{\Lambda}_{ij,k}^{-1} - (\mu_{ij,k} - \mu_k)(\mu_{ij,k} - \mu_k)^{\mathrm{T}}\right)_{ab,cd} \,. \tag{6.67}$$

Substituting this expression into the equation (6.51) analogous to the derivation of gradient for $\mu_{k,ab}$ yields the equation

$$\nabla_{\mathbf{\Lambda}_k} LL(\mu, \mathbf{\Lambda}, \gamma_k) = \frac{1}{2} \sum_{1 \le i < j \le L} p(k|ij) \left(\mathbf{\Lambda}_k^{-1} - \mathbf{\Lambda}_{ij,k}^{-1} - (\mu_{ij,k} - \mu_k)(\mu_{ij,k} - \mu_k)^{\mathrm{T}}\right) . \tag{6.68}$$

6.5.6 The gradient of the log likelihood with respect to γ_k

With $r_{ij} \in \{0, 1\}$ defining a residue pair in physical contact or not in contact, the mixing weights can be modelled as a softmax function according to eq. (4.6). The derivative of the mixing weights $g_k(r_{ij})$ is:

$$\frac{\partial g_{k'}(r_{ij})}{\partial \gamma_k} = \begin{cases} g_k(r_{ij})(1 - g_k(r_{ij})) & : k' = k \\ g_{k'}(r_{ij}) - g_k(r_{ij}) & : k' \neq k \end{cases} \tag{6.69}$$

The partial derivative of the likelihood function with respect to γ_k is:

$$\frac{\partial}{\partial \gamma_k} LL(\mu, \mathbf{\Lambda}, \gamma_k) = \sum_{1 \leq i < j \leq L} \frac{\sum_{k'=0}^{K} \frac{\partial}{\partial \gamma_k} g_{k'}(r_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(0|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})}}{\sum_{k'=0}^{K} g_{k'}(r_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})}} \tag{6.70}$$

$$= \sum_{1 \leq i < j \leq L} \frac{\sum_{k'=0}^{K} g_{k'}(r_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} \cdot \begin{cases} 1 - g_k(r_{ij}) & \text{if } k' = k \\ -g_k(r_{ij}) & \text{if } k' \neq k \end{cases}}{\sum_{k'=0}^{K} g_{k'}(r_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})}} \tag{6.71}$$

$$= \sum_{1 \leq i < j \leq L} \sum_{k'=0}^{K} p(k'|ij) \begin{cases} 1 - g_k(r_{ij}) & \text{if } k' = k \\ -g_k(r_{ij}) & \text{if } k' \neq k \end{cases} \tag{6.72}$$

$$= \sum_{1 \leq i < j \leq L} p(k|ij) - g_k(r_{ij}) \sum_{k'=0}^{K} p(k'|ij)$$

$$= \sum_{1 \leq i < j \leq L} p(k|ij) - g_k(r_{ij}) \tag{6.73}$$

6.6 Bayesian Statistical Model for Prediction of Protein Residue-Residue Distances

6.6.1 Modelling the dependence of $\mathbf{w}_{ij}$ on distance

It is straightforward to extend the model presented in 4.2 for distances.

The mixture weights $g_k(r_{ij})$ in eq. (4.5) are modelled as softmax over linear functions $\gamma_k(r_{ij})$ (Figure **??**fig:softmax-linear-fct):

$$g_k(r_{ij}) = \frac{\exp \gamma_k(r_{ij})}{\sum_{k'=0}^{K} \exp \gamma_{k'}(r_{ij})}, \tag{6.74}$$

$$\gamma_k(r_{ij}) = -\sum_{k'=0}^{k} \alpha_{k'}(r_{ij} - \rho_{k'}). \tag{6.75}$$

The functions $g_k(r_{ij})$ remain invariant when adding an offset to all $\gamma_k(r_{ij})$. This degeneracy can be removed by setting $\gamma_0(r_{ij}) = 0$ (i.e., $\alpha_0 = 0$ and $\rho_0 = 0$). Further, the components are ordered, $\rho_1 > \ldots > \rho_K$ and it is demanded that $\alpha_k > 0$ for all k. This ensures that for $r_{ij} \to \infty$ we will obtain $g_0(r_{ij}) \to 1$ and hence $p(\mathbf{w}|\mathbf{X}) \to \mathcal{N}(0, \sigma_0^2 \mathbf{I})$.

The parameters ρ_k mark the transition points between the two Gaussian mixture components $k-1$ and k, i.e., the points at which the two components obtain equal weights. This follows from $\gamma_k(r_{ij}) - \gamma_{k-1}(r) = \alpha_t(r_{ij} - \rho_t)$ and hence $\gamma_{k-1}(\rho_k) = \gamma_k(\rho_k)$. A change in ρ_k or α_k only changes the behaviour of $g_{k-1}(r_{ij})$ and $g_k(r_{ij})$ in the transition region around ρ_k. Therefore, this particular definition of $\gamma_k(r_{ij})$

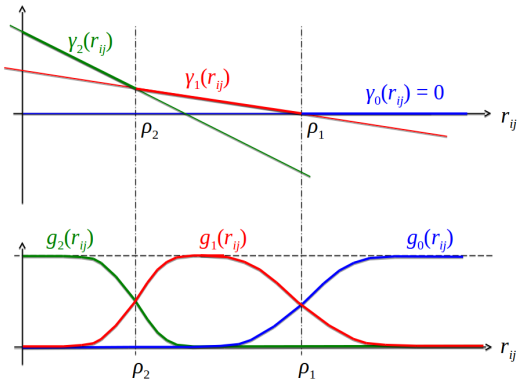


Figure 6.10: The Gaussian mixture coefficients $g_k(r_{ij})$ of $p(\mathbf{w}_{ij}|r_{ij})$ are modelled as softmax over linear functions $\gamma_k(r_{ij})$. ρ_k sets the transition point between neighbouring components $g_{k-1}(r_{ij})$ and $g_k(r_{ij})$, while α_k quantifies the abruptness of the transition between $g_{k-1}(r_{ij})$ and $g_k(r_{ij})$.

makes the parameters α_k and ρ_k as independent of each other as possible, rendering the optimisation of these parameters more efficient.

6.6.2 Training the Hyperparameters ρ_k and α_k for distance-dependent prior

6.7 Training Random Forest Contat Prior

6.7.1 Sequence Derived Features

Given a multiple sequence alignment of a protein family, various sequence features can be derived that have been found to be informative of a residue-residue contact.

In total there are **250** features that can be divided into global, single position and pairwise features and are described in the following sections. If not stated otherwise, *weighted* features have been computed using amino acid counts or amino acid frequencies based on weighted sequences as described in section 6.2.3.

6.7.1.1 Global Features

These features describe alignment characteristics. Every pair of residues (i, j) from the same protein will be attributed the same feature.

Table 6.1: Features characterizing the total alignment

Feature	Description	No. Features per residue pair (i, j)
L	log of protein length L	1

Feature	Description	No. Features per residue pair (i, j)
N	number of sequences N	1
Neff	number of effective sequences Neff computed as the sum over sequence weights (see section 6.2.3)	1
gaps	average percentage of gaps over all positions	1
diversity	$\frac{\sqrt{N}}{L}$, N=number of sequences, L=protein length	1
amino acid composition	weighted amino acid frequencies in alignment	20
secondary structure prediction	average three state propensities PSIPRED (v4.0)[134]	3
secondary structure prediction	average three state propensities Netsurfp (v1.0)[126]	3
contact prior protein length	simple contact predictor based on expected number of contacts per protein with respect to protein length (see next subsection 6.7.1.4)	1

There are in total **32** global alignment features.

6.7.1.2 Single Position Features

These features describe characteristics of a single alignment column. Every residue pair (i, j) will be described by two features, once for each position.

Table 6.2: Single Position Sequence Features

Feature	Description	No. Features per residue pair (i, j)
shannon entropy (20 states)	$-\sum_{a=1}^{20} p_a \log p_a$	2
shannon entropy (21 states)	$-\sum_{a=1}^{21} p_a \log p_a$	2
kullback leibler divergence	between weighted observed and background amino acid frequencies [135]	2
jennson shannon divergence	between weighted observed and background amino acid frequencies [135]	2
PSSM	log odds ratio of weighted observed and background amino acid frequencies [135]	40

Feature	Description	No. Features per residue pair (i, j)
secondary structure prediction	three state propensities PSIPRED (v4.0) [134]	6
secondary structure prediction	three state propensities Netsurfp (v1.0) [126]	6
solvent accessibility prediction	RSA and RSA Z-score Netsurfp (v1.0) [126]	4
relative position in sequence	$\frac{i}{L}$ for a protien of length L	2
number of ungapped sequences	$\sum_n w_n I(x_{ni} \neq 20)$ for sequences x_n and sequence weights w_n	2
percentage of gaps	$\frac{\sum_n w_n I(x_{ni}=20)}{N_{\text{eff}}}$ for sequences x_n and sequence weights w_n	2
Average physico-chemical properties	Atchley Factors 1-5 [136]	10
Average physico-chemical properties	Polarity accordign to Grantham, 1974. Data taken from AAindex Database [137].	2
Average physico-chemical properties	Polarity according to Zimmermann et al., 1986. Data taken from AAindex Database [137].	2
Average physico-chemical properties	Isoelectric point according to Zimmermann et al., 1968. Data taken from AAindex Database [137].	2
Average physico-chemical properties	Hydrophobicity scale according to Wimley & White, 1996. Data taken from UCSF Chimera [138].	2
Average physico-chemical properties	Hydrophobicity index according to Kyte & Doolittle, 1982. Data taken from AAindex Database [137].	2
Average physico-chemical properties	Hydrophobicity according to Cornette [139].	2
Average physico-chemical properties	Bulkiness according to Zimmerman et al., 1968. Data taken from AAindex Database [137].	2
Average physico-chemical properties	Average volumes of residues according to Pontius et al., 1996. Data taken from AAindex Database [137].	2

There are in total **96** single sequence features.

Additionally, all single features will be computed within a window of size 5. The window feature for center residue i will be computed as the mean feature over residues $[i-2, \ldots, i, \ldots, i+2]$. Whenever the window extends the range of the sequence (for $i<2$ and $i>(L-2)$), the window feature will be computed only for valid sequence positions. This results in additional **96** window features.

6.7.1.3 Pairwise Features

These features are computed for every pair of columns (i, j) in the alignment with $i < j$.

Table 6.3: Pairwise Sequence Features

Feature	Description	No. Features per residue pair (i, j)
sequence separation	$j - i$	1
gaps	pairwise percentage of gaps using weighted sequences	1
number of ungapped sequences	$\sum_n w_n I(x_{ni} \neq 20, x_{nj} \neq 20)$ for sequences x_n and sequence weights w_n	1
correlation physico-chemical features	pairwise correlation of all physico-chemical properties listed in 6.7.1.2	13
pairwise potential	Average quasi-chemical energy of interactions in an average buried environment. Data taken from AAindex Database [137].	1
pairwise potential	Average quasi-chemical energy of transfer of amino acids from water to the protein environment. Data taken from AAindex Database [137].	1
pairwise potential	Average general contact potential by Li&Fang [52]	1
pairwise potential	Average statistical potential from residue pairs in beta-sheets by Zhu&Braun [140]	1
joint_shannon_entropy (20 state)	$-\sum_{a=1}^{20}\sum_{b=1}^{20} p(a,b)\log p(a,b)$	1
joint_shannon_entropy (21 state)	$-\sum_{a=1}^{21}\sum_{b=1}^{21} p(a,b)\log p(a,b)$	1
mutual information (MI)	several variants: MI with pseudo-counts, MI with pseudo-counts + APC, normalized MI	3

Figure 6.11: Observed number of contacts per residue has a non-linear relationship with protein length. Distribution is shown for several thresholds of sequence separation.

Feature	Description	No. Features per residue pair (i, j)
OMES	according to Fodor&Aldrich [141] with and without APC	2

There are in total **26** pairwise sequence features.

6.7.1.4 Protein length dependent Contact Prior

The average number of contats per residue, computed as the observed number of contacts divided by protein length L, has a non-linear relationship with protein length L as can be seen in Figure 6.11.

In log space, the average number of contats per residue can be fitted with a linear regression (see Figure 6.12) and yields the following functions:

- $f(L) = 1.556 + 0.596 \log(L)$ for sequence separation of 0 positions
- $f(L) = -1.273 + 0.59 \log(L)$ for sequence separation of 8 positions
- $f(L) = -1.567 + 0.615 \log(L)$ for sequence separation of 12 positions
- $f(L) = -2.0 + 0.624 \log(L)$ for sequence separation of 24 positions

A simple contact predictor can be formulated as the ratio of the expected number of contacts per residue, given by $f(L)$, and the possible number of contacts per residue which is $L - 1$,

$$p(r_{ij} = 1|L) = \frac{f(L)}{L - 1} ,$$

with $r_{ij} = 1$ representing a contact between residue i and j.

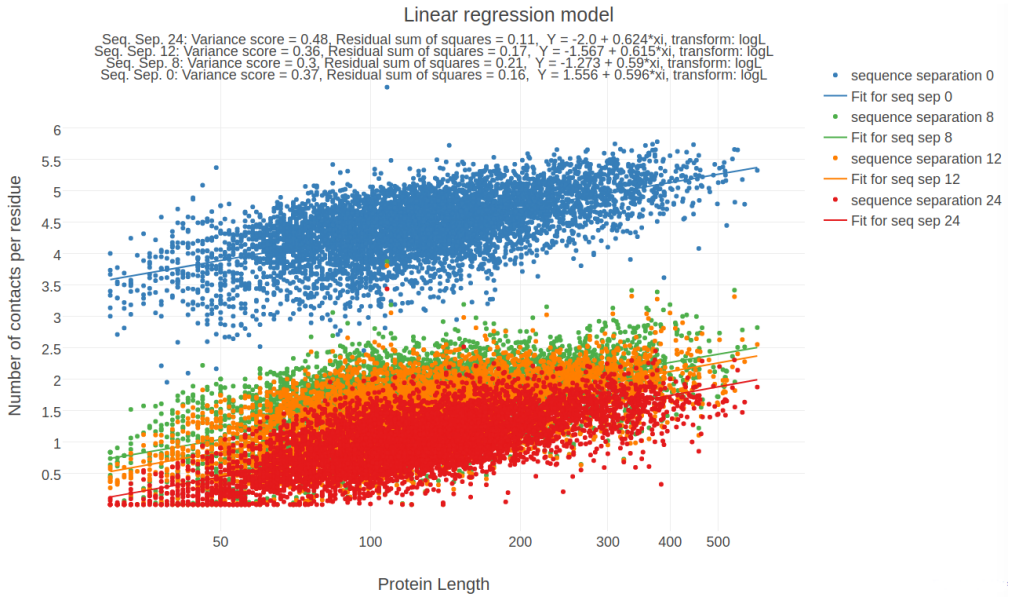


Figure 6.12: (ref:caption-avg-nr-contacts-per-residue-vs-log-protein-length-linfit)

(ref:caption-avg-nr-contacts-per-residue-vs-log-protein-length-linfit) Linear regression fits for average number of contats per residue on logarithm of protein length. Distribution and linear regression fits are shown for different sequence separation thresholds.

