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

Biological Background

In 1972, Anfinsen and his collegues received the Nobel Prize for their research on protein folding which lead to the postulation of 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 [1]. With certain exceptions (e.g. intrinsically disordered proteins [2], prions[3]), this dogma has proven to hold true at least for globular proteins.

Ever since, it is regarded as the biggest challenge in structural bioinformatics to realiably predict a protein's structure given only its amino acid sequence [4,5]. *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, due to the high degree of conformational flexibility, the search space of possible conformations cannot be explored exhaustively for proteins of typical length. Given a protein with 101 residues that has 100 peptide bonds with two torsion angles each and assuming three stable conformations for each of the bond angles, there will be $3^2 00 \approx 10^{95}$ configurations. This number of conformations cannot be sampled sequentially in a lifetime, even when sampling at high rates. Yet, proteins fold almost instantaneously within milliseconds. This discrepancy is known as Levinthal's paradox [6] and limits purely *de novo* based protein structure prediction to small proteins.

Far more successfull are template-based modelling approaches. Given the observation that structure is more conserved than sequence in a protein family [7], the structure of a target protein can be inferred from a homologue protein [8]. The degree of structural conservation is linked to the level of pairwise sequence identity [9]. Therefore, the accuracy of a model crucially depends on the sequence identity between target and template and determines the applicability of the model [10]. By definition, homology derived models are unable to capture new folds and their main limitation lies in the availability and identification of suitable templates [11].

The number of solved protein structures increases steadily but only slowly, as experimental methods are both time consuming and expensive [11]. The PDB[12] is the main repository for marcomolecular structures and currently (October 2017) holds about 135,000 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 left plot in Figure 1.1).

All three experimental techniques have advantages and limitations with respect to certain modelling aspects. X-ray crystallography involves protein overexpression, purification and crystallization and finding the the correct experimental conditions to arrive at a pure and

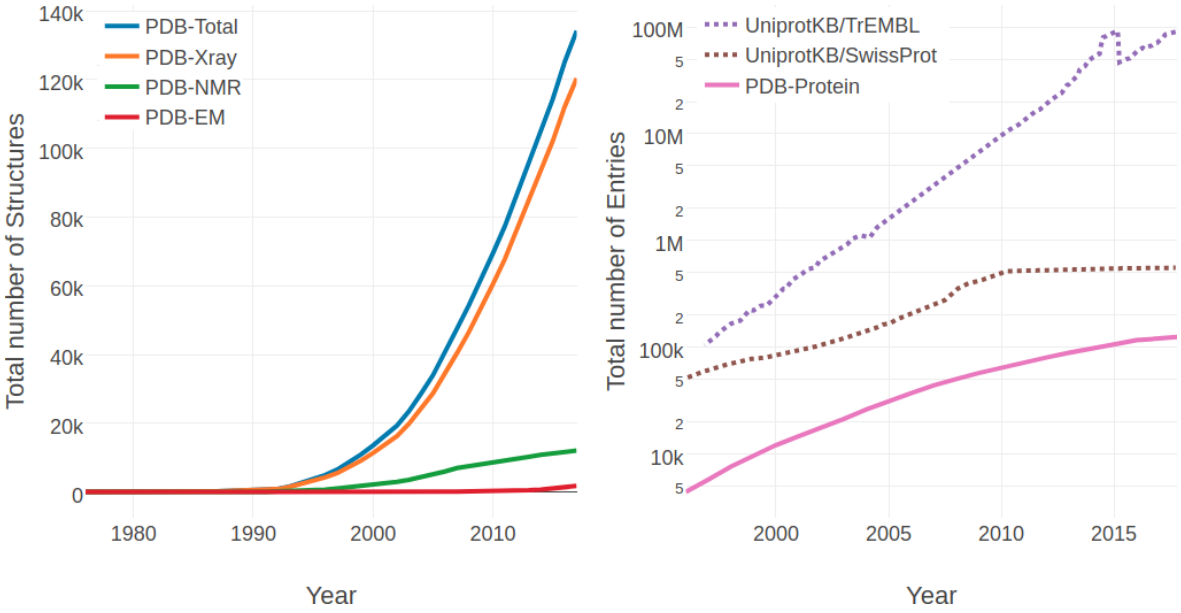

Figure 1.1: Yearly growth of number of solved structures in the PDB [12] and protein sequences in the Uniprot [13]. **Left** Yearly growth of structures in PDB by structure determination method. **Right** Yearly growth of protein sequences in the UniprotKB/TrEMBL, containing automatically annotated sequences, and in the UniprotKB/SwissProt, which is curated by experts who critically review experimental and predicted, and protein structures in the PDB.

regular crystal is a challenging and sometimes impossible task. Especially membrane proteins are difficult to study owing to their overall flexibility and hydrophobic surfaces which requires suitable detergents to extract the proteins from their membrane environment which in turn makes crystallization even more challenging [14,15]. Furthermore, the unnatural crystal environment can result in crystal-induced artifacts, like altererd sidechain conformations due to crystal packing interactions [16]. In contrast, Nuclear magnetic resonance (NMR) spectroscopy studies the protein in solution under physiological conditions and enables the observation of intramolecular dynamics, reaction kinetics or protein folding as ensembles of protein structures can be observed [17]. On the downside, validation of NMR-derived structure ensembles is complicated and there is an upper size limit of about 25 kDa for efficient use of the technique [18]. Recently, cryo-EM has undergone a "resolution revolution" and macromolecules have been solved to near-atomic resolutions [19,20]. Technological developments, such as better electron detectors as well as advanced image processing software has enabled high resolution structure determination and led to an exponential growth in number of structures deposited in the PDB. Cryo-EM is particularly suited to study large macromolecular complexes without the need to make crystals and therefore complements the other two structure determination techniques.

In contrast to the tedious task of determining the tertiary structure of a protein to atomic resolution, it has become very easy to decipher the primary sequence of proteins. Since the completion of the human genome in 2003, high-throughput sequencing technologies have been developed at an extraordinary pace, thereby not only decreasing the amount of time needed to sequence whole genomes but also drastically reducing costs [21]. The price for sequencing a single genome has dropped from the US$3 billion spent by the Human Genome Project to as little as US$1,000[22]. At the beginning of 2017, Illumina announced the launch of their latest high-throughput sequencing technology, NovaSeq, which is capable of sequencing $\sim$48 human genomes in parallel at 30x coverage within ˜45hours [23]. Advances in sequencing

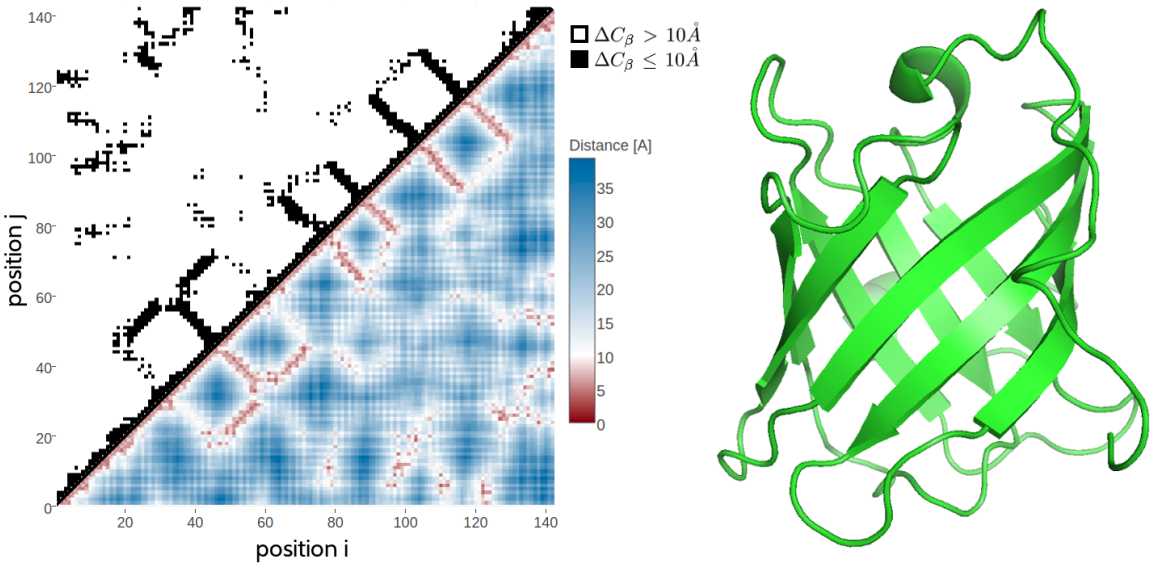

Figure 1.2: 2D and 3D representations of triabin, a thombin inhibitor from triatoma pallidipennis (PDB identifier 1avg chain I). **Left** The upper left matrix illustrates a contact map using an $10\mathring{A}$ C_β cutoff. A black square is drawn at position (i, j) if the C_β atoms of residues i and j are closer than $8\mathring{A}$ in the structure. The lower right matrix illustrates a distance map. Color reflects C_β distances between residue pairs with red colors representing $\Delta C_\beta \leq 10\mathring{A}$ and blue colors representing $\Delta C_\beta > 10\mathring{A}$. **Right** 3D Structure showing an eight-stranded beta-barrel.

technologies have led to the emergence of new fields of studies, like metagenomics and single-cell genomics, that enable sequencing of microorganisms that cannot be cultured in a lab [24–26]. With these approaches the genomic coverage of the microbial world is expanding which is directly reflected in a substantial increase in novel protein families [27–29]. About 90 million sequences (October 2017) have been translated into protein amino acid sequences and are stored in the UniprotKB/TrEMBL database, the leading resource for protein sequences [13].