6.7.2 Hyperparameter Optimization for Random Forest Prior

There are several hyperparameters in a random forest model that need to be tuned to achieve best balance between predictive power and runtime. While more trees in the random forest generally improve performance of the model, they will slow down training and prediction. A crucial hyperparamter is the number of features that is randomly selected for a split at each node in a tree [142]. Stochasticity introduced by the random selection of features is a key characteristic of random forests as it reduces correlation between the trees and thus the variance of the predictor. Selecting many features typically increases performance as more options can be considered for each split, but at the same time increases risk of overfitting and decreases speed of the algorithm. In general, random forests are robust to overfitting, as long as there are enough trees in the ensemble and the selection of features for splitting a node introduces sufficient stochasticity. Overfitting can furthermore be prevented by restricting the depth of the trees, which is known as pruning or by enforcing a minimal node size with respect to the number of features per node. A positive side-effect of taking these measures is a speedup of the algorithm. [143]

In the following, I use 5-fold cross-validation to identify the optimal architecture of the random forest. I used the module RandomForestClassifier in the Python package `sklearn (v. 0.19)` [144] and trained the models on sequence features extracted from MSAs as described in section 6.7.1. Single position features are

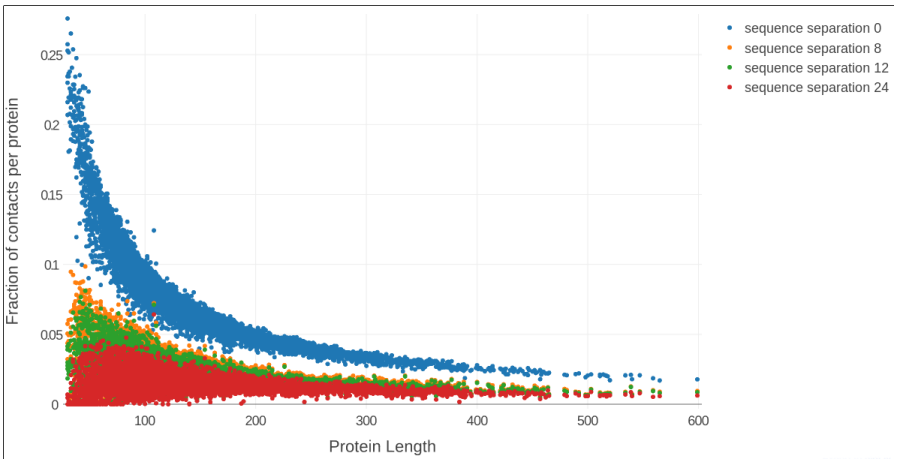


Figure 6.13: Fraction of contacts among all possible contacts ($\frac{L(L-1)}{2}$) in a protein against protein length L. The distribution has a non-linear relationship. At a sequence separation threshold >8 positions the fraction of contacts for intermediate size proteins with length >100 is approximately 2%.

computed with a window of size five as described in section 6.7.1.2.

Proteins constitute highly imbalanced datasets with respect to the number of residue pairs that form and form not physical contacts. As can be seen in Figure 6.13, depending on the enforced sequence separation threshold the percentage of contacts varies between approximately 1% and 5%.

Most studies applying machine learning algorithms to the problem of predicting residue-residue contacts, chose the standard approach of rebalancing the data set by undersampling of the majority class.

Study	Proportion of Contacts	Proportion of Non-contacts
Wu et al. (2008) [51]	1	4
Li et al. (2011) [52]	1	1, 2
Wang et al. (2011) [53]	1	4
DiLena et al. (2012) [61]	1	˜4
Wang et al. (2013) [54]	1	˜4

I followed the same strategy and undersampled residue pairs that are not physical contacts with a proportion of contacts to non-contacts of 1:5. The training set is comprised of 50.000 residue pairs $< 8\mathbf{r}A$ ("contacts" ") and 250.000 residue pairs $> 8\mathbf{r}A$ ("non-contacts") so that each of the five cross-validation models will be trained on 40.000 contacts and 200.000 non-contacts. As the training set has been undersampled for non-contacts, it is not representative of real world proteins and the models should be validated on a more realistic validation set. Each of the five models is therefore cross-validated on an own independent dataset of residue pairs extracted from 40 proteins by means of the standard contact prediction benchmark (mean precision against top ranked contacts).

First I assessed performance of models for combinations of the parameter *n_estimators*, defining the number of trees in the forest and the parameter *max_depth* defining the maximum depth of the trees:

- $n\_estimators \in \{100, 500, 1000\}$
- $max\_depth \in \{10, 100, 1000\}$

Figure 6.14 shows that using 1000 trees and a maximum tree depth of 100 or 1000

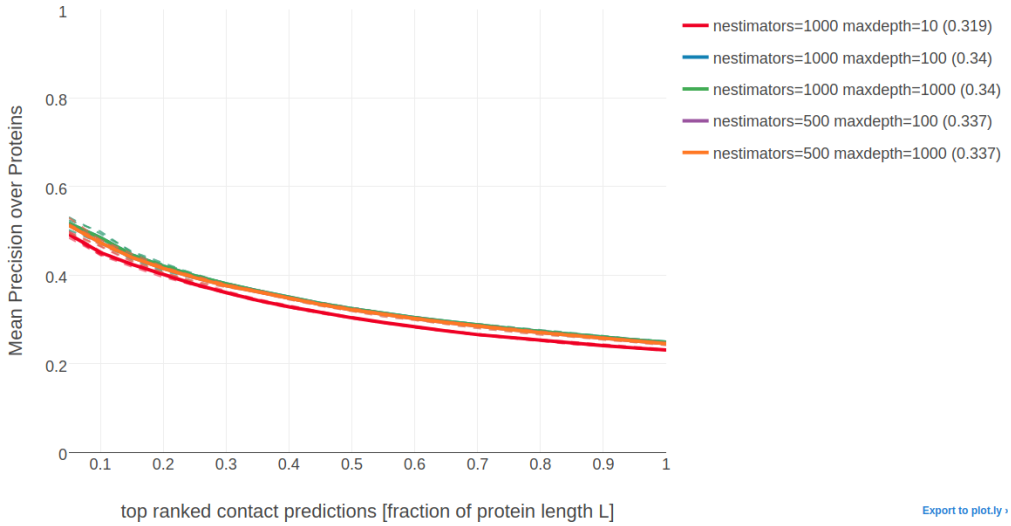


Figure 6.14: Mean precision over 200 proteins against highest scoring contact predictions from random forest models for different settings of *n_estimators* and *max_depth*. Dashed lines show the performance of models that have been learned on the five different subsets of training data. Solid lines give the mean precision over the five models that were trained with the same parameter settings. Only those models are shown that yielded the five best mean precision values (given in parantheses in the legend). Random forest models with 1000 trees and *max_depth*=10 or *max_depth*=100 perform nearly identical as well as models with 500 trees and *max_depth*=10 or *max_depth*=100.

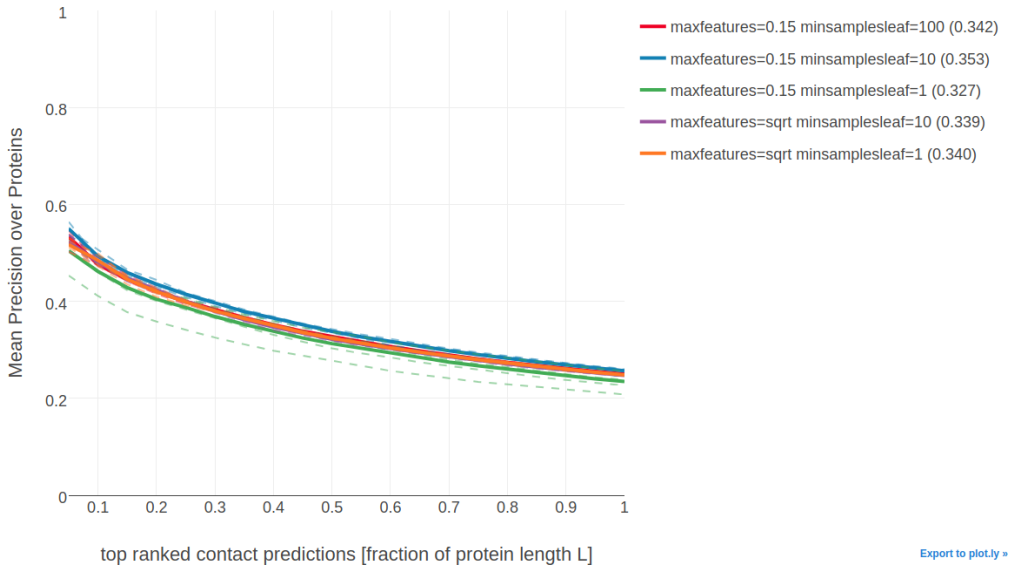


Figure 6.15: Mean precision over 200 proteins against highest scoring contact predictions from random forest models with different settings of *min_samples_leaf* and *max_features*. Dashed lines show the performance of models with the same parameter setting that have been learned on the five different subsets of training data. Solid lines give the mean precision over these five models. Only those models are shown that yielded the five best mean precision values (given in parantheses in the legend).

`max_features = 0.15` and `min_samples_leaf = 10` for further analysis. Tuning the hyperparameters in a different order or on a larger dataset gives similar results.

In a next step I assessed dataset specific settings, such as the window size over which single positions features will be computed, the distance threshold to define non-contacts and the optimal proportions of contacts and non-contacts in the training set. I used the previously identified settings of random forest hyperparameters (`n_estimators=1000, min_samples_leaf=10, max_depth=100, max_features=0.15`).

- ratio of non-contacts/contacts $\in \{2, 5, 10, 20\}$ within a fixed total dataset size of 300 000 residue pairs
- window size: $\in \{5, 7, 9, 11\}$
- non-contact threshold $\in \{8, 15, 20\}$

As can be seen in appendix E.1 and E.2, the default choice of using a window size of five positions and the non-contact threshold of $8\mathring{A}$ proves to be the optimal setting. Furthermore, using five-times as many non-contacts as contacts in the training set results in highest mean precision as can be seen in appendix E.3. These estimates might be biased in a way since the random forest hyperparameters have been optimized on a dataset using exactly these optimal settings.

6.7.3 Feature Selection

Many features obtain low *Gini importance* scores and can most likely be removed from the data set which will also reduce model complexity. It has been found, that prediction performance might even increase after removing the most irrelevant feaures [123]. For example, during the development of *EPSILON-CP*, a deep neural network method for contact prediction, the authors performed feature selection using boosted trees. By removing 75% of the most non-informative features (mostly attributed to amino acid composition), the performance of their predictor increased slightly [69]. Other studies have also emphasized the importance of feature selection to improve performance and reduce model complexity [50,52].

I therefore developed a feature selection pipeline that retrains the random forest model on subsets of features. The subsets are composed of those features having *Gini importance* larger than the $\{10, 30, 50, 70, 90\}$-percentile of the distribution obtained by training a model on all features. Performance is then evaluated by means of the standard contact prediction benchmark (mean precision against top ranked contacts) for the models trained on these subsets of features.

6.7.4 Using Pseudo-likelihood Coevolution Score as Additional Feature

In addition to the 250 sequence derived features, the pseudo-likelihood contact score (L2norm + APC) is used as a feature. The random forest was trained on 100.000 residue pairs in contact ($\Delta C_\beta < 8\mathbf{r}A$) and 500.000 residue pairs not in

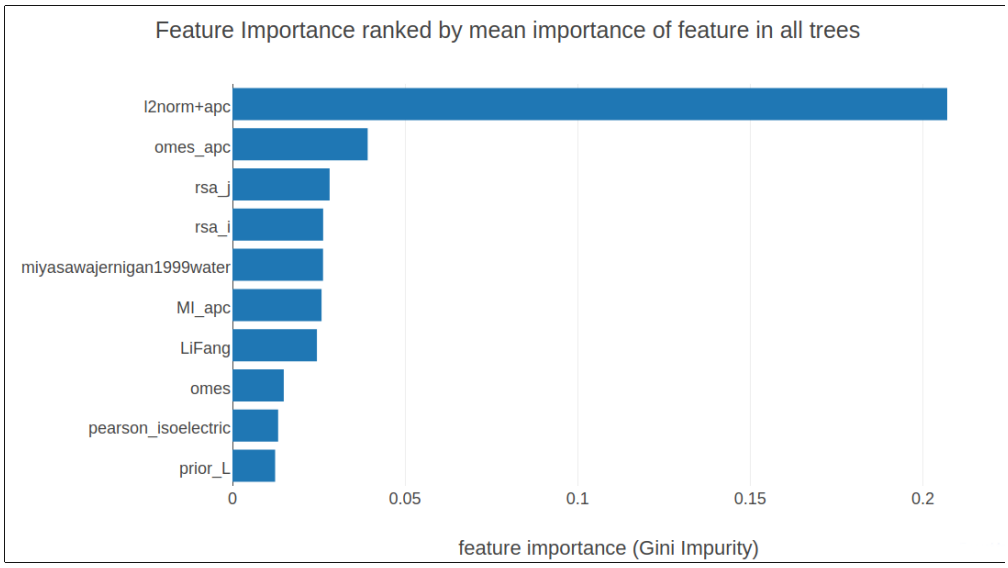


Figure 6.16: Most important features in the random forest model. Features are ranked according to Gini importance which is the mean decrease in Gini impurity over all splits and all trees in the forest.

contact ($\Delta C_\beta > 8\mathbf{r}A$) using the cross-validated hyperparameters as described in the last section.

The pseudo-likelihood contact score comprises by far the most important feature as can be seen in the following Figure 6.16.

Training the model only on the 26 most important features improves precision of the model compared to using the full feature set as is illustrated in the following figure 6.16.

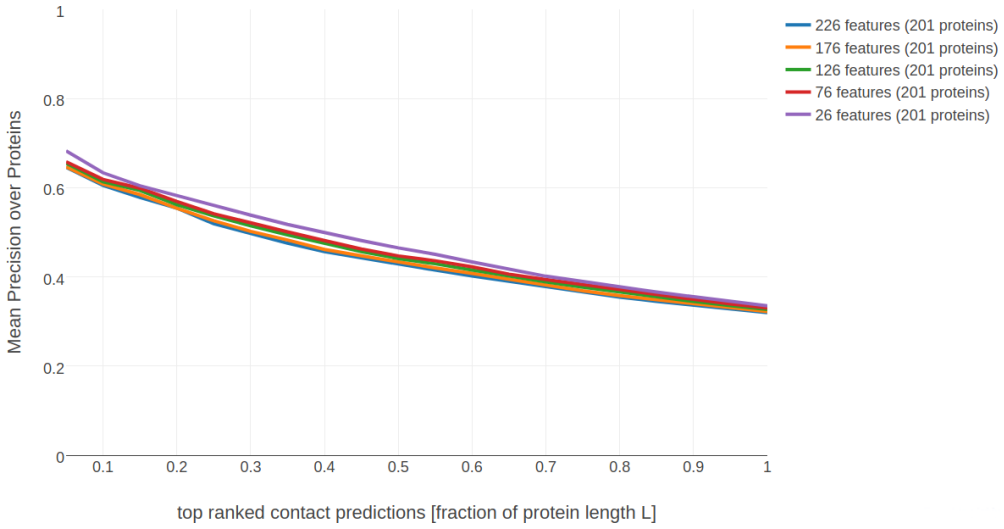


Figure 6.17: Mean precision over proteins in testset for the top ranked contacts for variaous random forest models trained on subsets of features. Subsets of features have been selected as described in section 6.7.3. Learning a random forest model on the 26 most important features yields the best model with respect to precision.