The resultant gap between the number of protein structures and protein sequences is constantly widening (see right plot in Figure 1.1) despite tremendous efforts in automating experimental structure determination and new developments such as electron crystallography[5,30]. This trend illustrates the need of computational approaches that can complement experimental structural biology efforts in order to close this gap. Over the last decades, template-based methods have matured to a point where they are able to generate high-resolution structural models that are routinley and conveniently used in life-science research and by the biological community [5,31]. *De novo* methods aiming at predicting protein structures from sequence alone are required in case no homologue template structure can be identified or the protein sequence represents a novel fold. Albeit purely *de novo* approaches are hampered by the combinatorial explosion of possible conformations for larger proteins, combining them with structural information from heterogenous sources can help to reduce the degrees of freedom in the conformational search space [5]. For example, sparse low-resolution experimental data from chemical cross-linking/mass spectroscopy or nuclear Overhauser enhancement (NOE) distance data generated from NMR experiments, provide distance restraints to guide folding to a correct structure [32–34]. Sophisticated integrative approaches, exploiting structural information from different types of experiments have proven to be a powerful approach [35–37].

Recently, computational methods have been developed using co-evolution information from

deep multiple sequence alignments to predict contacts between pairs of amino acid residues. More surprisingly, the modern contact prediction approaches produce predictions that are sufficiently accurate to successfully deduce the native fold of the protein [38]. It has long been known that native contacts can be used to reliably reconstruct native protein 3D structure [39]. Residue-residue contacts can be visualized in a contact map which is a binary $L \times L$ matrix, with L being protein length. For two residues, i and j, the binary element in the matrix $C(i, j)$ is

$$C(i,j) = \begin{cases} 1, & \text{if } \Delta C_\beta < T \\ 0, & \text{otherwise} \end{cases} \tag{1.1}$$

where ΔC_β is the euclidean distance between C_β atoms (C_α for glycine) of residues i and j and T is a distance threshhold (typically 8 $\mathring{A}$). Figure 1.2 shows an example of a residue-residue contact map generated from a small protein domain.

Eventhough a contact map provides only a 2D represenation of the protein structure, it retains the full 3D structural information of a protein. While it has been shown that only a small subset of native contacts is sufficient to allow accurate modelling of the protein structure, the quality of predicted residue-residue contacts crucially controls the quality of the final structural model [40,41]. The last years have seen an enourmous wealth of studies applying predicted residue-residue contacts not only as distance constraints for *de novo* modelling of protein structures, but also in many different fields in structural biology, such as domain prediction [42], studying alternative conformations [43] or mutational landscapes [44]. The next chapter gives an introduction to state-of-the-art contact prediction approaches, how the predicted residue-residue contacts are applied and which challenges the current methods have to face. The aim of this thesis is therefore to improve the models for residue-residue contact prediction by developing a Bayesian framework that addresses some of these challenges.

2

Introduction to 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 [45]. 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 [46]. 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. Other physico-chemical quantities such as amino acid charge or hydrogen bonding capacity can also induce compensatory effects[47]. 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 2.1).

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

2.1 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.

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 [45,47–50].

Many methods have been developed that are rooted in information theory and use MI measures to describe the dependencies between sites in the alignment [51–53]. Phylogenetic and entropic biases have been identified as strong sources of noise that confound the true coevolution signal [53–55]. Different variants of MI based approaches address these effects and improve on the signal-to-noise ratio [54,56,57]. The most prominent correction for background

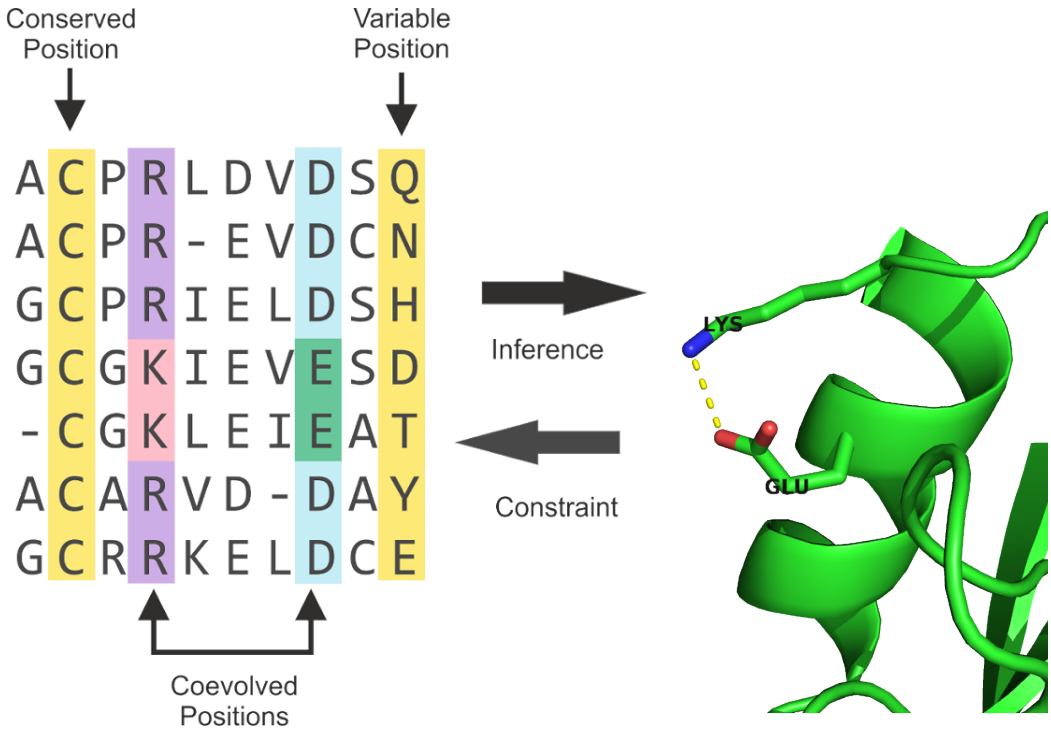

Figure 2.1: The evolutionary record of a protein family reveals evidence of compensatory mutations between spatially neighboring residues that are under selective pressure with respect to some physico-chemical constraints. Mining protein family sequence alignments for residue pairs with strong coevolutionary signals using statistical methods allows inference of spatial proximity for these residue pairs.

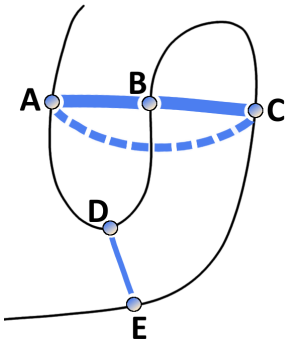

Figure 2.2: Effects of chained covariation obscure signals from true physical interactions. Consider residues A through E with physical interactions between the residue pairs A-B, B-C and D-E. The thickness of blue lines between residues reflects the strength of statistical dependencies between the corresponding alignment columns. Strong statistical dependencies between residue pairs (A,B) and (B,C) can induce a strong dependency 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.

noises is APC that is still used by many modern methods and is discussed in section 2.4.4 [58]. 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 [59,60].

The traditional covariance approaches suffered from high false positive rates because of their inability to cope with transitive effects that arise from chains of correlations between multiple residue pairs [61–63]. The concept of transitve effects is illustrated in Figure 2.2. 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 become even larger than signals of other directly interacting pairs (D,E) and thus lead to false predictions [62].

Local statistical methods consider residue pairs independent of one another which is why they cannot distinguish between direct and indirect correlation signals. In contrast, global statistical models presented in the next section learn a joint probability distribution over all residues allowing to disentangle transitive effects [62,63]. Eventhough local statistical methods cannot compete with modern predictors, *OMES* and MI based scores often serve as a baseline in performance benchmarks for contact prediction [64,65].

2.2 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 [61,63].

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 [61]. 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 2.4 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 [62,66]. 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 2.4.3.1.

2.3 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* [67], *SVM-SEQ* [68]), random forests (*ProC_S3* [69], *TMhhcp* [70], *PhyCMap* [71]), neural networks (*NETCSS* [72], *SAM* [73], [74], *SPINE-2D* [75], *NNCon* [76]) deep neural networks (*DNCon* [77], *CMAPpro* [78]) and ensembles of genetic algorithm classfiers (*GaC* [79]).

Different contact predictors, especially when rooted in distinct principles like sequence-based and coevolution methods, provide orthogonal information on the likelihood that a pair of residues makes a contact [67,80]. 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 sequence features and predictions from the coevolution methods *PSICOV* and *plmDCA* [81]. In a follow-up version *PSICOV* has been replaced with *gaussianDCA* and the sequence-based method *PhyCMap* [82]. *EPC-MAP* was published in 2014 integrating *GREMLIN* as a coevolution feature with physicochemical information from predicted ab initio protein structures [83]. In 2015, *MetaPSICOV* was released combining predictions from *PSICOV*, *mfDCA* and *CCMpred* with other sequence derived feautures [84]. *RaptorX* uses *CCMpred* as coevolution feature and other standard contact prediction features within an ultra-deep neural network [85]. 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 [86,87].

Another conceptual advancement besides the combination of sources of information is based on the fact that contacts are not randomly or independently distributed. DiLena and colleagues found that over 98% of long-range contacts (sequence separation > 24 positions) are in close proximity of other contacts, compared to 30% for non-contacting pairs [78]. The distribution of contacts is governed by local structural elements, like interactions between helices or β-sheets, leading to characteristic patterns in the contact map that can be recognised [88]. Deep learning provides the means to model higher level abstractions of data and several methods apply multi-layered algorithms to refine predictions by learning patterns that reflect the local neighborhood of a contact [78,84,85,89].

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 [90]. 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 [85,86].

2.4 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 [91,92], 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. Each sequence $\mathbf{x}_n = (\mathbf{x}_{n1}, ..., \mathbf{x}_{nL})$ is of length L and every position constitutes a categorical variable x_i that can take values from an alphabet indexed by $\{0, ..., 20\}$, where 0 stands for a gap and $\{1, ..., 20\}$ stand for the 20 types of amino acids. The measured constraints are given by the empirically observed single and pairwise amino acid frequencies that can be calculated as

$$f_i(a) = f(x_i = a) = \frac{1}{N} \sum_{n=1}^{N} I(x_{ni} = a) \;, \tag{2.1}$$

$$f_{ij}(a, b) = f(x_i = a, x_j = b) = \frac{1}{N} \sum_{n=1}^{N} I(x_{ni} = a, x_{nj} = b) \;. \tag{2.2}$$

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

$$\begin{aligned} f(x_i = a) &\equiv p(x_i = a) \\ &= \sum_{\mathbf{x\prime}_1, \ldots, \mathbf{x\prime}_L = 1}^{q} p(x\prime) I(x\prime_i = a) \end{aligned} \tag{2.3}$$

$$\begin{aligned} f(x_i = a, x_j = b) &\equiv p(x_i = a, x_j = b) \\ &= \sum_{\mathbf{x\prime}_1, \ldots, \mathbf{x\prime}_L = 1}^{q} p(x\prime) I(x\prime_i = a, x\prime_j = b) \;. \end{aligned} \tag{2.4}$$

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. (2.4) by introducing Lagrange multipliers $\mathbf{w}_{ij}$ and v_i, results in the formulation of the *Potts model*,

$$p(\mathbf{x}|\mathbf{v},\mathbf{w}) = \frac{1}{Z(\mathbf{v},\mathbf{w})} \exp\left(\sum_{i=1}^{L} v_i(x_i) \sum_{1\leq i<j\leq L}^{L} w_{ij}(x_i,x_j)\right) . \tag{2.5}$$

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 $\mathbf{x}$,

$$Z(\mathbf{v},\mathbf{w}) = \sum_{\mathbf{x\prime}_1,\ldots,\mathbf{x\prime}_L=1}^{q} \exp\left(\sum_{i=1}^{L} v_i(x_i) \sum_{1\leq i<j\leq L}^{L} w_{ij}(x_i,x_j)\right) . \tag{2.6}$$

2.4.1 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 [93,94].

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.

2.4.1.1 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. (2.6).

The overparametrization is referred to as *gauge invariance* in statistical physics literature and can be eliminated by removing parameters [63,95]. 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. [63] imposed by the restraints,

$$\sum_{a=1}^{q} v_{ia} = \sum_{a=1}^{q} w_{ijab} = \sum_{a=1}^{q} w_{ijba} = 0 \tag{2.7}$$

for all i, j, b or the *lattice-gas gauge* used by Morcos et al [95] and Marks et al [38] imposed by restraints

$$\mathbf{w}_{ij}(q, a) = \mathbf{w}_{ij}(a, q) = v_i(q) = 0 \tag{2.8}$$

for all i, j, a [96].

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 [97–99].

2.4.2 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 $\mathbf{X}$:

$$\begin{aligned} \mathrm{LL}(\mathbf{v}, \mathbf{w}|\mathbf{X}) &= \sum_{n=1}^{N} \log p(\mathbf{x}_n) \\ &= \sum_{n=1}^{N} \left[\sum_{i=1}^{L} v_i(x_{ni}) + \sum_{1 \le i < j \le L} w_{ij}(x_{xn}, x_{nj}) - \log Z \right] \end{aligned} \tag{2.9}$$

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 2.9). 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,

$$\begin{aligned} R(\mathbf{v}, \mathbf{w}) &= \log \left[\mathcal{N}(\mathbf{v}|\mathbf{0}, \lambda_v^{-1} I) \mathcal{N}(\mathbf{w}|\mathbf{0}, \lambda_w^{-1} I) \right] \\ &= -\frac{\lambda_v}{2} ||\mathbf{v}||_2^2 - \frac{\lambda_w}{2} ||\mathbf{w}||_2^2 + \text{const.} , \end{aligned} \tag{2.10}$$

where the strength of regularization is tuned via the regularization coefficients λ_v and λ_w [100–102].

However, optimizing the log-likelihood requires computing the partition function Z given in eq. (2.6) 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.

2.4.3 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 [61]. In 2002 they applied their idea to 11 small proteins using an iterative Monte Carlo procedure to obtain estimates of the model parameters and achieved an increase in accuracy of 10-20% compared to the local statistical models [103]. 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 [63]. 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 [104]. 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 [102].

Also in 2011, Morcos et al. introduced a naive mean-field inversion approximation to the partition function, named *mfDCA* [95]. 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 [38].

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* [98].

A related approach to mean-field approximation is sparse inverse covariance estimation, named *PSICOV*, developed by Jones et al. (2012) [65]. PSICOV uses an L1-regularization, known as graphical Lasso, to invert the correlation matrix and learn a sparse graphical model [105]. 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 [106].

Another related approach to *mfDCA* and *PSICOV* is *gaussianDCA*, proposed in 2014 by Baldassi et al. [107]. 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 implementa-

tions of pseudo-likelihood maximization that vary in slight details, perform similarly and thus are equally popular in the community, such as CCMpred [100], plmDCA[101] and GREMLIN [102].

2.4.3.1 Maximum Likelihood Inference for Pseudo-Likelihood

The pseudo-likelihood is a rather old estimation principle that was suggested by Besag already in 1975 [108]. 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:

$$\begin{aligned} p(\mathbf{x}|\mathbf{v},\mathbf{w}) &\approx \prod_{i=1}^{L} p(x_i|\mathbf{x}_{\setminus xi},\mathbf{v},\mathbf{w}) \\ &= \prod_{i=1}^{L} \frac{1}{Z_i} \exp\left(v_i(x_i) \sum_{1\leq i<j\leq L} w_{ij}(x_i,x_j)\right) \end{aligned} \tag{2.11}$$

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

$$Z_i = \sum_{a=1}^{q} \exp\left(v_i(a) \sum_{1\leq i<j\leq L} w_{ij}(a,x_j)\right) \tag{2.12}$$

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,

$$\begin{aligned} \mathrm{pLL}(\mathbf{v},\mathbf{w}|\mathbf{X}) &= \sum_{n=1}^{N}\sum_{i=1}^{L} \log p(x_i|\mathbf{x}_{\setminus xi},\mathbf{v},\mathbf{w}) \\ &= \sum_{n=1}^{N}\sum_{i=1}^{L} \left[v_i(x_{ni}) + \sum_{j=i+1}^{L} w_{ij}(x_{ni},x_{nj}) - \log Z_{ni}\right] , \end{aligned} \tag{2.13}$$

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

$$\hat{\mathbf{v}},\hat{\mathbf{w}} = \underset{\mathbf{v},\mathbf{w}}{\operatorname{argmax}}\ \mathrm{pLL}(\mathbf{v},\mathbf{w}|\mathbf{X}) + R(\mathbf{v},\mathbf{w}) \ . \tag{2.14}$$

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 [94,97–99]. 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 [97,108,109]. That is, as the number of sequences in the alignment increases, pseudo-likelihood estimates converge towards the true full likelihood parameters.

2.4.4 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 2.4.3 heuristically map the 21×21 dimensional coupling matrix $\mathbf{w}_{ij}$ to a single scalar quantity.

mpDCA [63] and *mfDCA* [38,95] 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. Most pseudo-likelihood methods (*plmDCA* [98,101], *CCMpred* [100], *GREMLIN* [102]) compute the *Frobenius norm* of the coupling matrix $\mathbf{w}_{ij}$ to obtain a scalar contact score C_{ij},

$$C_{ij} = ||\mathbf{w}_{ij}||_2 = \sqrt{\sum_{a,b=1}^{q} w_{ijab}^2} \, . \tag{2.15}$$

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 [98,107,110]. *PSICOV* [65] 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* [63]. Therefore, the coupling matrices should be transformed to *zero-sum gauge* before computing the Frobenius norm

$$\mathbf{w}'_{ij} = \mathbf{w}_{ij} - \mathbf{w}_{ij}(\cdot, b) - \mathbf{w}_{ij}(a, \cdot) + \mathbf{w}_{ij}(\cdot, \cdot) \, , \tag{2.16}$$

where $\cdot$ denotes average over the respective indices [98,100,101,107].

Another commonly applied heuristic known as APC has been introduced by Dunn et al. in order to reduce background noise arising from correlations between positions with high entropy or phylogenetic couplings [58]. APC is a correction term that is computed from the raw contact map as the product over average row and column contact scores $\overline{C_i}$ divided by the average contact score over all pairs $\overline{C_{ij}}$. The corrected contact score C_{ij}^{APC} is obtained by subtracting the APC term from the raw contact score C_{ij},

$$C_{ij}^{APC} = C_{ij} - \frac{\overline{C_i} \, \overline{C_j}}{\overline{C_{ij}}} \, . \tag{2.17}$$

Visually, APC creates a *smoothing* effect on the contact maps that is illustrated in Figure 2.3 and it has been found to substantially boost contact prediction performance [58,102]. It was first adopted by *PSICOV* [65] but is now used by most methods to adjust raw contact scores.

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 [111]. 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.

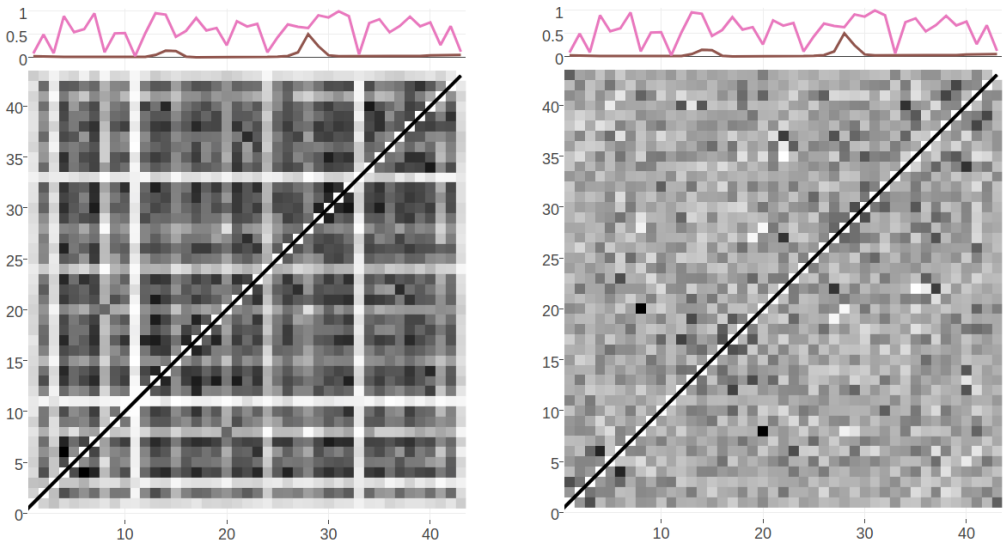

Figure 2.3: Contact maps computed from pseudo-likelihood couplings. Subplot on top of the contact maps illustrates the normalized Shannon entropy (pink line) and percentage of gaps for every position in the alignment (brown line). **Left**: Contact map computed with Frobenius norm as in eq. (2.15). Overall 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. (2.17).