A

Abbreviations

APC Avarage Product Correction

CASP Critical Assessment of protein Structure Prediction

CD Contrastive Divergence

DCA Direct Coupling Analysis

DI Direct Information

EM electron microscopy

IDP intrinsically disordered proteins

MAP Maximum a posteriori

MCMC Markov Chain Monte Carlo

MI mutual information

ML Maximum-Likelihood

MLE Maximum-Likelihood Estimate

MRF Markov-Random Field

MSA Multiple Sequence Alignment

Neff Number of effective sequences

PCD Persistent Contrastive Divergence

PDB protein data bank

SGD stochastic gradient descent

A.1 Amino Acid Alphabet

One-letter Abbreviation	Three-letter Abbreviation	Amino Acid	One-letter Abbreviation	Three-letter Abbreviation	Amino Acid
A	Ala	**A**lanine	M	Met	**M**ethionine
C	Cys	**C**ysteine	N	Asn	Asparagi**N**e
D	Asp	Aspartic Aci**D**	P	Pro	**P**roline
E	Glu	Glutamic Acid	Q	Gln	Glutamine
F	Phe	Phenylalanine	R	Arg	A**R**ginine
G	Gly	**G**lycine	S	Ser	**S**erine
H	His	**H**istidine	T	Thr	**T**hreonine
I	Ile	**I**soleucine	V	Val	**V**aline
K	Lys	Lysine	T	Trp	**T**ryptophan
L	Leu	**L**eucine	Y	Tyr	T**Y**rosine

B

Dataset Properties

The following figures display various statistics about the dataset used throughout this thesis. See section 6.1 for information on how this dataset has been generated.

B.1 Alignment Diversity

B.2 Proportion of Gaps in Alignment

B.3 Alignment Size (number of sequences)

B.4 Protein Length

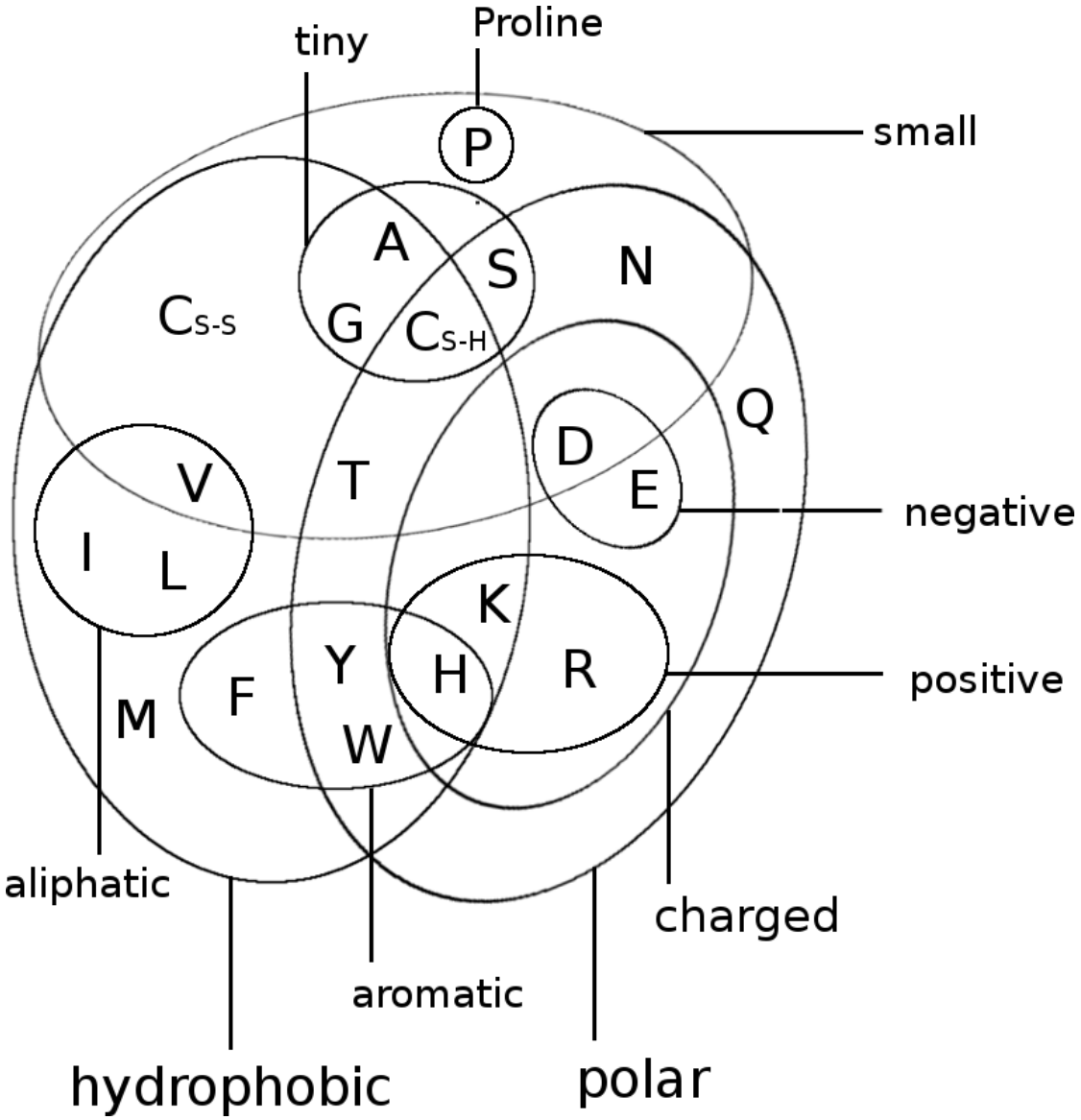


Figure B.1: Distribution of alignment diversity ($= \sqrt{(\frac{N}{L})}$) in the dataset an its ten subsets.

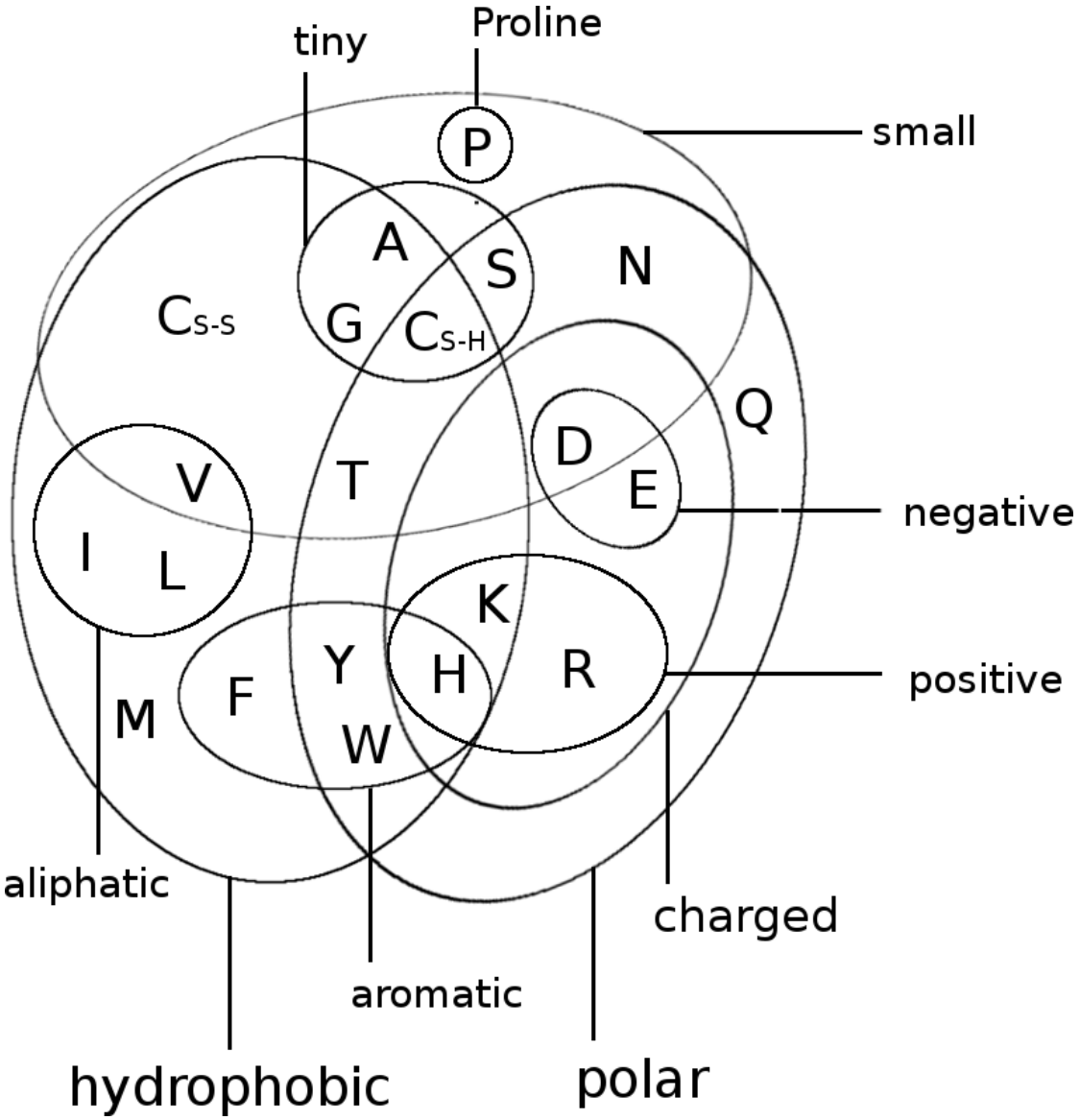


Figure B.2: Distribution of gap percentage of alignments in the dataset an its ten subsets.

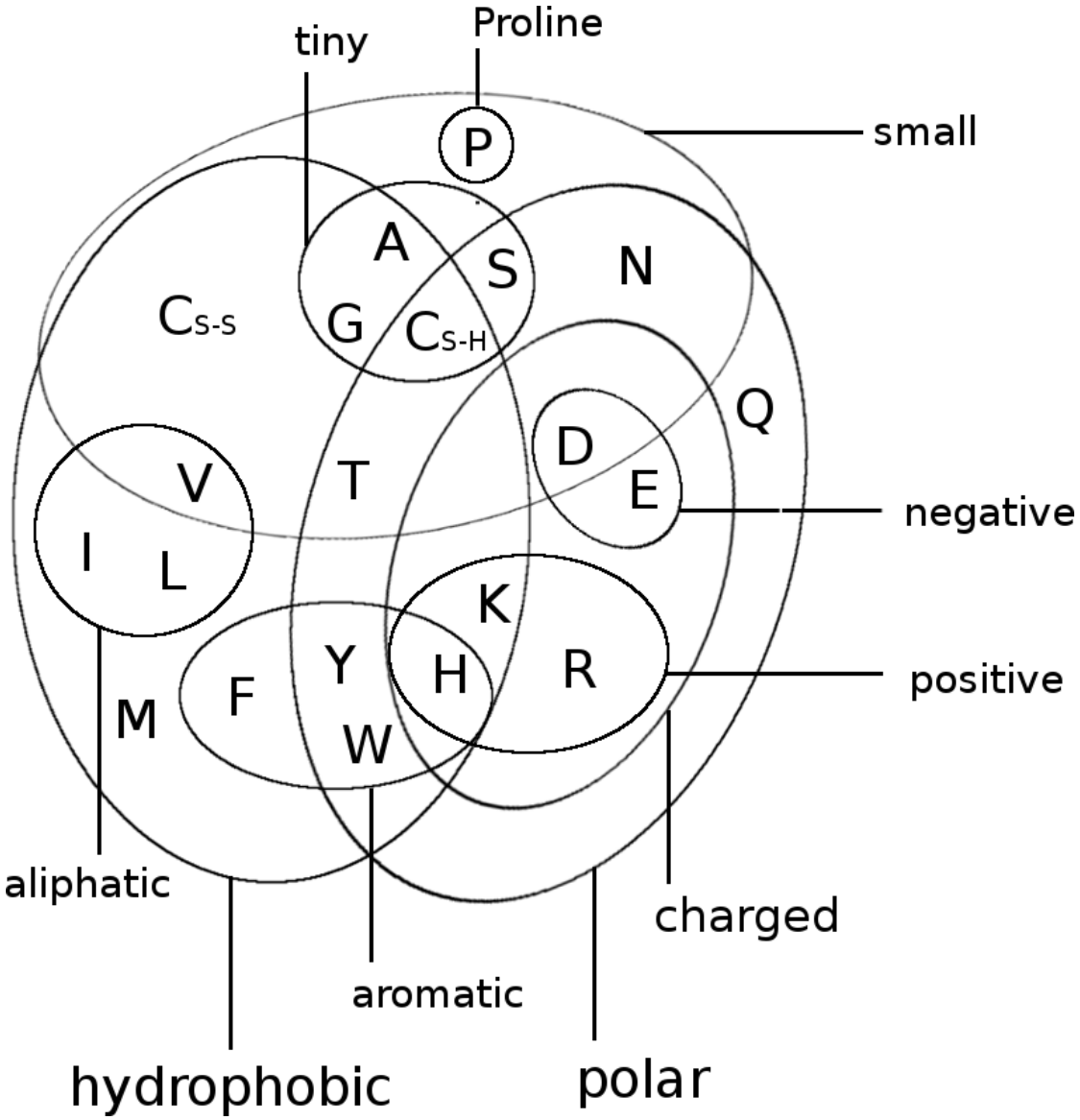


Figure B.3: Distribution of alignment size (number of sequences N) in the dataset an its ten subsets.