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 an 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 [111] discussed above, estimate a background model from the final contact matrix, thus depending on the statistical model used to infer the contact matrix.

2.5 Applications

The most popular and historically motivated application for contact prediction is contact-guided *de novo* structure prediction.

It has long been known that the native protein 3D structure can be reconstructed from an error-free contact map [39]. Also, protein fold reconstruction from sparse inter-residue proximity constraints obtained from experiments such as cross-linking/mass spectrometry, Foerster resonance energy transfer (FRET) or sparse nuclear Overhauser enhancement (NOE) distance data generated from NMR experiments has been demonstrated [32,112–116]. Predicted contacts, however, have long been regarded as being of little use for structure prediction because of their high false-positive rates [117,118]. Only with the emergence of global statistical models for contact prediction which drastically reduced false-positive rates there has been renewed interest in *de novo* structure prediction aided by predicted contacts. In 2011, Marks et al. showed that the top scoring contacts predicted with their mean-field approach

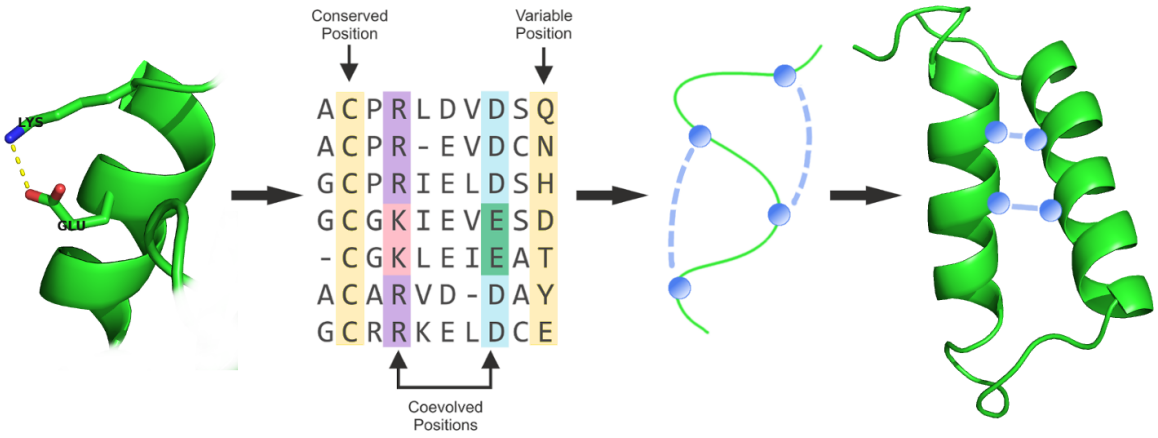

Figure 2.4: Generalized structure prediction pipeline integrating predicted contacts in form of distance constraints that guide conformational sampling.

mfDCA are sufficiently accurate to successfully deduce the native fold of the protein [38]. In the following years, methods to predict contacts have been improved and applied to model many more protein structures culminating in the high-throughput prediction of 614 protein structures out of which more than 100 represent novel folds by Ovchinnikov and colleagues in 2017 [119–127].

Many contact-guided protocols have been established since, that typically integrate predicted contacts in form of distance constraints into an energy function to guide the conformational sampling process: Unicon3D [128], RASREC [129], RBOAleph [130], GDFuzz3D [131], Pcons-Fold [132], C2S_Pipeline [133], FRAGFOLD + PSICOV [134], FILM3 [135], EVFold [38]. Figure 2.4 presents a generalized structure prediction pipeline using predicted contacts.

The optimal quality of inferred contacts and their effective utilization is still subject to discussion and further research. It has been demonstrated that only a small subset of native contacts is sufficient to produce accurate structural models [39,40,133,136–138]. Sathyapriya and colleagues developed a rational strategy to select important native contacts and successfully reconstructed the structure to near native resolution with only 8% of contacts [136]. Kim and colleagues formulated that only one correct contact for every 12 residues in the protein is sufficient to allow accurate topology level modeling given that the contacts are nonlocal and broadly distributed [40]. These studies emphasize that certain contacts are more important than others. Long-range contacts are rare and most informative for protein structure prediction because they define the overal fold and packing of tertiary structure whereas short-range contacts define local secondary structure [139]. It is a consistent finding that eventhough long-range contacts are of higher relevance than short-range contacts for structure reconstruction, their information alone is not sufficient [134,136,140]. Since a small number of correct residue-residue contacts is sufficient to improve protein structure prediction and many reconstruction protocols can tolerate missing contact information much better than erroneous contact information, it has been stressed that methods development should focus on predicting a small number of high confident contacts [40,41]. Marks and colleagues observed that isolated false positives have a much stronger detrimental effect on structure prediction than false positives close to true contacts [38]. Zhang et al. found that their tool Touchstone II required an accuracy of long-range contact predictions of at least 22% to generate a positive effect to structure prediction [141]. Frequently, folding protocols employ a filtering step to eliminate unsatisfied or conflicting constraints possibly originating from false-positive contacts [142,143]. Generally it is assumed that higher precision of predicted long-range contacts results in improved structural models, albeit there is no strong correlation

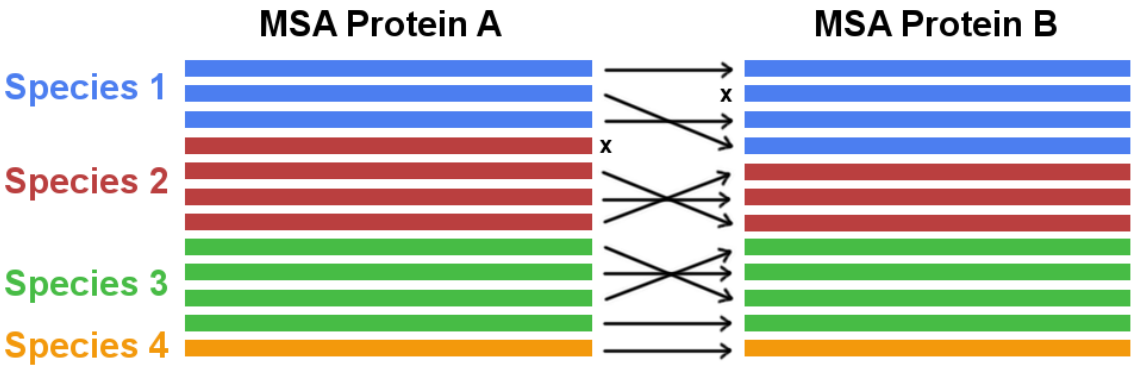

Figure 2.5: Concatenating two multiple sequence alignments. In case multiple paralogs exist for a gene in one species the correct interaction partner needs to be identified and matched (marked with arrows). Sequences that cannot be paired with a unique interaction partner need to be discarded (marked with x).

as model quality depends on many other factors such as the secondary structure composition of the protein, the domain size, the usage of additional sources of structural information, the type of distance constraint function and the particular structure reconstruction protocol [38,134,139,141,144].

Coevolution has not only been studied for residues pairs within a protein but also for residue pairs across protein–protein interfaces [120,121,126,145,146]. Eventhough the methodology of detecting coevolving amino acid pairs from the MSA is the same, a new challenge arises for the correct identification of orthologous interacting partners. Without the correct pairing of interacting partners for every species the detection of coevolutionary signals would be compromised. However, the generation of a MSA of paired sequences is complicated in the presence of multiple paralogs of a gene in a single genome. The problem of paralog matching is visualized in Figure 2.5. For prokaryotes, sequence paires are typically identified by exploiting the bacterial gene organisation in form of operons, i.e. co-localized genes will be co-expressed and are more likely to physically interact. Co-localisation of genes has also been applied to match genes from eurkaryotes, assuming that Uniprot accession numbers can be used as a proxy for genomic distances [146]. New strategies have been developed based on the idea that an alignment with correctly matched paralogs will maximize the coevolution score [147,148].

A related objective is the study of the oligomerization status of proteins. The study of homo-oligomers is simplified in the sense that the identical protein sequence of both interaction partners renders the concatenation of two MSAs unnessary and allows to work with one MSA. A different challenge lies in the correct distinction between the physical contacts of the monomeric structure and the interprotein contacts. With the availability of monomeric structural data the idea is to filter out those high scoring contacts that form contacts in the monomeric structure or are located in the protein core. The remaining high scoring false positive contacts at the surface of the protein are potential contacts at the interface that can be incorporated into a docking protocol to drive complex formation [149,150].

Contacts are also used to analyse potential alternative conformations of proteins [43,151–155]. Coevolutionary analysis detects all evolutionarily significant residue–residue correlations, regardless of whether the interaction is formed in a transient state of the protein or its stable form. Therefore, predicted contact maps might capture multiple states of a protein, since they are of functional importance and thus under evolutionary pressure. Sfriso and colleagues developed an automated pipiline that introduces filtered predicted contacts as ensemble re-

straints into a molecular dynamics simulations and is able to detect alternative relevant conformational states [151].

Quality assessment of structural models, involving model selection and ranking, is a crucial task in structural biology. Predicted residue-residue contacts can indicate the best protein structure among a set of properly folded and misfolded structures by counting the number of satisfied contacts [118,156]. Besides ranking of models, predicted contacts have been used as features for training machine learning methods that predict the global quality of a structural model [157,158].

As mentioned before, methods for protein fold reconstruction from experimental distance constraints have been successfully applied for many years. Several integrative approaches have been developed that combine complementary sources of sparse structural constraints, including predicted contacts, to accurately determine protein structure [36,37].

Sadowski used predicted contacts to parse domain boundaries based on the simple idea that contacts are more abundant within domains than between domains [42].