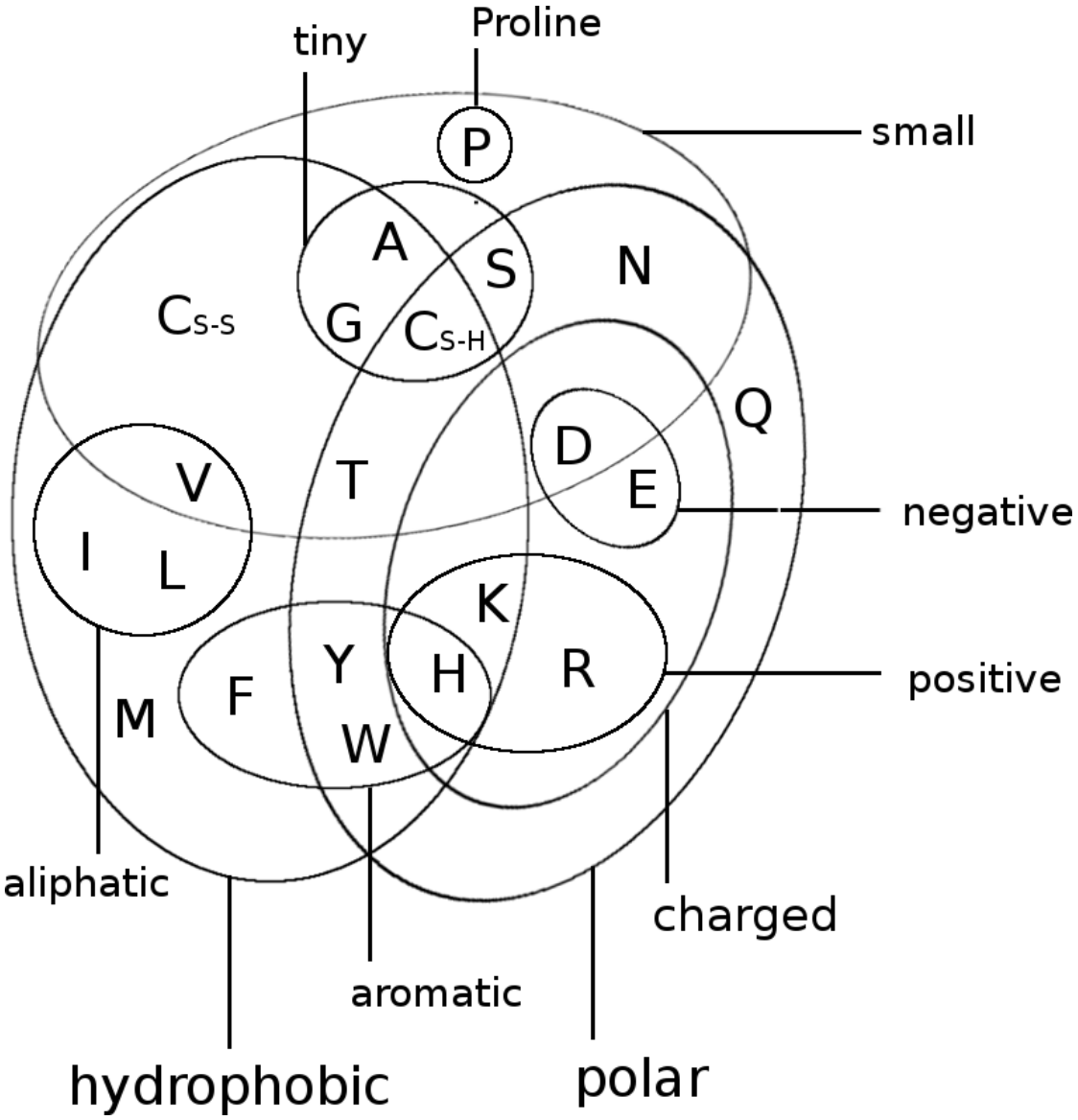


Figure B.4: Distribution of protein length L in the dataset an its ten subsets.

C

Amino Acid Interaction Preferences Reflected in Coupling Matrices

C.1 Pi-Cation interactions

Figure C.1 shows a Tyrosine and a Lysine residue forming a cation-π interaction in protein 2ayd. The corresponding coupling matrix in figure C.2 reflects the strong interaction preference.

C.2 Disulfide Bonds

Figure C.3 shows two cysteine residues forming a covalent disulfide bond in protein 1alu. The corresponding coupling matrix in figure C.4 reflects the strong interaction preference of cysteines.

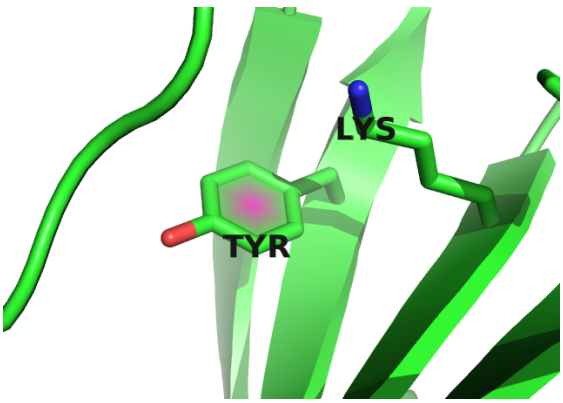


Figure C.1: Tyrosing (residue 37) and Lysine (residue 48) forming a cation-π interaction in protein 2ayd.

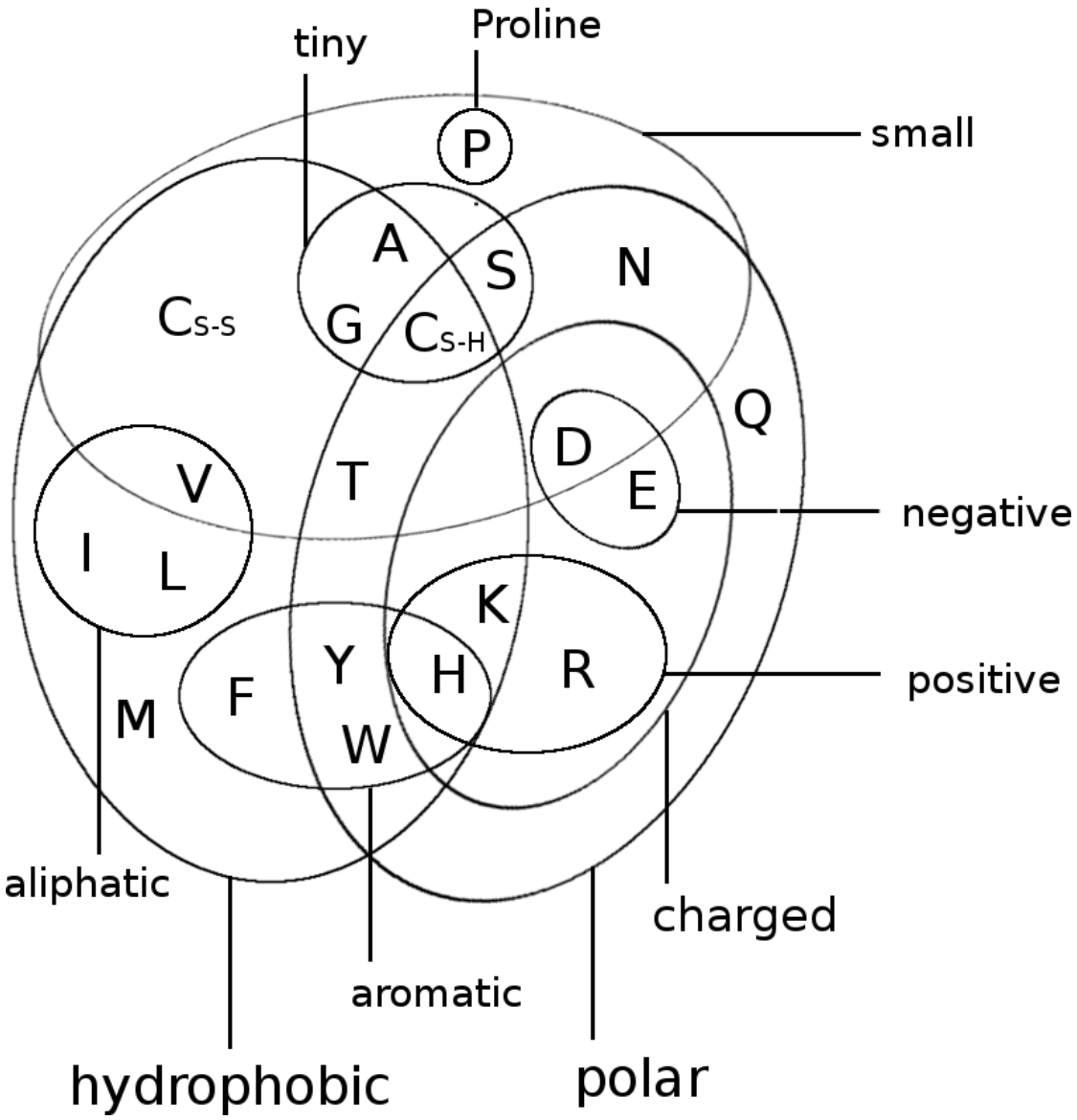


Figure C.2: Coupling Matrix for residue pair i=37 and j=48 of PDB 2ayd chain A domain 1. Size of the bubbles represents coupling strength and color represents the direction of coupling: red = positive coupling, blue = negative coupling. Bars at the x-axis represent single potentials for residue i=37 and bars at the y-axis represent single potentials for residue j=48. Height of the bars represents potential strength and color represents positive (red) and negative (blue) values.

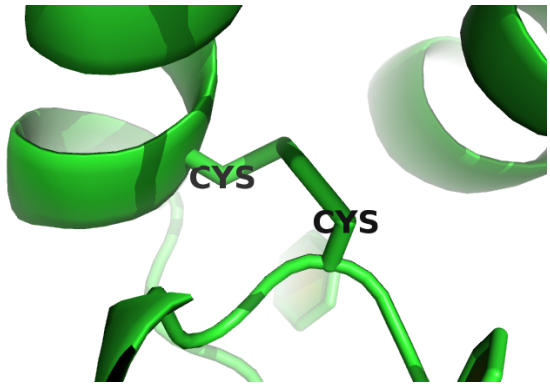


Figure C.3: Two cystein residues (residues 54 and 64) forming a covalent disulfide bond in protein 1alu.

C.3 Aromatic-Proline Interactions

Figure @ref(fig:coupling-matrix-aromatic-proline-pymol)shows a proline and a tryptophan residue forming such a CH/π interaction in protein 1aol. The corresponding coupling matrix in figure C.6 reflects this interaction with strong positive coupling between proline and tryptophan.

C.4 Network-like structure of aromatic residues

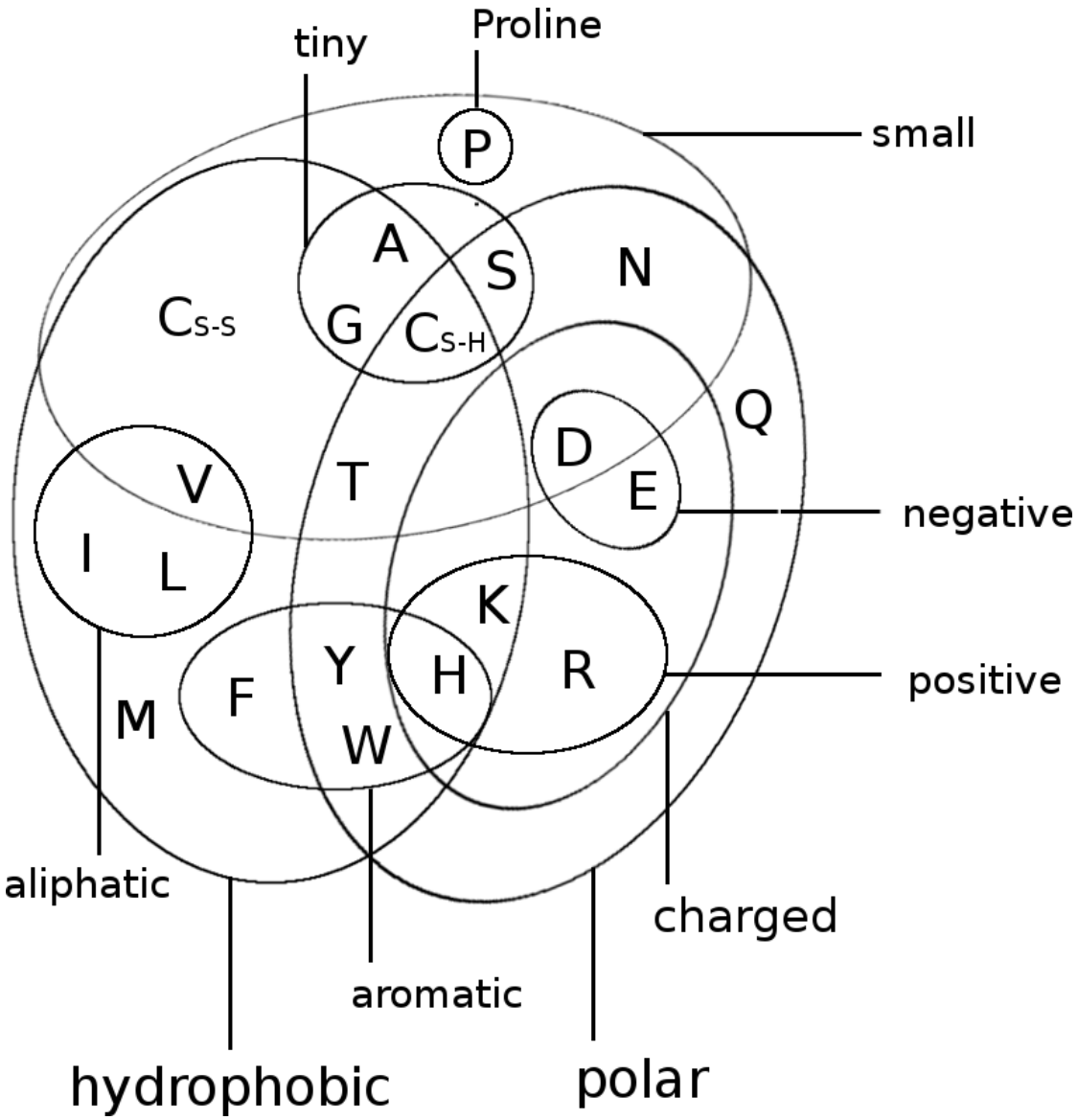


Figure C.4: Coupling Matrix for residue pair i=54 and j=64 of PDB 1alu chain A. Size of the bubbles represents coupling strength and color represents the direction of coupling: red = positive coupling, blue = negative coupling. Bars at the x-axis represent single potentials for residue i=54 and bars at the y-axis represent single potentials for residue j=64. Height of the bars represents potential strength and color represents positive (red) and negative (blue) values.

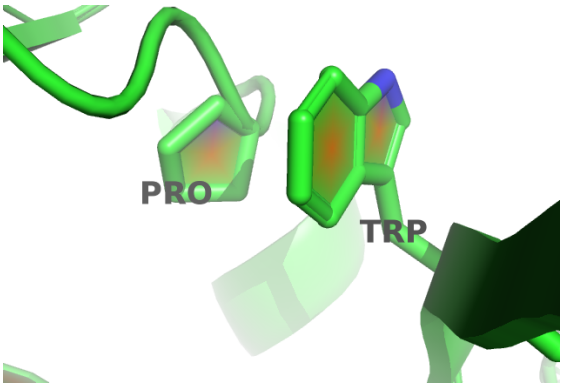


Figure C.5: Proline and tryptophan (residues 17 and 34) stacked on top of each otherengaging in a CH/π interaction in protein 1alu.

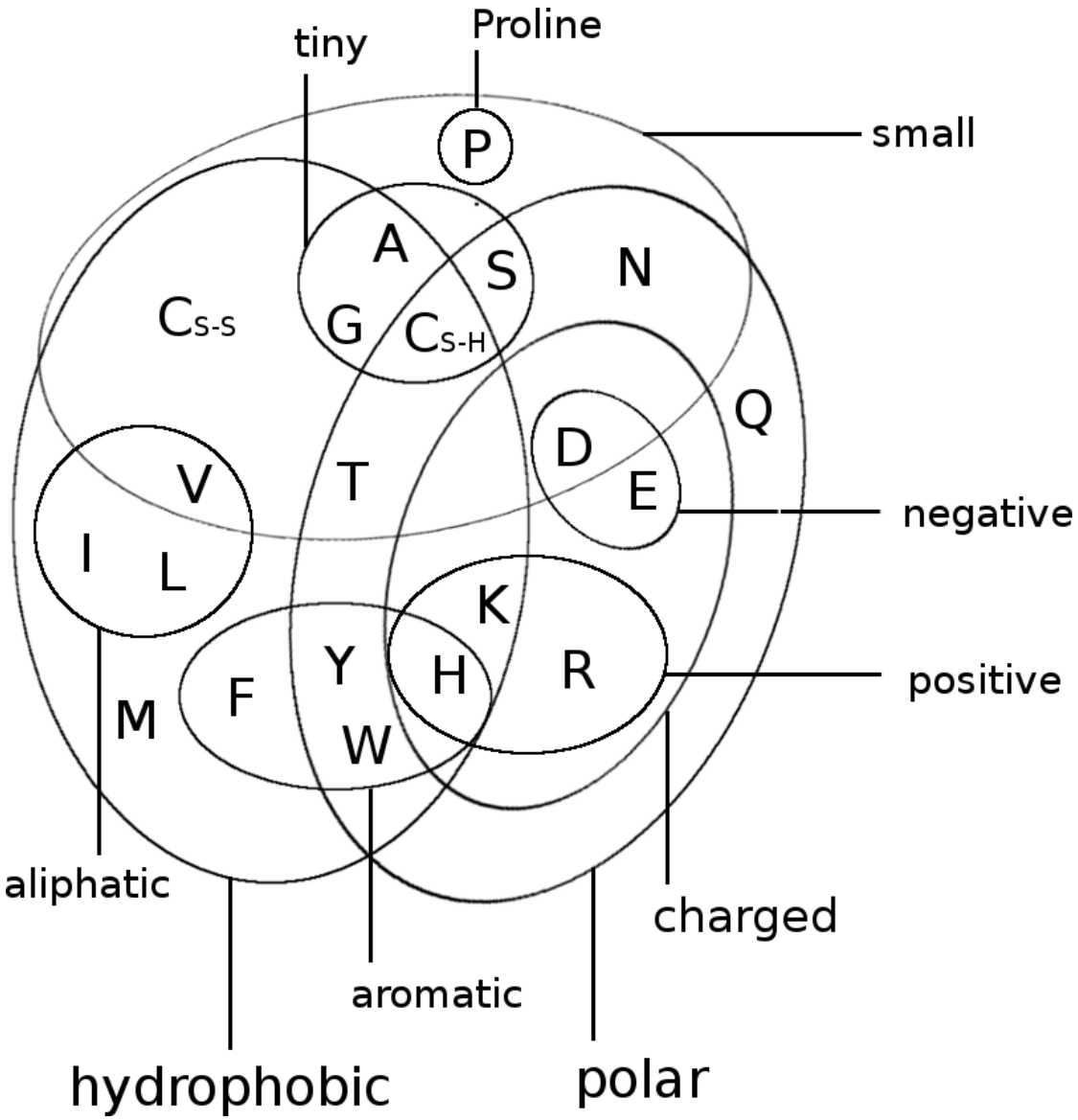


Figure C.6: Coupling Matrix for residue pair i=17 and j=34 of PDB 1aol chain A. Size of the bubbles represents coupling strength and color represents the direction of coupling: red = positive coupling, blue = negative coupling. Bars at the x-axis represent single potentials for residue i=17 and bars at the y-axis represent single potentials for residue j=34. Height of the bars represents potential strength and color represents positive (red) and negative (blue) values.

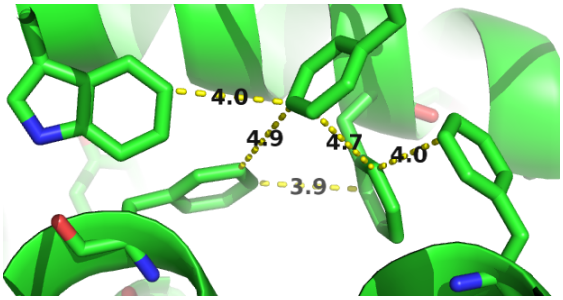


Figure C.7: Network-like structure of aromatic residues in the protein core. 80% of aromatic residues are involved in such networks that are important for protein stability [13].

D

Optimizing Full Likelihood with Gradient Descent

D.1 Divergence of objective function for big learning rates and Neff values

When using large learning rates with alignments comprising many sequences (=large Neff), the optimization diverges.

D.2 Number of iterations for different learning rates

D.3 Number of iterations for different learning rate schedules and fixed initial learning rate α_0 = 1e-4

(ref:caption-full-likelihood-opt-numit_lin_learning-rate-schedule) Distribution of the number of iterations until convergence for gradient descent optimizations of the full likelihood using different decay rates with a default learning rate schedule. Initial learning rate α_0 is fixed to 1e-4 and maximum number of iterations is set to 5000. The learning rate is decreased according to $\alpha = \alpha_0/(1 + \gamma \cdot t)$ with t being the iteration number and γ the decay rate and its value is given after the underscore in the legend names.

(ref:caption-full-likelihood-opt-numit_sig_learning-rate-schedule) Distribution of the number of iterations until convergence for gradient descent optimizations of the full likelihood using different decay rates with a sigmoidal learning rate schedule. Initial learning rate α_0 is fixed to 1e-4 and maximum number of iterations is

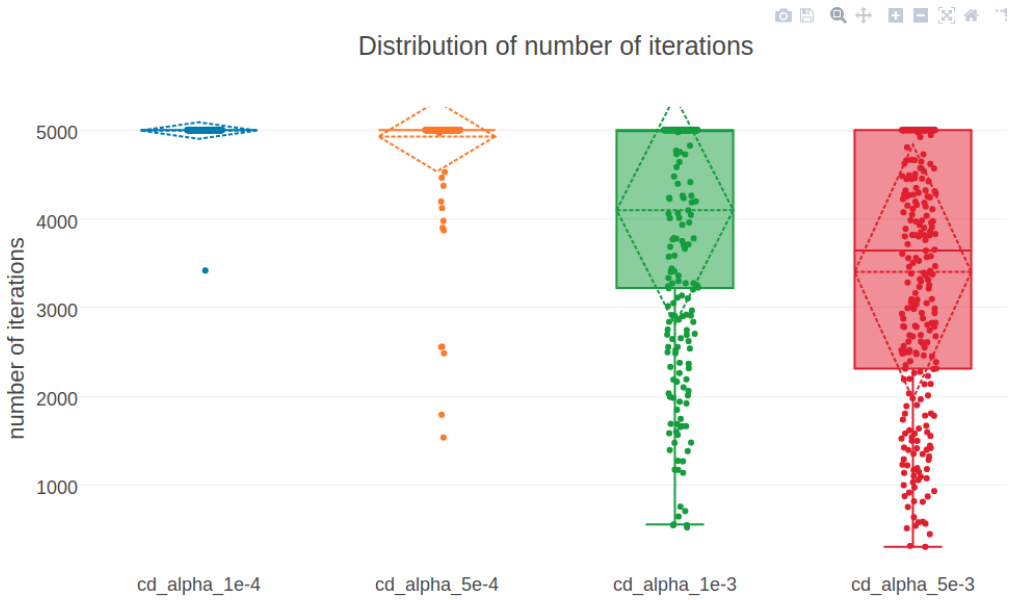


Figure D.1: Distribution of the number of iterations until convergence for gradient descent optimizations of the full likelihood using different learning rates. The learning rate is decreased according to $\alpha = \alpha_0/(1 + 0.01 \cdot t)$ with t being the iteration number and the maximum number of iterations is set to 5000. *cd_alpha-1e-4*: using an initial learning rate of 1e-4. *cd_alpha-5e-4*: using an initial learning rate of 5e-4. *cd_alpha-1e-3*: using an initial learning rate of 1e-3. *cd_alpha-5e-3*: using an initial learning rate of 5e-3.

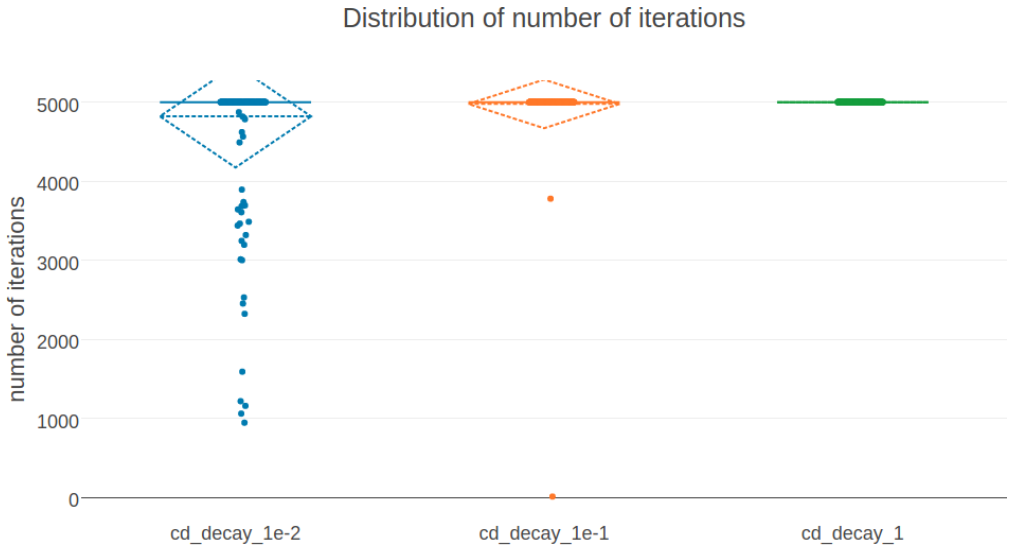


Figure D.2: (ref:caption-full-likelihood-opt-numit-lin-learning-rate-schedule)

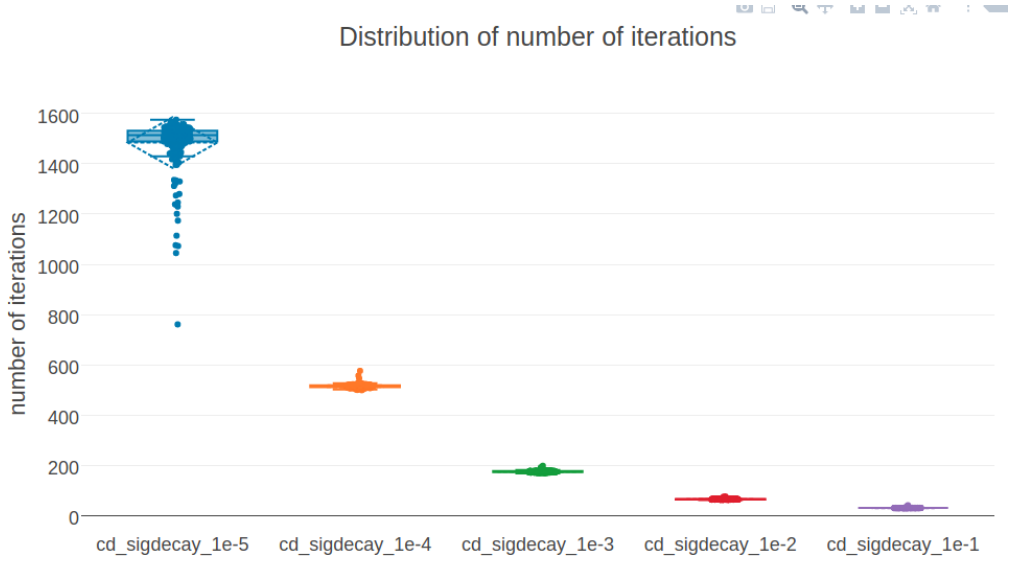


Figure D.3: (ref:caption-full-likelihood-opt-numit-sig-learning-rate-schedule)

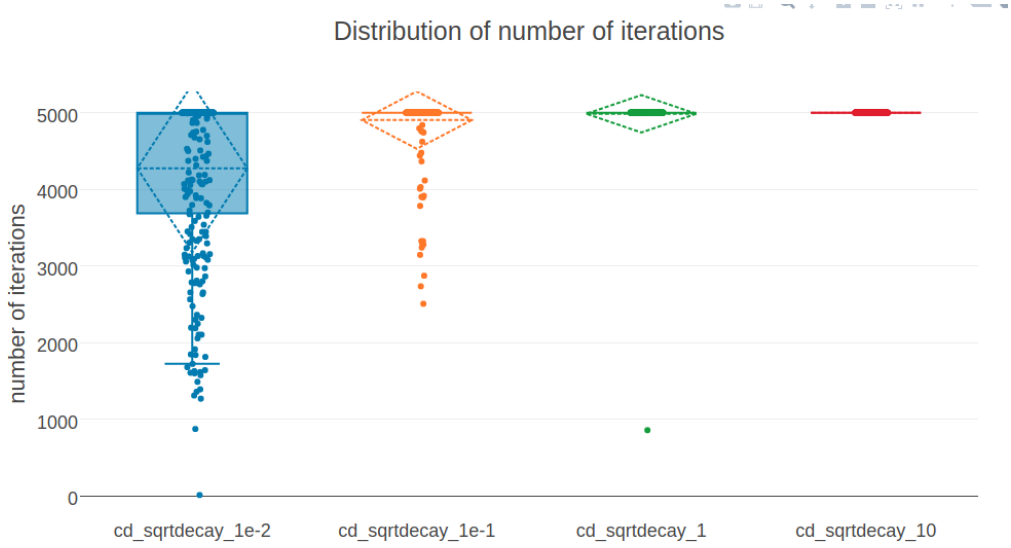


Figure D.4: (ref:caption-full-likelihood-opt-numit-sqrt-learning-rate-schedule)

set to 5000. The learning rate is decreased according to $\alpha_{t+1} = \alpha_t/(1+\gamma \cdot t)$ with t being the iteration number and γ the decay rate and its value is given after the underscore in the legend names.