Eventhough the coevolutionary methods have been developed for proteins, they have been successfully applied to analyse nucleotide coevolution and to predict RNA tertiary structures with the help of predicted nucleotide-nucleotide contacts [159–161]. Much less RNA sequences are required compared to protein sequences in order to extract statistically significant signals because of the reduced number of model parameters when working with a four letter alphabet (compared to a 20 letter alphabet with proteins). On the downside, alignment errors resulting from the complicated determination of RNA multiple sequence alignments limits the accuracy of coevolution analysis [161]. Despite the diminished accuracy, predicted nucleotide contacts have been demonstrated to improve RNA structure prediction over conventional methods [160].

The stastistical models used for coevolution analysis provide information about which residue pairs are important in evolution for folding or functional constraints. They can be used to assign probabilities to sequences that reflect the overal compliance of a sequence with the protein family under study and thereby provide quantitative predictions of mutational effects [44,162,163]. Computational screening of mutational effects can support and complement the costly and time-consuming directed evolution or mutational screening experiments [44]. With a similar idea in mind, the coevolution models have been applied to sequences of human immune repertoires [164,165]. Antibody affinity maturation can be viewed as a Darwinian process with the affinity to the target antigen being the main fitness criterion. Therefore, given the model representing the antibody sequence family, the probability for a sequence reflects the binding affinity to the target antigen. Quantifying the effect of mutations is also helpful for protein design. Coevolving positions might be of particular interest as hotspots for engineering protein stability or functional specificity because they determine positions relevant to protein structure and function [166].

Skwark and colleagues applied the popular coevolution statistical models to genomes and developed a statistical method called *genomeDCA* [167]. They are able to identify coevolving polymorphic locus pairs based on the idea that the corresponding proteins form protein-protein interactions that are under strong evolutionary pressure. In a case study on two large human pathogen populations they found that three quarters of coevolving loci are located in genes that determine beta-lactam (antibiotic) resistence.

Fox and colleagues turn the idea of DCA upside down. They developed a benchmark for testing the accuracy of large MSAs by evaluating the agreeement between the predicted and the native contacts [168]. Based on the assumption that better alignments provide more accurate contact predictions, the alignment quality is inferred from the precision of predicted contacts.

2.6 Evaluating Contact Prediction Methods

Choosing an appropriate benchmark for contact prediction is determined by the further utilization of the predictions. Most prominently, predicted contacts are used to assist structure prediction as outlined in the last section 2.5. 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 already discussed. From these considerations emerged various standard benchmarks that have been established by the CASP community over many years [90,169,170]. 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_α 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,

$$\text{precision} = \frac{TP}{TP + FP} \,, \tag{2.18}$$

where TP is a true positive contact and FP is false positive contact. 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 [171]. Another informative metric is mean error defined as:

$$\text{mean error} = \frac{\text{error}}{TP + FP} \begin{cases} error = \Delta C_\beta - T & \text{if } \Delta C_\beta > T \\ error = 0, & \text{otherwise} \end{cases} \tag{2.19}$$

where ΔC_β is the actual distance of a residue pair in the native structure, and T is the distance threshold defining a true contact. The mean error helps to asses how wrong false positive predictions are. 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 [90].

Currently best methods perform in the range XXX. Sequence feature based methods: 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 [71,172]. TODOOOPLOT

2.6.1 Sequence Separation

Local residue pairs separated by only some positions in sequence (e.g $|i - j| < 6$) are usually filtered out for evaluating contact prediction methods. They are trivial to predict as they typically correspond to contacts within secondary structure elements and reflect the local

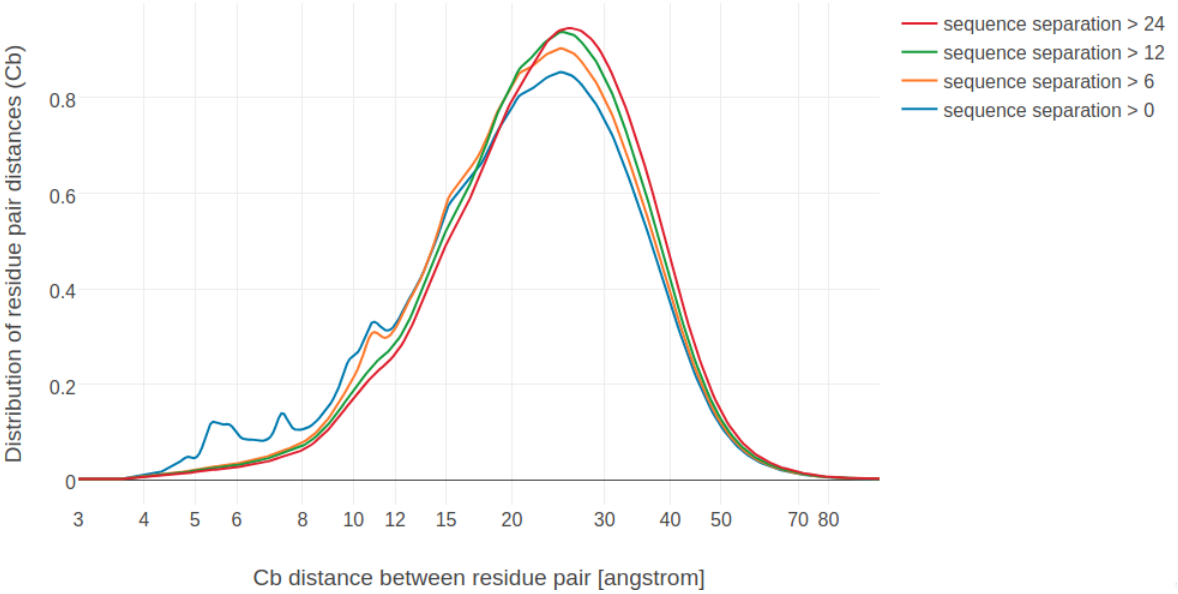

Figure 2.6: Distribution of residue pair C_β distances over 6741 proteins in the dataset (see Methods 7.1) at different minimal sequence separation thresholds.

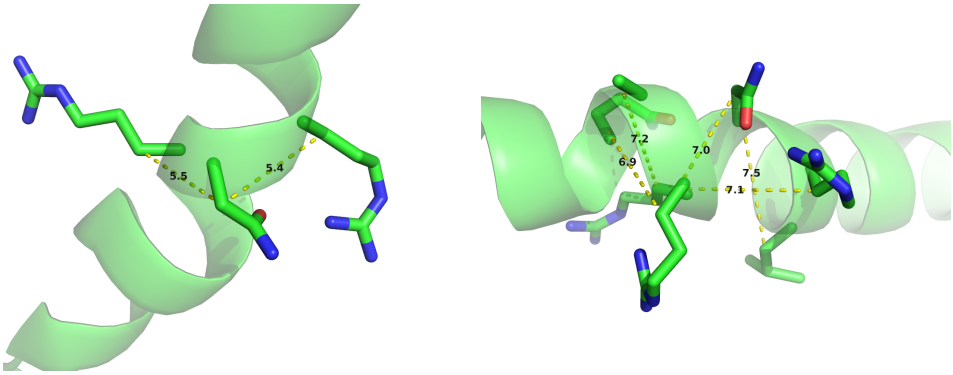

Figure 2.7: C_β distances between neighboring residues in α-helices. Left: Direct neighbors in α-helices have C_β distances around $5.4\mathring{A}$ due to the geometrical constraints from α-helical architecture. Right: Residues separated by two positions ($|i-j|=2$) are less geometrically restricted to C_β distances between $7\mathring{A}$ and $7.5\mathring{A}$.

geometrical constraints. Figure 2.6 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 2.7).

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 [90,170]. 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 [169].

2.6.2 Interpretation of Evaluation Results

There are certain subtleties to be considered when interpreting contact prediction evaluation results.

The rigid C_β distance definition of a contact is a very rough measure of true physical interactions between amino acid sidechains. More importantly, interactions between sidechains depend on their physico-chemical properties, on their orientation and different environments within proteins (see section **??**) [173]. 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 [41]. Anishchenko and colleagues 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 [174].

Another issue concerns structural variation within a protein family. Evolutionary couplings are inferred from all family members in the MSA and therefore predicted contacts 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 [174]. Therefore, choosing the right trade-off between sensitivity and specificity when generating alignments is a crucial step as well as choosing the target protein structure for evaluation.

Finally, an important aspect not considered in the standard benchmarks is the spread of predicted contacts. It is perfectly possible to improve precison of predicted contacts without translating this improvement to better structural models. The reason being that structurally redundant contacts, that is contacts in the immediate sequence neighborhood of other contacts, do not give additional information to constrain the fold [38,40,80]. For example, given a contact between residues i and j, there is hardly an added value knowing that there is a contact between residues $i+1$ and $j+1$ when it comes to predicting the overal topology. This observation is highly relevant for deep learning methods due to their unique ability to abstract higher order interactions and recognize contact patterns. Several measures of the contact spread have been developed, like the mean euclidian distance between true and predicted contacts, but are not commonly evaluated yet [38,144].

2.7 Challenges for Coevolutionary Inference

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

2.7.1 Phylogenetic Effects as a Source of Noise

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 [95,96,175]. 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 [57,61,62]. Figure 2.8 illustrates a simplified example, where dependence of sequences due to phylogeny leads to a covariation signal. To reduce the

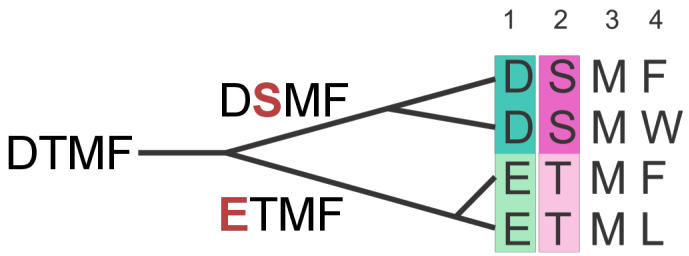

Figure 2.8: The phylogenetic dependence of closely related sequences can produce covariation signals. Here, two independent mutation events (highlighted in red) in two branches of the tree result in a perfect covariation signal for two positions.

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 7.3) [65,95,96,176].