(ref:caption-full-likelihood-opt-numit_sqrt_learning-rate-schedule) Distribution of the number of iterations until convergence for gradient descent optimizations of the full likelihood using different decay rates with a square root learning rate schedule. Initial learning rate α_0 is fixed to 1e-4 and maximum number of iterations is set to 5000. The learning rate is decreased according to $\alpha_{t+1} = \alpha_t/(1+\gamma \cdot t)$ with t being the iteration number and γ the decay rate and its value is given after the underscore in the legend names.

E

Training of the Random Forest Contact Prior

E.1 Evaluating window size with 5-fold Cross-validation

E.2 Evaluating non-contact threshold with 5-fold Cross-validation

E.3 Evaluating ratio of non-contacts and contacts in the training set with 5-fold Cross-validation

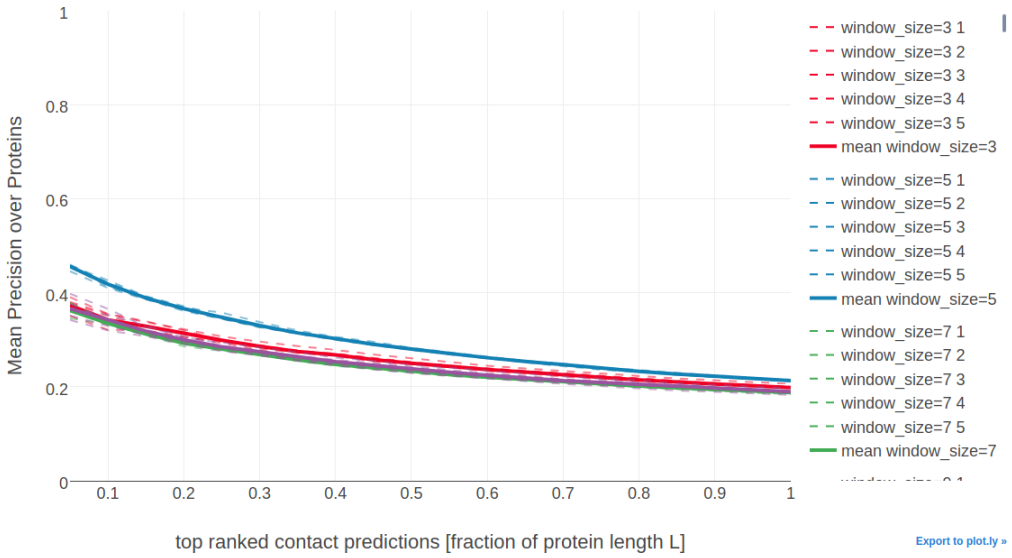


Figure E.1: Mean precision over validation set of 200 proteins for top ranked contact predictions for different choices of window size for single position features. Dashed lines represent the models trained on four subsets of the training data according to the 5-fold cross-validation scheme. Solid lines represent the mean over the five cross-validation models.

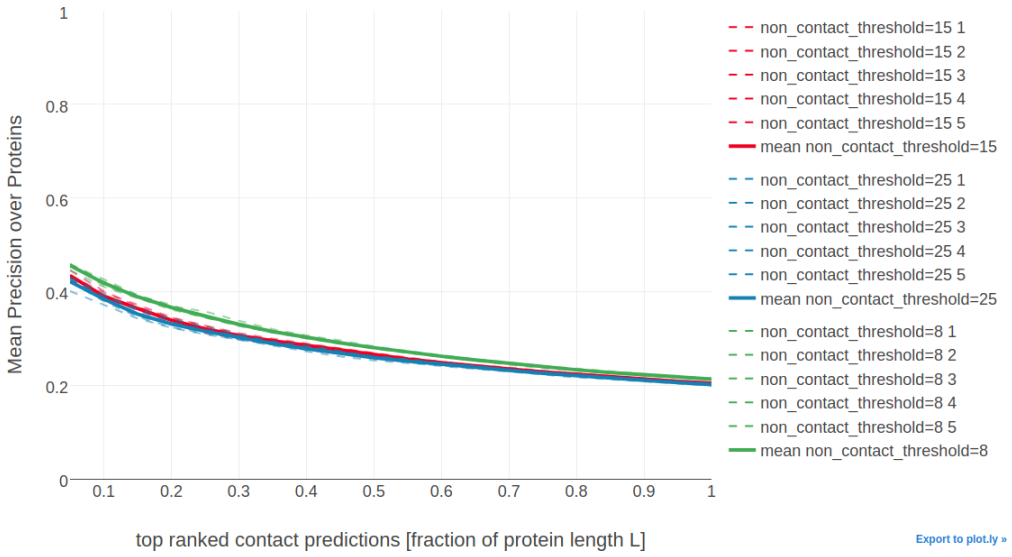


Figure E.2: Mean precision over validation set of 200 proteins for top ranked contact predictions for different choices of the non-contact threshold to define non-contacts. Dashed lines represent the models trained on four subsets of the training data according to the 5-fold cross-validation scheme. Solid lines represent the mean over the five cross-validation models.

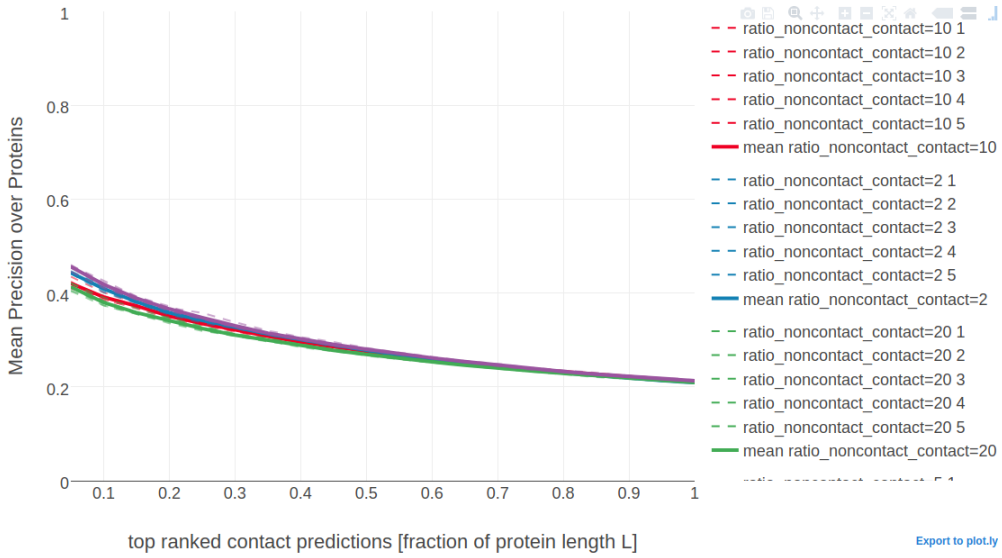


Figure E.3: Mean precision over validation set of 200 proteins for top ranked contact predictions for different choices of dataset composition with respect to the ratio of contacts and non-contacts. Dashed lines represent the models trained on four subsets of the training data according to the 5-fold cross-validation scheme. Solid lines represent the mean over the five cross-validation models.

List of Figures

List of Tables

References

1. Anfinsen, C.B. (1973). Principles that Govern the Folding of Protein Chains. Science (80-.). *181*, 223–230. Available at: http://www.sciencemag.org/content/181/4096/223.

2. Wright, P.E., and Dyson, H. (1999). Intrinsically unstructured proteins: re-assessing the protein structure-function paradigm. J. Mol. Biol. *293*, 321–331. Available at: http://www.ncbi.nlm.nih.gov/pubmed/10550212 http://linkinghub.elsevier.com/retrieve/pii/S0022283699931108.

3. Samish, I., Bourne, P.E., and Najmanovich, R.J. (2015). Achievements and challenges in structural bioinformatics and computational biophysics. Bioinformatics *31*, 146–150. Available at: https://oup.silverchair-cdn.com/oup/backfile/Content{_}public/Journal/bioinformatics/31/1/10.1093{_}bioinformatics{_}r8gIyHaHX5ANqUoy6w0PUuete2b{~}5ZU{~}{~}D6KOB1vq5A8MgCnrq3pDHUn0OSgz0QFmtU2RY
.

4. Lesk, A.M., and Chothia, C. (1980). How different amino acid sequences determine similar protein structures: The structure and evolutionary dynamics of the globins. J. Mol. Biol. *136*, 225–270. Available at: http://linkinghub.elsevier.com/retrieve/pii/0022283680903733.

5. Sander, C., and Schneider, R. (1991). Database of homology-derived protein structures and the structural meaning of sequence alignment. Proteins *9*, 56–68. Available at: http://www.ncbi.nlm.nih.gov/pubmed/2017436.

6. Chothia, C., and Lesk, A.M. (1986). The relation between the divergence of sequence and structure in proteins. EMBO J. *5*, 823–6. Available at: http://www.ncbi.nlm.nih.gov/pubmed/3709526 http://www.pubmedcentral.nih.gov/articlerender.fcgi?

7. Mart-Renom, M.A., Stuart, A.C., Fiser, A., Snchez, R., Melo, F., and ali, A. (2000). Comparative Protein Structure Modeling of Genes and Genomes. Annu. Rev. Biophys. Biomol. Struct. *29*, 291–325. Available at: http://www.annualreviews.org/doi/10.1146/annurev.biophys.29.1.291.

8. Dorn, M., Silva, M.B. e, Buriol, L.S., and Lamb, L.C. (2014). Three-dimensional protein structure prediction: Methods and computational strategies. Comput. Biol. Chem. *53*, 251–276. Available at: http://linkinghub.elsevier.com/retrieve/pii/S1476927114001248.

9. Berman, H.M. (2000). The Protein Data Bank. Nucleic Acids Res. *28*, 235–242. Available at: http://nar.oxfordjournals.org/content/28/1/235.abstract.

10. Egelman, E.H. (2016). The Current Revolution in Cryo-EM. Biophysj *110*, 1008–1012. Available at: http://www.cell.com/biophysj/pdf/

S0006-3495(16)00142-9.pdf.

11. The UniProt Consortium (2013). Update on activities at the Universal Protein Resource (UniProt) in 2013. Nucleic Acids Res. *41*, D43–7. Available at: http://nar.oxfordjournals.org/content/41/D1/D43.

12. Waters, M.L. (2002). Aromatic interactions in model systems. Curr. Opin. Chem. Biol. *6*, 736–741. Available at: http://dx.doi.org/10.1016/S1367-5931(02)00359-9.

13. Burley, S., and Petsko, G. (1985). Aromatic-aromatic interaction: a mechanism of protein structure stabilization. Science (80-.). *229*, 23–28. Available at: http://www.sciencemag.org/cgi/doi/10.1126/science.3892686.

14. Thornton, J.M. (1981). Disulphide bridges in globular proteins. J. Mol. Biol. *151*, 261–87. Available at: http://www.ncbi.nlm.nih.gov/pubmed/7338898.

15. Bastolla, U., and Demetrius, L. (2005). Stability constraints and protein evolution: the role of chain length, composition and disulfide bonds. Protein Eng. Des. Sel. *18*, 405–415. Available at: https://academic.oup.com/peds/article-lookup/doi/10.1093/protein/gzi045.

16. Donald, J.E., Kulp, D.W., and DeGrado, W.F. (2011). Salt bridges: geometrically specific, designable interactions. Proteins *79*, 898–915. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3069487{\&}tool=pmcentrez{\&}

17. Burley, S., and Petsko, G. (1986). Amino-aromatic interactions in proteins. FEBS Lett. *203*, 139–143. Available at: http://dx.doi.org/10.1016/0014-5793(86)80730-X.

18. Crowley, P.B., and Golovin, A. (2005). Cation-pi interactions in protein-protein interfaces. Proteins *59*, 231–9. Available at: http://www.ncbi.nlm.nih.gov/pubmed/15726638.

19. Slutsky, M.M., and Marsh, E.N.G. (2004). Cation-pi interactions studied in a model coiled-coil peptide. Protein Sci. *13*, 2244–51. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2279832{\&}tool=pmcentrez{\&}

20. Levinthal, C. (1969). How to Fold Graciously. 22–24. Available at: http://www.citeulike.org/user/FBerkemeier/article/380320.

21. Rost, B. (1999). Twilight zone of protein sequence alignments. Protein Eng. Des. Sel. *12*, 85–94. Available at: http://peds.oxfordjournals.org/content/12/2/85.full.

22. Yan, R., Xu, D., Yang, J., Walker, S., and Zhang, Y. (2013). A comparative assessment and analysis of 20 representative sequence alignment methods for protein structure prediction. Sci. Rep. *3*, 2619. Available at: http://www.nature.com/srep/2013/130910/srep02619/full/srep02619.html.

23. Meier, A. (2015). Automatic Prediction of Protein 3D Structures by Probabilistic Multi-template Homology Modeling. PLoS Comput Biol *11*, 1–20.

24. Gu, J., and Bourne, P.E. (2009). Structural Bioinformatics (Wiley-Blackwell) Available at: http://www.amazon.com/Structural-Bioinformatics-Jenny-Gu/dp/0470181052.

25. Eswar, N., Webb, B., Marti-Renom, M.A., Madhusudhan, M.S., Eramian,

D., Shen, M.-Y., Pieper, U., and Sali, A. (2007). Comparative protein structure modeling using MODELLER. Curr. Protoc. Protein Sci. *Chapter 2*, Unit 2.9. Available at: http://www.ncbi.nlm.nih.gov/pubmed/18429317.

26. Bates, P.A., Kelley, L.A., MacCallum, R.M., and Sternberg, M.J. (2001). Enhancement of protein modeling by human intervention in applying the automatic programs 3D-JIGSAW and 3D-PSSM. Proteins *Suppl 5*, 39–46. Available at: http://www.ncbi.nlm.nih.gov/pubmed/11835480.

27. Arnold, K., Bordoli, L., Kopp, J., and Schwede, T. (2006). The SWISS-MODEL workspace: a web-based environment for protein structure homology modelling. Bioinformatics *22*, 195–201. Available at: http://bioinformatics.oxfordjournals.org/content/22/2/195.short.

28. Gbel, U., Sander, C., Schneider, R., and Valencia, A. (1994). Correlated mutations and residue contacts in proteins. Proteins *18*, 309–17. Available at: http://www.ncbi.nlm.nih.gov/pubmed/8208723.

29. Godzik, A., and Sander, C. (1989). Conservation of residue interactions in a family of Ca-binding proteins. "Protein Eng. Des. Sel. *2*, 589–596. Available at: https://academic.oup.com/peds/article-lookup/doi/10.1093/protein/2.8.589.

30. Neher, E. (1994). How frequent are correlated changes in families of protein sequences? Proc. Natl. Acad. Sci. U. S. A. *91*, 98–102. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=42893{\&}tool=pmcentrez{\&}rendertyp

31. Taylor, W.R., and Hatrick, K. (1994). Compensating changes in protein multiple sequence alignments. "Protein Eng. Des. Sel. *7*, 341–348. Available at: http://peds.oxfordjournals.org/content/7/3/341.abstract.