2.7.2 Entropic Effects as a Source of Noise

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 [62]. 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 co-evolution signals from positions with high entropy and underestimated signals for highly conserved interactions [55]. Several heuristics have been proposed to reduce entropy effects, such as Row-Column-Weighting (RCW) [57] or Average Product Correction (APC) [58] (see section 2.4.4).

2.7.3 Finite Sampling Effects

Spurious correlations can arise from random statistical noise and blur true co-evolution 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 [102,175]. 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 [174].

Interesting targets for contact prediction are protein families without any associated structural information. As can be seen in Figure 2.9, 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 [102,177]. Nevertheless, because of the already mentioned sequencing biases, better and more sensitive statistical models are indespensible to extend the applicability domain of coevolutionary methods.

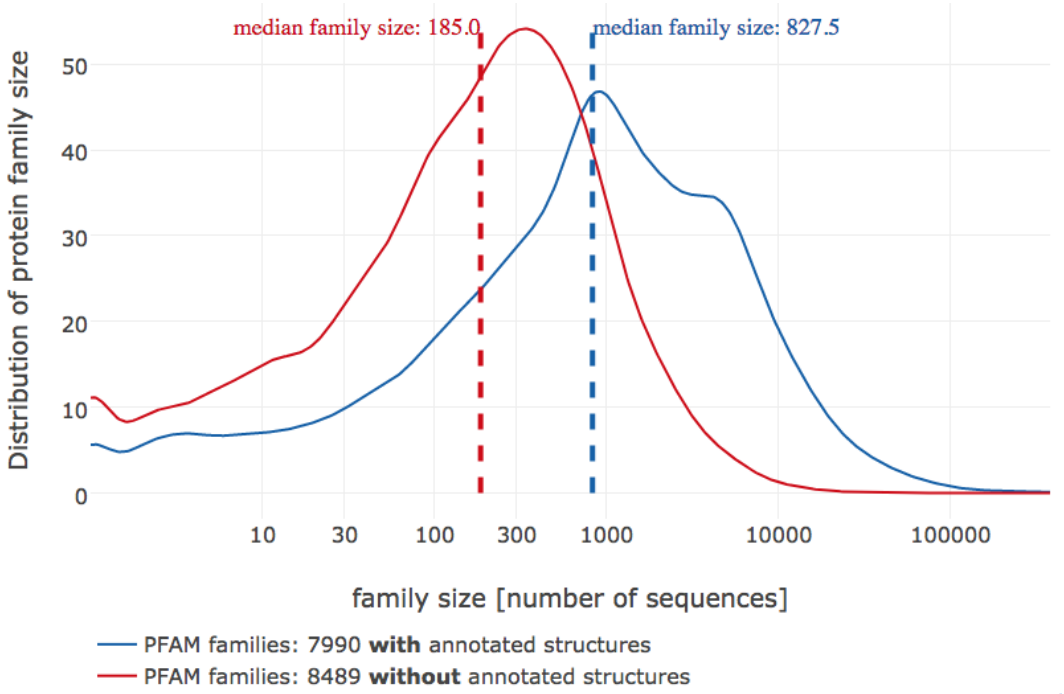

Figure 2.9: Distribution of PFAM family sizes. Less than half of the families in PFAM (7990 compared to 8489 families) do not have an annotated structure. The median family size in number of sequences for families with and without annotated structures is 185 and 827 respectively. Data taken from PFAM 31.0 (March 2017, 16712 entries) [178].

2.7.4 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 [179]. 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 [174]. Therefore, MSAs need to be generated with great care and covariation methods need to be tailored to these specific types of proteins [180,181].

Furthermore, 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 [171,182]. However, deep alignments can capture coevolutionary signals from different subfamilies [149]. 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 [166]. Therefore, a trade-off between specificity and diversity of the alignment is required to reach optimal results [119].

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 coevolution models for contact prediction treat gaps as the 21st amino acid. This introduces an imbalance as gaps and amino acids express different behaviours which can result in gap-induced artefacts [110].

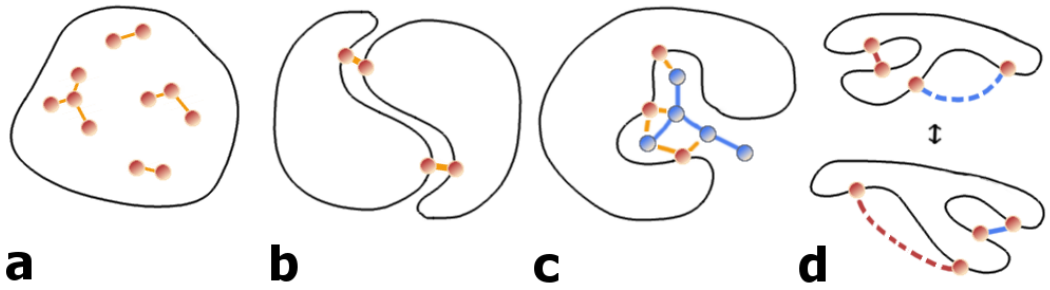

Figure 2.10: Possible sources of coevolutionary signals. **a)** Physical interactions between intra-domain residues. **b)** Interactions across the interface of predominantly homo-oligomeric complexes. **c)** Interactions mediated by ligands or metal atoms. **d)** Transient interactions due to conformational flexibility.

2.7.5 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 2.10) [175]. 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 (Figure 2.10 b). Currently it is hard to reliably distinguish intra- and inter-molecular contacts [149]. 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 [174]. 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 [119,125,149,152,153,183] or used co-evolutionary signals to predict homo-dimeric complexes [150].

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 (Figure 2.10 c) [176,184]. Confirming this hypothesis, a study showed that the cumulative strength of couplings for a particular residue can be used to predict functional sites [119,175].

Another important aspect is conformational flexibility (Figure 2.10 c). 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 [185]. Several studies demonstrated successfully that coevolutionary signals can capture interactions specific to different distinct conformations [95,119,151,153].

3

Interpretation of Coupling Matrices

Contact prediction methods learning a *Potts model* for the MSA of a protein familiy, map the inferred 20 x 20 dimensional coupling matrices w_{ij} onto scalar values to obtain contact scores for each residue pair as outlined in section 2.4.4. As a result, the full information contained in coupling matrices is lost, such as the contribution of individual couplings w_{ijab}, whether a coupling is positive or negative, higher order dependencies between couplings or possibly biological meaningful signals. The following sections give some intuition for the information contained in coupling matrices.

3.1 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 2.4.4, a contact score $C_{i,j}$ for a residue pair (i, j) is commonly computed as the Frobenius norm over the coupling matrix, $C_{i,j} = ||\mathbf{w}_{ij}||_2 = \sqrt{\sum_{a,b=1}^{20} {w_{ijab}}^2}$.

The plots in Figure 3.1 show the correlation of squared coupling values ${w_{ijab}}^2$ with binary contact class (contact=1, non-contact=0) and the standard deviation of squared coupling values ${w_{ijab}}^2$ for contacts computed on a dataset of 100.000 residue pairs per class (for details see methods section 7.7.1). All couplings have a weak positive class correlation, meaning the stronger the squared coupling value, the more likely a contact can be inferred. Correlation is weak because most couplings w_{ijab} are close to zero since typically only few amino acid pairings per residue pair carry evidence and produce a signal. Generally, couplings that involve an aliphatic amino acid such as isoleucine (I), leucine (L), valine (V) or an alanine (A) express the strongest class correlation. In contrast, cysteine pairs (C-C) or pairs involving only the charged residus arginine (R), glutamic acid (E), lysine (K) or aspartic acid (D) correlate only weakly with contact class. Interestingly, for residue pairs being in physical contact, C-C and couplings involving charged residues have the highest standard-deviation among all couplings as can be seen in the right plot in Figure 3.1. Standard deviation of squared coupling values from non-contacts shows no relevant patterns and is on average one magnitude smaller than for the contact class (see Appendix Figure D.1).

Different couplings are of varying importance for contact inference and have distinct characteristics. When looking at the raw coupling values (without squaring), these charateristics become even more pronounced. The plots in Figure 3.2 show the correlation of raw coupling

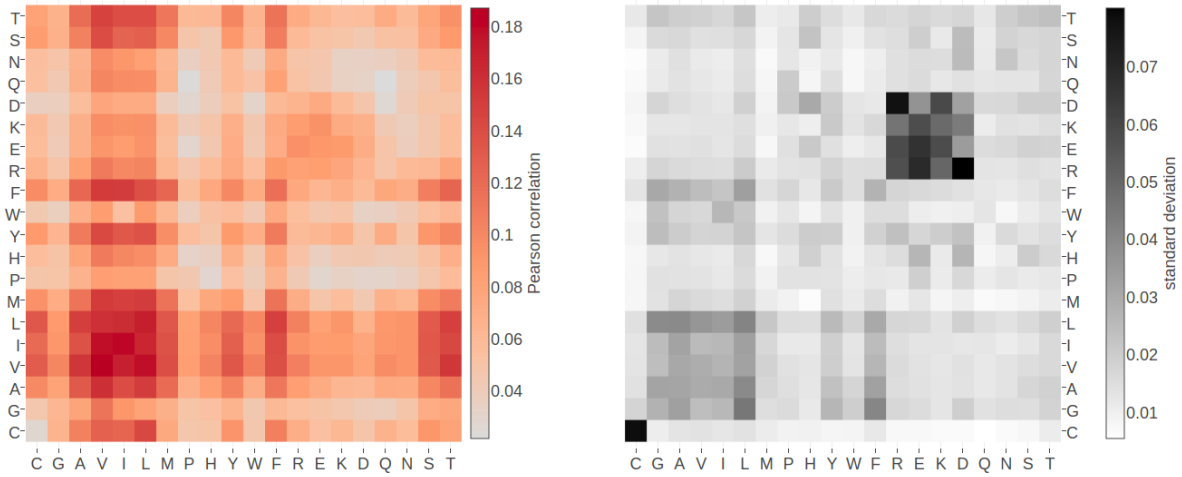

Figure 3.1: **Left** Pearson correlation of squared coupling values $(w_{ijab})^2$ with contact class (contact=1, non-contact=0). **Right** Standard deviation of squared coupling values for residue pairs in contact. Dataset contains 100.000 residue pairs per class (for details see methods section 7.7.1). Amino acids are abbreviated with one-letter code and they are broadly grouped with respect to physico-chemical properties listed in Appendix B.

values w_{ijab} with contact class and the standard deviation of coupling values for contacts. Standard deviation of coupling values for non-contacts shows no relevant patterns and is on average half as big as for the contact class (see Appendix Figure D.2). Interestingly, in contrast to the findings for squared coupling values, couplings for charged residue pairs, involving arginine (R), glutamic acid (E), lysine (K) and aspartic acid (D), have the strongest class correlation (positive and negative), whereas aliphatic coupling pairs correlate to a much lesser extent. This implies that squared coupling value is a better indicator of a contact than the raw signed coupling value for aliphatic couplings. On the contrary, the raw signed coupling values for charged residue pairs are much more indicative of a contact than the magnitude of their squared values. Raw couplings for cysteine (C-C) pairs, proline (P) and tryptophane (W) correlate only weakly with contact class. For these pairs neither a squared coupling value nor the raw coupling value seems to be a good indicator for a contact.

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.

3.2 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. These patterns allow a biological interpretation of the coupling values that reveal details of the physico-chemical interdependency between both residues.

Figure 3.3 visualizes the inferred coupling matrix and single potentials v_i and v_j for a residue pair (i, j) computed with the pseudo-likelihood method. The single potentials v_{ia} and v_{ja} describe the tendency for each amino acid a to appear at positions i and j, and the couplings w_{ijab} describe the tendency of amino acid a at position i to co-occur with amino acid b at position j. A cluster of strong coupling values can be observed for the couplings between the charged residues glutamic acid (E), aspartic acid (D), lysine (K) and arginine (R) and

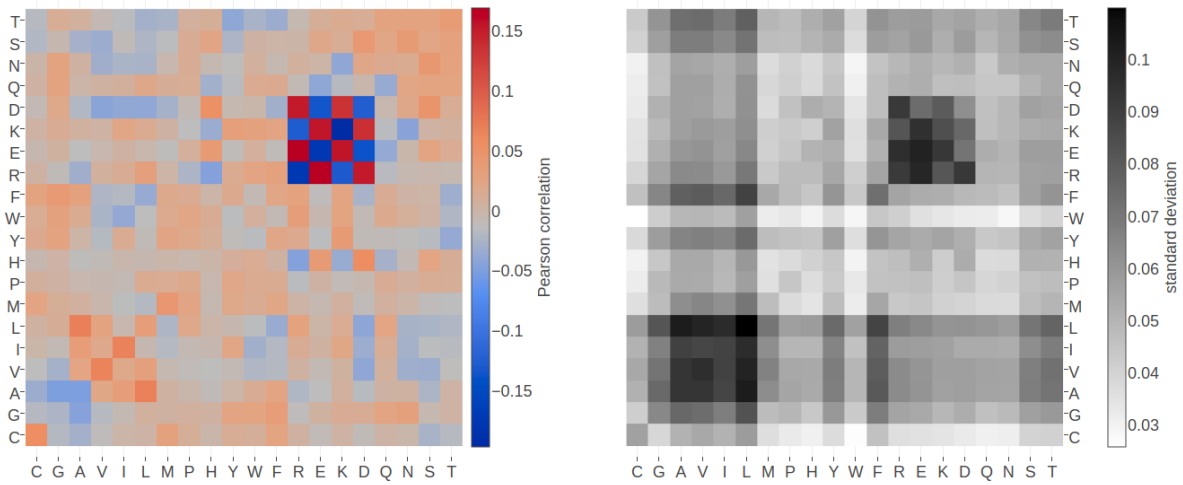

Figure 3.2: **Left** Pearson correlation of raw signed coupling values w_{ijab} with contact class (contact=1, non-contact=0). **Right** Standard deviation of coupling values for residue pairs in physical contact. Dataset contains 100.000 residue pairs per class (for details see section 7.7.1). Amino acids are abbreviated with one-letter code and they are broadly grouped with respect to physico-chemical properties listed in Appendix B.

the polar residue glutamine (Q). Positive coupling values arise between positively charged residues (K, R) and negatively charged residues (E, D), whereas couplings between equally charged residues have negative values. These exemplary couplings (E-R, E-K, K-D) perfectly reflect the interaction preference for residues forming salt bridges. Indeed, in the protein structure the first residue (E) forms a salt bridge with the second residue (R) as can be seen in the left plot in Figure 3.5.

Figure 3.4 visualizes the coupling matrix for a pair of hydrophobic residues. Hydrophobic pairings, such as alanine (A) - isoleucine (I), or glycine (G) - isoleucine (I) have strong coupling values but the couplings also reflect a sterical constraint. Alanine is a small hydrophobic residue and it is favoured at both residue positions: it has strong positive single potentials $v_i(A)$ and $v_j(A)$ and strong positive couplings with isoleucine (I), leucine (L) and methionine (M). But alanine is disfavoured to appear at both positions at the same time since the A-A coupling is negative. Figure 3.5 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 matrices, including pi-cation interactions (see Appendix E.1), aromatic-proline interactions (see Appendix E.3), sulfur-aromatic interactions or disulphide bonds (see Appendix E.2).

Coucke and collegues performed a thorough quantitative analysis of coupling matrices selected from confidently predicted residue pairs [186]. 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.

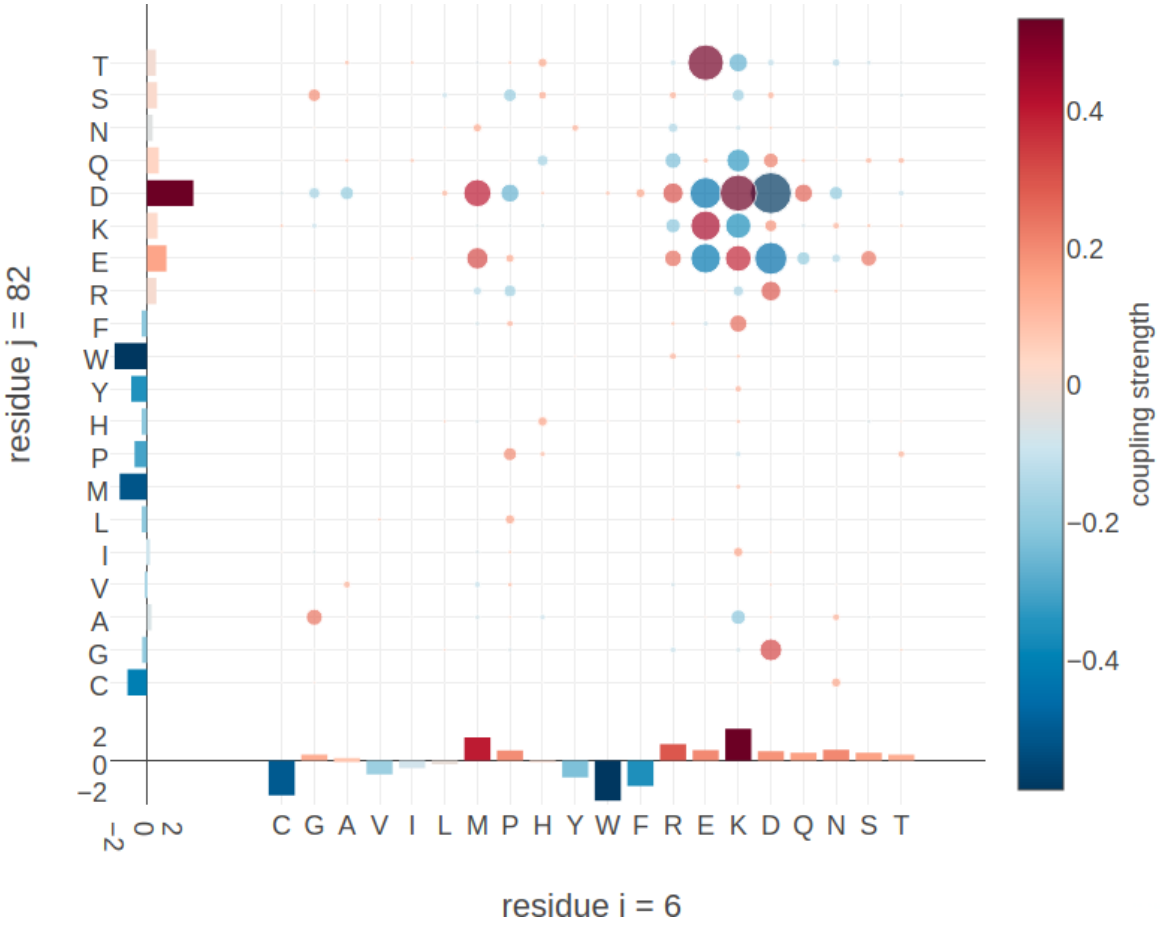

Figure 3.3: Couplings w_{ijab} and single potentials v_{ia} and v_{ja} computed with pseudo-likelihood for residues 6 and 82 in protein chain 1a9x_A_05. The matrix shows the 20x20 couplings w_{ijab} with color representing coupling strength and direction (red = positive coupling value, blue = negative coupling value) and diameter of bubbles representing absolute coupling value $|w_{ijab}|$. Bars at the x-axis and y-axis correspond to the *Potts* model single potentials v_i and v_j respectively. Color reflects the value of single potentials. Amino acids are abbreviated with one-letter code and they are broadly grouped with respect to physico-chemical properties listed in Appendix B.