32. Oliveira, L., Paiva, A.C.M., and Vriend, G. (2002). Correlated mutation analyses on very large sequence families. Chembiochem *3*, 1010–7. Available at: http://onlinelibrary.wiley.com/doi/10.1002/1439-7633(20021004)3:10{\%}3C1010::AID-CBIC1010{\%}3E3.0.CO;2-T/full.

33. Shindyalov, I., Kolchanov, N., and Sander, C. (1994). Can three-dimensional contacts in protein structures be predicted by analysis of correlated mutations? "Protein Eng. Des. Sel. *7*, 349–358. Available at: http://peds.oxfordjournals.org/content/7/3/349.

34. Clarke, N.D. (1995). Covariation of residues in the homeodomain sequence family. Protein Sci. *4*, 2269–78. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?art

35. Korber, B. (1993). Covariation of Mutations in the V3 Loop of Human Immunodeficiency Virus Type 1 Envelope Protein: An Information Theoretic Analysis. Proc. Natl. Acad. Sci. *90*, 7176–7180. Available at: http://www.pnas.org/content/90/15/7176.abstract?ijkey=6e1c9cbdef66bd0beefedc88ea07591b837f2213

36. Martin, L.C., Gloor, G.B., Dunn, S.D., and Wahl, L.M. (2005). Using information theory to search for co-evolving residues in proteins. Bioinformatics *21*, 4116–24. Available at: http://bioinformatics.oxfordjournals.org/content/21/22/4116.full.

37. Atchley, W.R., Wollenberg, K.R., Fitch, W.M., Terhalle, W., and Dress, A.W. (2000). Correlations Among Amino Acid Sites in bHLH Protein Domains:

An Information Theoretic Analysis. Mol. Biol. Evol. *17*, 164–178. Available at: http://mbe.oxfordjournals.org/content/17/1/164.abstract?ijkey=2a1f0a044a8fd2213955e4f2c17f

38. Fodor, A.A., and Aldrich, R.W. (2004). Influence of conservation on calculations of amino acid covariance in multiple sequence alignments. Proteins *56*, 211–21. Available at: http://www.ncbi.nlm.nih.gov/pubmed/15211506.

39. Tillier, E.R., and Lui, T.W. (2003). Using multiple interdependency to separate functional from phylogenetic correlations in protein alignments. Bioinformatics *19*, 750–755. Available at: http://bioinformatics.oxfordjournals.org/content/19/6/750.abstra

40. Gouveia-Oliveira, R., and Pedersen, A.G. (2007). Finding coevolving amino acid residues using row and column weighting of mutual information and multidimensional amino acid representation. Algorithms Mol. Biol. *2*, 12. Available at: http://www.almob.org/content/2/1/12.

41. Dunn, S.D., Wahl, L.M., and Gloor, G.B. (2008). Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction. Bioinformatics *24*, 333–40. Available at: http://bioinformatics.oxfordjournals.org/content/24/3/333.

42. Kass, I., and Horovitz, A. (2002). Mapping pathways of allosteric communication in GroEL by analysis of correlated mutations. Proteins *48*, 611–7. Available at: http://www.ncbi.nlm.nih.gov/pubmed/12211028.

43. Noivirt, O., Eisenstein, M., and Horovitz, A. (2005). Detection and reduction of evolutionary noise in correlated mutation analysis. Protein Eng. Des. Sel. *18*, 247–53. Available at: http://peds.oxfordjournals.org/content/18/5/247.full.

44. Juan, D. de, Pazos, F., and Valencia, A. (2013). Emerging methods in protein co-evolution. Nat. Rev. Genet. *14*, 249–61. Available at: http://www.readcube.com/articles/10.1038/nrg3414?locale=en.

45. Jones, D.T., Buchan, D.W.A., Cozzetto, D., and Pontil, M. (2012). PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. Bioinformatics *28*, 184–90. Available at: http://bioinformatics.oxfordjournals.org/content/28/2/184.full.

46. Lapedes, A., Giraud, B., Liu, L., and Stormo, G. (1999). Correlated mutations in models of protein sequences: phylogenetic and structural effects. *33*, 236–256. Available at: http://www.citeulike.org/user/qluo/article/5092214.

47. Weigt, M., White, R.A., Szurmant, H., Hoch, J.A., and Hwa, T. (2009). Identification of direct residue contacts in protein-protein interaction by message passing. Proc. Natl. Acad. Sci. U. S. A. *106*, 67–72. Available at: http://www.pnas.org/content/106/1/67.abstract.

48. Burger, L., and Nimwegen, E. van (2008). Accurate prediction of protein-protein interactions from sequence alignments using a Bayesian method. Mol. Syst. Biol. *4*, 165. Available at: http://msb.embopress.org/content/4/1/165.abstract.

49. Burger, L., and Nimwegen, E. van (2010). Disentangling direct from indirect co-evolution of residues in protein alignments. PLoS Comput. Biol. *6*, e1000633.

Available at: http://dx.plos.org/10.1371/journal.pcbi.1000633.

50. Cheng, J., and Baldi, P. (2007). Improved residue contact prediction using support vector machines and a large feature set. BMC Bioinformatics *8*. Available at: http://dx.doi.org/10.1186/1471-2105-8-113.

51. Wu, S., and Zhang, Y. (2008). A comprehensive assessment of sequence-based and template-based methods for protein contact prediction. Bioinformatics *24*, 924–31. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2648832{\&}tool=p

52. Li, Y., Fang, Y., and Fang, J. (2011). Predicting residue-residue contacts using random forest models. Bioinformatics *27*, 3379–84. Available at: http://bioinformatics.oxfordjournals.org/content/27/24/3379.long.

53. Wang, X.-F., Chen, Z., Wang, C., Yan, R.-X., Zhang, Z., and Song, J. (2011). Predicting residue-residue contacts and helix-helix interactions in transmembrane proteins using an integrative feature-based random forest approach. PLoS One *6*, e26767. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3203928{\&}tool=p

54. Wang, Z., and Xu, J. (2013). Predicting protein contact map using evolutionary and physical constraints by integer programming. Bioinformatics *29*, i266–73. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3694661{\&}tool=pmcentre

55. Fariselli, P., Olmea, O., Valencia, A., and Casadio, R. (2001). Prediction of contact maps with neural networks and correlated mutations. Protein Eng. Des. Sel. *14*, 835–843. Available at: http://peds.oxfordjournals.org/content/14/11/835.long.

56. Shackelford, G., and Karplus, K. (2007). Contact prediction using mutual information and neural nets. Proteins *69 Suppl 8*, 159–64. Available at: http://www.ncbi.nlm.nih.gov/pubmed/17932918.

57. Hamilton, N., Burrage, K., Ragan, M.A., and Huber, T. (2004). Protein contact prediction using patterns of correlation. Proteins Struct. Funct. Bioinforma. *56*, 679–684. Available at: http://onlinelibrary.wiley.com/doi/10.1002/prot.20160/abstract.

58. Xue, B., Faraggi, E., and Zhou, Y. (2009). Predicting residue-residue contact maps by a two-layer, integrated neural-network method. Proteins *76*, 176–83. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2716487{\&}tool=pmcentre

59. Tegge, A.N., Wang, Z., Eickholt, J., and Cheng, J. (2009). NNcon: improved protein contact map prediction using 2D-recursive neural networks. Nucleic Acids Res. *37*, W515–8. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2703959{\

60. Eickholt, J., and Cheng, J. (2012). Predicting protein residue–residue contacts using deep networks and boosting. *28*, 3066–3072.

61. Di Lena, P., Nagata, K., and Baldi, P. (2012). Deep architectures for protein contact map prediction. Bioinformatics *28*, 2449–57. Available at: http://bioinformatics.oxfordjournals.org/content/28/19/2449.full{\#}sec-14.

62. Chen, P., and Li, J. (2010). Prediction of protein long-range contacts using an ensemble of genetic algorithm classifiers with sequence profile centers. BMC Struct. Biol. *10 Suppl 1*, S2. Available at: http://bmcstructbiol.

biomedcentral.com/articles/10.1186/1472-6807-10-S1-S2.

63. Kosciolek, T., and Jones, D.T. (2015). Accurate contact predictions using coevolution techniques and machine learning. Proteins Struct. Funct. Bioinforma., n/a–n/a. Available at: http://www.ncbi.nlm.nih.gov/pubmed/26205532.

64. Skwark, M.J., Abdel-Rehim, A., and Elofsson, A. (2013). PconsC: combination of direct information methods and alignments improves contact prediction. Bioinformatics *29*, 1815–6. Available at: http://bioinformatics.oxfordjournals.org/content/29/14/1815.long.

65. Skwark, M.J., Michel, M., Menendez Hurtado, D., Ekeberg, M., and Elofsson, A. (2016). Accurate contact predictions for thousands of protein families using PconsC3. bioRxiv.

66. Schneider, M., and Brock, O. (2014). Combining Physicochemical and Evolutionary Information for Protein Contact Prediction. PLoS One *9*, e108438. Available at: http://www.plosone.org/article/info{\%}3Adoi{\%}2F10.1371{\%}2Fjournal.pone.010

67. Jones, D.T., Singh, T., Kosciolek, T., and Tetchner, S. (2015). MetaPSICOV: combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins. Bioinformatics *31*, 999–1006. Available at: http://bioinformatics.oxfordjournals.org/content/31/7/999.short.

68. Wang, S., Sun, S., Li, Z., Zhang, R., and Xu, J. (2016). Accurate De Novo Prediction of Protein Contact Map by Ultra-Deep Learning Model. PLoS Comput. Biol. *13*, e1005324. Available at: http://arxiv.org/abs/1609.00680 http://www.ncbi.nlm.nih.gov/pubmed/28056090 http://www.pubmedcentral.nih.gov/articleren

69. Stahl, K., Schneider, M., and Brock, O. (2017). EPSILON-CP: using deep learning to combine information from multiple sources for protein contact prediction. BMC Bioinformatics *18*, 303. Available at: http://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-017-1713-x.

70. He, B., Mortuza, S.M., Wang, Y., Shen, H.-B., and Zhang, Y. (2017). NeBcon: Protein contact map prediction using neural network training coupled with nave Bayes classifiers. Bioinformatics. Available at: https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btx164.

71. Monastyrskyy, B., D'Andrea, D., Fidelis, K., Tramontano, A., and Kryshtafovych, A. (2015). New encouraging developments in contact prediction: Assessment of the CASP11 results. Proteins. Available at: http://www.ncbi.nlm.nih.gov/pubmed/26474083.

72. Jaynes, E.T. (1957). Information Theory and Statistical Mechanics I. Phys. Rev. *106*, 620–630. Available at: https://link.aps.org/doi/10.1103/PhysRev.106.620.

73. Jaynes, E.T. (1957). Information Theory and Statistical Mechanics. II. Phys. Rev. *108*, 171–190. Available at: https://link.aps.org/doi/10.1103/PhysRev.108.171.

74. Wainwright, M.J., and Jordan, M.I. (2007). Graphical Models, Exponential Families, and Variational Inference. Found. Trends Mach. Learn. *1*, 1–305.

Available at: http://www.nowpublishers.com/article/Details/MAL-001.

75. Murphy, K.P. (2012). Machine Learning: A Probabilistic Perspective (MIT Press).

76. Morcos, F., Pagnani, A., Lunt, B., Bertolino, A., Marks, D.S., Sander, C., Zecchina, R., Onuchic, J.N., Hwa, T., and Weigt, M. (2011). Direct-coupling analysis of residue coevolution captures native contacts across many protein families. Proc. Natl. Acad. Sci. U. S. A. *108*, E1293–301. Available at: http://www.pnas.org/content/108/49/E1293.full.

77. Marks, D.S., Colwell, L.J., Sheridan, R., Hopf, T.A., Pagnani, A., Zecchina, R., and Sander, C. (2011). Protein 3D structure computed from evolutionary sequence variation. PLoS One *6*, e28766. Available at: http://dx.plos.org/10.1371/journal.pone.0028766.

78. Cocco, S., Feinauer, C., Figliuzzi, M., Monasson, R., and Weigt, M. (2017). Inverse Statistical Physics of Protein Sequences: A Key Issues Review. arXiv. Available at: https://arxiv.org/pdf/1703.01222.pdf.

79. Koller, D., and Friedman, N.I.R. (2009). Probabilistic graphical models: Principles and Techniques (MIT Press).

80. Ekeberg, M., Lvkvist, C., Lan, Y., Weigt, M., and Aurell, E. (2013). Improved contact prediction in proteins: Using pseudolikelihoods to infer Potts models. Phys. Rev. E *87*, 012707. Available at: http://link.aps.org/doi/10.1103/PhysRevE.87.012707.

81. Stein, R.R., Marks, D.S., and Sander, C. (2015). Inferring Pairwise Interactions from Biological Data Using Maximum-Entropy Probability Models. PLOS Comput. Biol. *11*, e1004182. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4520494{\&}tool=pmcentrez{\&}rendert

82. Seemayer, S., Gruber, M., and Sding, J. (2014). CCMpred-fast and precise prediction of protein residue-residue contacts from correlated mutations. Bioinformatics, btu500. Available at: http://bioinformatics.oxfordjournals.org/content/early/2014/08/12/bioinformatics.btu500.

83. Ekeberg, M., Hartonen, T., and Aurell, E. (2014). Fast pseudolikelihood maximization for direct-coupling analysis of protein structure from many homologous amino-acid sequences. J. Comput. Phys. *276*, 341–356. Available at: http://www.sciencedirect.com/science/article/pii/S0021999114005178.

84. Kamisetty, H., Ovchinnikov, S., and Baker, D. (2013). Assessing the utility of coevolution-based residue-residue contact predictions in a sequence- and structure-rich era. Proc. Natl. Acad. Sci. U. S. A. *110*, 15674–9. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3785744{\&}tool=pmcentrez{\&}rendert

85. Balakrishnan, S., Kamisetty, H., Carbonell, J.G., Lee, S.-I., and Langmead, C.J. (2011). Learning generative models for protein fold families. Proteins *79*, 1061–78. Available at: http://www.ncbi.nlm.nih.gov/pubmed/21268112.

86. Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. Biostatistics *9*, 432–41. Available at: http://biostatistics.oxfordjournals.org/content/9/3/432.abstract.

87. Banerjee, O., El Ghaoui, L., and D'Aspremont, A. (2008). Model Selection

Through Sparse Maximum Likelihood Estimation for Multivariate Gaussian or Binary Data. J. Mach. Learn. Res. *9*, 485–516. Available at: http://dl.acm.org/citation.cfm?id=1390681.1390696.

88. Baldassi, C., Zamparo, M., Feinauer, C., Procaccini, A., Zecchina, R., Weigt, M., and Pagnani, A. (2014). Fast and accurate multivariate gaussian modeling of protein families: predicting residue contacts and protein-interaction partners. PLoS One *9*, e92721. Available at: http://dx.plos.org/10.1371/journal.pone.0092721.

89. Besag, J. (1975). Statistical Analysis of Non-Lattice Data. Source Stat. *24*, 179–195. Available at: http://www.jstor.org http://www.jstor.org/stable/2987782.

90. Gidas, B. (1988). Consistency of maximum likelihood and pseudo-likelihood estimators for Gibbs Distributions. Stoch. Differ. Syst. Stoch. Control Theory Appl. Available at: http://www.researchgate.net/publication/244456377{_}Consistency{_}of{
likelihood{_}estimators{_}for{_}Gibbs{_}Distributions.

91. Feinauer, C., Skwark, M.J., Pagnani, A., and Aurell, E. (2014). Improving contact prediction along three dimensions. 19. Available at: http://arxiv.org/abs/1403.0379.

92. Zhang, H., Huang, Q., Bei, Z., Wei, Y., and Floudas, C.A. (2016). COMSAT: Residue contact prediction of transmembrane proteins based on support vector machines and mixed integer linear programming. Proteins Struct. Funct. Bioinforma., n/a–n/a. Available at: http://www.ncbi.nlm.nih.gov/pubmed/26756402.

93. Monastyrskyy, B., Fidelis, K., Tramontano, A., and Kryshtafovych, A. (2011). Evaluation of residue-residue contact predictions in CASP9. Proteins *79 Suppl 1*, 119–25. Available at: http://www.ncbi.nlm.nih.gov/pubmed/21928322 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3226919.

94. Monastyrskyy, B., D'Andrea, D., Fidelis, K., Tramontano, A., and Kryshtafovych, A. (2014). Evaluation of residue-residue contact prediction in CASP10. Proteins *82 Suppl 2*, 138–53. Available at: http://www.pubmedcentral.nih.gov/articler

95. Ashkenazy, H., Unger, R., and Kliger, Y. (2009). Optimal data collection for correlated mutation analysis. Proteins *74*, 545–55. Available at: http://www.ncbi.nlm.nih.gov/pubmed/18655065.

96. Marks, D.S., Hopf, T.A., and Sander, C. (2012). Protein structure prediction from sequence variation. Nat. Biotechnol. *30*, 1072–1080. Available at: http://www.nature.com/nbt/journal/v30/n11/full/nbt.2419.html.

97. Buslje, C.M., Santos, J., Delfino, J.M., and Nielsen, M. (2009). Correction for phylogeny, small number of observations and data redundancy improves the identification of coevolving amino acid pairs using mutual information. Bioinformatics *25*, 1125–31. Available at: http://www.ncbi.nlm.nih.gov/pubmed/19276150 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2672635.

98. Anishchenko, I., Ovchinnikov, S., Kamisetty, H., and Baker, D. (2017). Origins of coevolution between residues distant in protein 3D structures. Proc. Natl. Acad. Sci., 201702664. Available at: http://www.ncbi.nlm.nih.gov/pubmed/28784799

http://www.pnas.org/lookup/doi/10.1073/pnas.1702664114.

99. Finn, R.D., Coggill, P., Eberhardt, R.Y., Eddy, S.R., Mistry, J., Mitchell, A.L., Potter, S.C., Punta, M., Qureshi, M., and Sangrador-Vegas, A. *et al.* (2016). The Pfam protein families database: towards a more sustainable future. Nucleic Acids Res. *44*, D279–D285. Available at: https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkv1344.

100. Remmert, M., Biegert, A., Hauser, A., and Sding, J. (2012). HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. Nat. Methods *9*, 173–5. Available at: http://dx.doi.org/10.1038/nmeth.1818.

101. Espada, R., Parra, R.G., Mora, T., Walczak, A.M., and Ferreiro, D. (2015). Capturing coevolutionary signals in repeat proteins. BMC Bioinformatics *16*, 207. Available at: http://arxiv.org/abs/1407.6903.

102. Toth-Petroczy, A., Palmedo, P., Ingraham, J., Hopf, T.A., Berger, B., Sander, C., Marks, D.S., Alexander, P., He, Y., and Chen, Y. *et al.* (2016). Structured States of Disordered Proteins from Genomic Sequences. Cell *167*, 158–170.e12. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0092867416312430.

103. Avila-Herrera, A., and Pollard, K.S. (2015). Coevolutionary analyses require phylogenetically deep alignments and better null models to accurately detect inter-protein contacts within and between species. BMC Bioinformatics *16*, 268. Available at: http://www.biomedcentral.com/1471-2105/16/268.

104. Hopf, T.A., Colwell, L.J., Sheridan, R., Rost, B., Sander, C., and Marks, D.S. (2012). Three-dimensional structures of membrane proteins from genomic sequencing. Cell *149*, 1607–21. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3641

105. Betts, M.J., and Russell, R.B. Amino Acid Properties and Consequences of Substitutions. In Bioinforma. genet. (Chichester, UK: John Wiley & Sons, Ltd), pp. 289–316. Available at: http://doi.wiley.com/10.1002/0470867302.ch14.

106. Duarte, J.M., Sathyapriya, R., Stehr, H., Filippis, I., and Lappe, M. (2010). Optimal contact definition for reconstruction of contact maps. BMC Bioinformatics *11*, 283. Available at: http://www.biomedcentral.com/1471-2105/11/283.

107. Lee, B.-C., and Kim, D. (2009). A new method for revealing correlated mutations under the structural and functional constraints in proteins. Bioinformatics *25*, 2506–13. Available at: http://www.ncbi.nlm.nih.gov/pubmed/19628501.

108. Wang, Y., and Barth, P. (2015). Evolutionary-guided de novo structure prediction of self-associated transmembrane helical proteins with near-atomic accuracy. Nat. Commun. *6*, 7196. Available at: http://www.nature.com/ncomms/2015/150521/ncomms8196/abs/ncomms8196.html.

109. Sutto, L., Marsili, S., Valencia, A., and Gervasio, F.L. (2015). From residue coevolution to protein conformational ensembles and functional dynamics. Proc. Natl. Acad. Sci. U. S. A., 1508584112. Available at: http://www.pnas.org/content/early/2015/10/20/1508584112.abstract.

110. Jana, B., Morcos, F., and Onuchic, J.N. (2014). From structure to function: the convergence of structure based models and co-evolutionary

information. Phys. Chem. Chem. Phys. *16*, 6496. Available at: http://www.ncbi.nlm.nih.gov/pubmed/24603809 http://xlink.rsc.org/?DOI=c3cp55275f.

111. Dos Santos, R.N., Morcos, F., Jana, B., Andricopulo, A.D., and Onuchic, J.N. (2015). Dimeric interactions and complex formation using direct coevolutionary couplings. Sci. Rep. *5*, 13652. Available at: http://www.nature.com/srep/2015/150904/srep13652/full/srep13652.html.

112. Ovchinnikov, S., Kim, D.E., Wang, R.Y.-R., Liu, Y., DiMaio, F., and Baker, D. (2015). Improved de novo structure prediction in CASP11 by incorporating Co-evolution information into rosetta. Proteins. Available at: http://www.ncbi.nlm.nih.gov/pubmed/26677056.

113. Noel, J.K., Morcos, F., and Onuchic, J.N. (2016). Sequence co-evolutionary information is a natural partner to minimally-frustrated models of biomolecular dynamics. F1000Research *5*. Available at: http://f1000research.com/articles/5-106/v1.

114. Sfriso, P., Duran-Frigola, M., Mosca, R., Emperador, A., Aloy, P., and Orozco, M. (2016). Residues Coevolution Guides the Systematic Identification of Alternative Functional Conformations in Proteins. Structure *24*, 116–126. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0969212615004657.

115. Coucke, A., Uguzzoni, G., Oteri, F., Cocco, S., Monasson, R., and Weigt, M. (2016). Direct coevolutionary couplings reflect biophysical residue interactions in proteins. J. Chem. Phys. *145*, 174102. Available at: http://scitation.aip.org/content/aip/journal/jcp/145/17/10.1063/1.4966156.

116. Hinton, G.E. (2002). Training Products of Experts by Minimizing Contrastive Divergence. Neural Comput. *14*, 1771–1800. Available at: http://www.gatsby.ucl.ac.uk/publications/tr/tr00-004.pdf.

117. Hyvrinen, A. (2006). Consistency of pseudolikelihood estimation of fully visible Boltzmann machines. Available at: https://www.cs.helsinki.fi/u/ahyvarin/papers/NC06.pdf.

118. Hyvarinen, A. (2007). Connections Between Score Matching, Contrastive Divergence, and Pseudolikelihood for Continuous-Valued Variables. IEEE Trans. Neural Networks *18*, 1529–1531. Available at: http://ieeexplore.ieee.org/document/4298117/.

119. Ma, J., Wang, S., Wang, Z., and Xu, J. (2015). Protein contact prediction by integrating joint evolutionary coupling analysis and supervised learning. Bioinformatics, btv472. Available at: http://bioinformatics.oxfordjournals.org/content/early/2015/09/04/bioinformatics.btv472.

120. Ho, T.K. (1998). The random subspace method for constructing decision forests. IEEE Trans. Pattern Anal. Mach. Intell. *20*, 832–844. Available at: http://ieeexplore.ieee.org/document/709601/.

121. Tin Kam Ho (1995). Random decision forests. In Proc. 3rd int. conf. doc. anal. recognit. (IEEE Comput. Soc. Press), pp. 278–282. Available at: http://ieeexplore.ieee.org/document/598994/.

122. Breiman, L. (2001). Random Forests. Mach. Learn. *45*, 5–32. Available at:

http://link.springer.com/10.1023/A:1010933404324.

123. Menze, B.H., Kelm, B.M., Masuch, R., Himmelreich, U., Bachert, P., Petrich, W., and Hamprecht, F.A. (2009). A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data. BMC Bioinformatics *10*, 213. Available at: http://www.ncbi.nlm.nih.gov/pubmed/19591666 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2724423.

124. Strobl, C., Boulesteix, A.-L., Zeileis, A., and Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. BMC Bioinformatics *8*, 25. Available at: http://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-8-25.

125. Miyazawa, S., and Jernigan, R.L. (1999). Self-consistent estimation of inter-residue protein contact energies based on an equilibrium mixture approximation of residues. Proteins *34*, 49–68. Available at: http://www.ncbi.nlm.nih.gov/pubmed/10336383.

126. Petersen, B., Petersen, T.N., Andersen, P., Nielsen, M., and Lundegaard, C. (2009). BMC Structural Biology A generic method for assignment of reliability scores applied to solvent accessibility predictions. BMC Struct. Biol. *9*. Available at: http://www.biomedcentral.com/1472-6807/9/51.

127. Sillitoe, I., Lewis, T.E., Cuff, A., Das, S., Ashford, P., Dawson, N.L., Furnham, N., Laskowski, R.A., Lee, D., and Lees, J.G. *et al.* (2015). CATH: comprehensive structural and functional annotations for genome sequences. Nucleic Acids Res. *43*, D376–D381. Available at: https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gku947.

128. Ruder, S. (2017). An overview of gradient descent optimization algorithms. arXiv. Available at: http://arxiv.org/abs/1609.04747.

129. Bottou, L. (2012). Stochastic Gradient Descent Tricks. In Neural networks: Tricks of the trade (Springer, Berlin, Heidelberg), pp. 421–436. Available at: http://link.springer.com/10.1007/978-3-642-35289-8{_}25.

130. Schaul, T., Zhang, S., and Lecun, Y. (2013). No More Pesky Learning Rates. arXiv. Available at: https://arxiv.org/pdf/1206.1106.pdf.

131. Carreira-Perpin, M. a, and Hinton, G.E. (2005). On Contrastive Divergence Learning. Artif. Intell. Stat. *0*, 17. Available at: http://learning.cs.toronto.edu/{~}hinton/absps/cdmi

132. Bengio, Y., and Delalleau, O. (2009). Justifying and Generalizing Contrastive Divergence. Neural Comput. *21*, 1601–21. Available at: http://www.iro.umontreal.ca/{~}lisa/publications2/index.php/attachments/single/105.

133. Tieleman, T. (2008). Training Restricted Boltzmann Machines using Approximations to the Likelihood Gradient. Proc. 25th Int. Conf. Mach. Learn. *307*, 7.

134. Jones, D.T. (1999). Protein secondary structure prediction based on position-specific scoring matrices 1 1Edited by G. Von Heijne. J. Mol. Biol. *292*, 195–202. Available at: http://www.ncbi.nlm.nih.gov/pubmed/10493868

http://linkinghub.elsevier.com/retrieve/pii/S0022283699930917.

135. Robinson, A.B., and Robinson, L.R. (1991). Distribution of glutamine and asparagine residues and their near neighbors in peptides and proteins. Proc. Natl. Acad. Sci. U. S. A. *88*, 8880–4. Available at: http://www.ncbi.nlm.nih.gov/pubmed/1924347 http://www.pubmedcentral.nih.gov/articlerend

136. Atchley, W.R., Zhao, J., Fernandes, A.D., and Drke, T. (2005). Solving the protein sequence metric problem. Proc. Natl. Acad. Sci. U. S. A. *102*, 6395–400. Available at: http://www.pnas.org/content/102/18/6395.abstract.

137. Kawashima, S., Pokarowski, P., Pokarowska, M., Kolinski, A., Katayama, T., and Kanehisa, M. (2008). AAindex: amino acid index database, progress report 2008. Nucleic Acids Res. *36*, D202–5. Available at: http://www.ncbi.nlm.nih.gov/pubmed/17998252 http://www.pubmedcentral.nih.gov/articleren

138. Wimley, W.C., and White, S.H. (1996). Experimentally determined hydrophobicity scale for proteins at membrane interfaces. Nat. Struct. Biol. *3*, 842–8. Available at: http://www.ncbi.nlm.nih.gov/pubmed/8836100.

139. Cornette, J.L., Cease, K.B., Margalit, H., Spouge, J.L., Berzofsky, J.A., and DeLisi, C. (1987). Hydrophobicity scales and computational techniques for detecting amphipathic structures in proteins. J. Mol. Biol. *195*, 659–685. Available at: http://linkinghub.elsevier.com/retrieve/pii/0022283687901896.

140. Zhu, H., and Braun, W. (1999). Sequence specificity, statistical potentials, and three-dimensional structure prediction with self-correcting distance geometry calculations of beta-sheet formation in proteins. Protein Sci. *8*, 326–42. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2144259{\&}tool=pmcentrez{

141. Fodor, A.A., and Aldrich, R.W. (2004). Influence of conservation on calculations of amino acid covariance in multiple sequence alignments. Proteins *56*, 211–21. Available at: http://www.ncbi.nlm.nih.gov/pubmed/15211506.

142. Bernard, S., Heutte, L., and Adam, S. (2009). Influence of Hyperparameters on Random Forest Accuracy. In (Springer, Berlin, Heidelberg), pp. 171–180. Available at: http://link.springer.com/10.1007/978-3-642-02326-2{_}18.

143. Louppe, G. (2014). Understanding Random Forests: From Theory to Practice. Available at: http://arxiv.org/abs/1407.7502.

144. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., and Dubourg, V. *et al.* (2011). Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. *12*, 2825–2830. Available at: http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html.