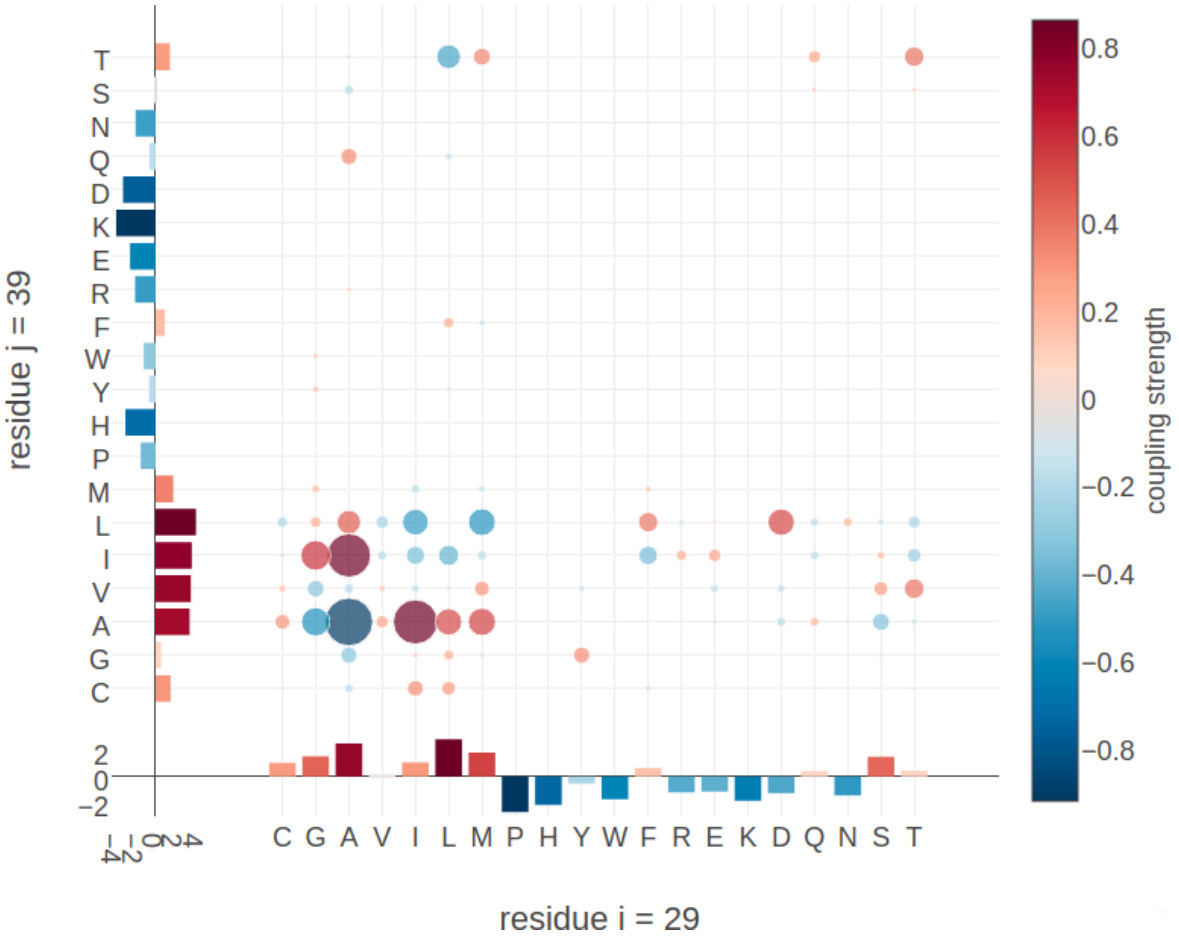

Figure 3.4: Couplings w_{ijab} and single potentials v_{ia} and v_{ja} computed with pseudo-likelihood for residues 29 and 39 in protein chain 1ae9_A_00. The matrix shows the 20x20 couplings w_{ijab} with color representing coupling strength and direction (red = positive coupling value, blue = negative coupling value) and diameter of bubbles representing absolute coupling value $|w_{ijab}|$. Bars at the x-axis and y-axis correspond to the *Potts* model single potentials v_i and v_j respectively. Color reflects the value of single potentials. Amino acids are abbreviated with one-letter code and they are broadly grouped with respect to physico-chemical properties listed in Appendix B.

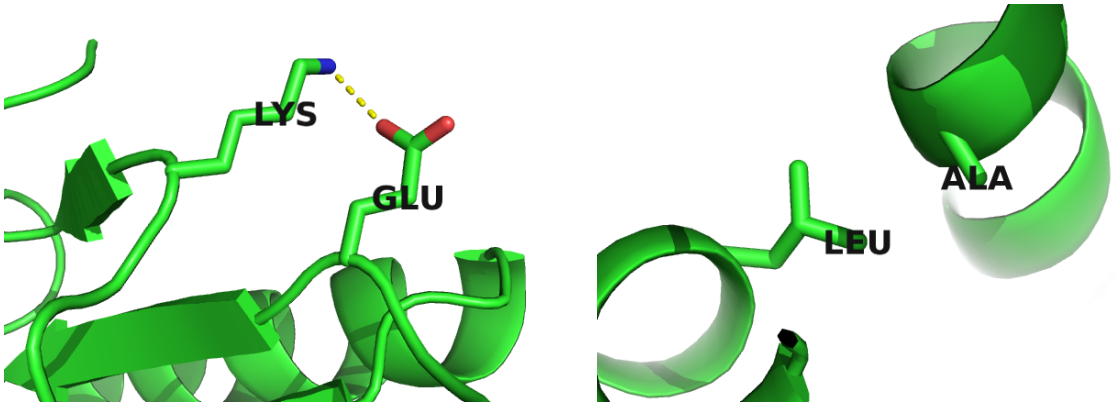

Figure 3.5: Interactions between protein side chains. **Left**: residue 6 (E) forms a salt bridge with residue 82 (R) in protein chain 1a9x_A_05. **Right**: residue 29 (A) and residue 39 (L) within the hydrophobic core of protein chain 1ae9_A_00.

3.3 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 3.6 shows the distribution of selected couplings for filtered residue pairs with $C_\beta - C_\beta$ distances $< 5\mathring{A}$ (see methods section 7.7.2 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 glutamic 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 3.1 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. It is not clear though, why hydrophobic pairs have an equally strong negative coupling signal at this distance range because this speaks against the hypothesis that hydrophobic pairs are commonly interchangeable. A vague explanation could be that a location in the tighly packed protein core calls for other very specific constraints, e.g. sterical fit or contact number, besides hydrophobic properties that are prohibitive for a particular hydrophobic residue at a certain position.

The distribution of aromatic coupling values like F-W is slightly skewed towards negative values, accounting for steric hindrance of their large sidechains at small distances. The yet very pronounced positive coupling signal for the bulky aromatic residues at this short distance range is not clear. The bulky planar aromatic rings of two aromatic residues often point away from each other when their C_β-C_β distances are small to avoid steric hindrance (see Apendix Figure E.5). A positive coupling signal might originate from other structural constraints from the local environment affecting both sidechains, similar to the scenario hypothetically explaining the negative coupling signal for hydrophobic residues.

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 still shifted and skewed, but less pronounced compared to the distributions at $C_\beta - C_\beta$ distances $< 5\mathring{A}$. For aromatic pairs like F-W, the distribution of coupling values has very long tails, suggesting rare but strong couplings for aromatic side chains at this distance.

Figure 3.8 shows the distribution of selected couplings for residue pairs far apart in the protein structure ($C_\beta - C_\beta$ distances $> 20\mathring{A}$).
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 3.1 that C-C coupling values, albeit having large standard-deviations, correlate only

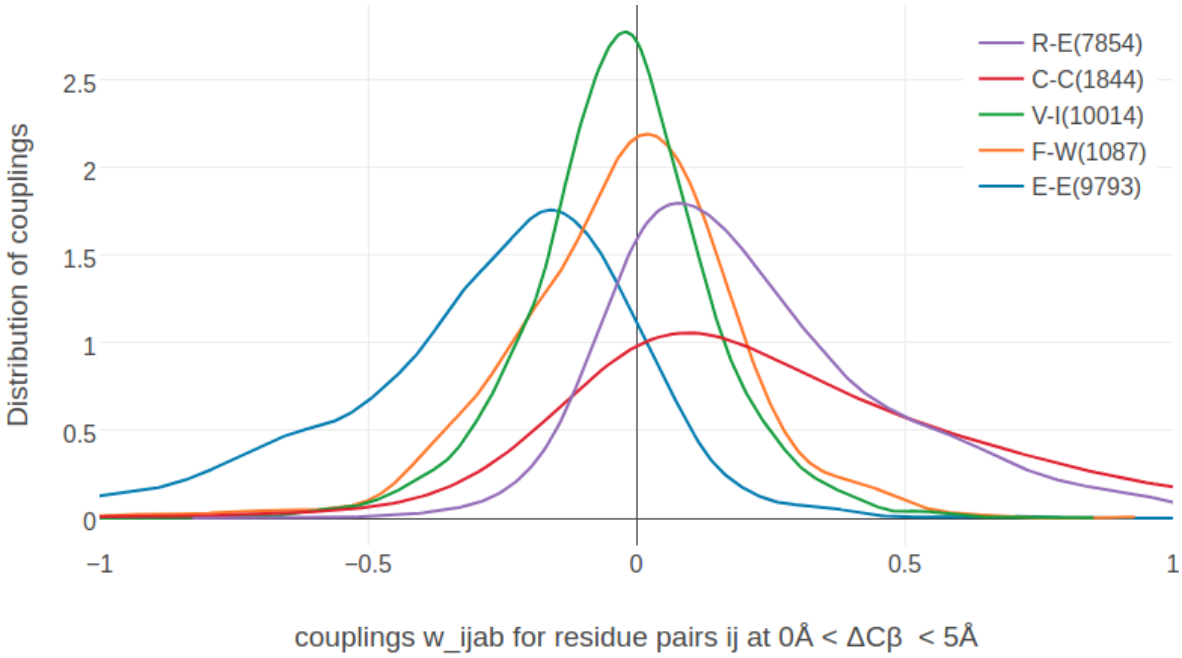

Figure 3.6: Distribution of selected couplings for filtered residue pairs with $C_\beta - C_\beta$ distances $< 5\mathring{A}$ (see methods section 7.7.2 for details). Number of coupling values used to determine the distribution is given in brackets in the legend. R-E = couplings for arginine and glutamic acid pairs, C-C = coupling for cystein residue pairs, V-I = coupling for valine and isoleucine pairs, F-W = coupling for phenylalanine and tryptophane pairs, E-E = coupling for glutamic acid residue pairs.

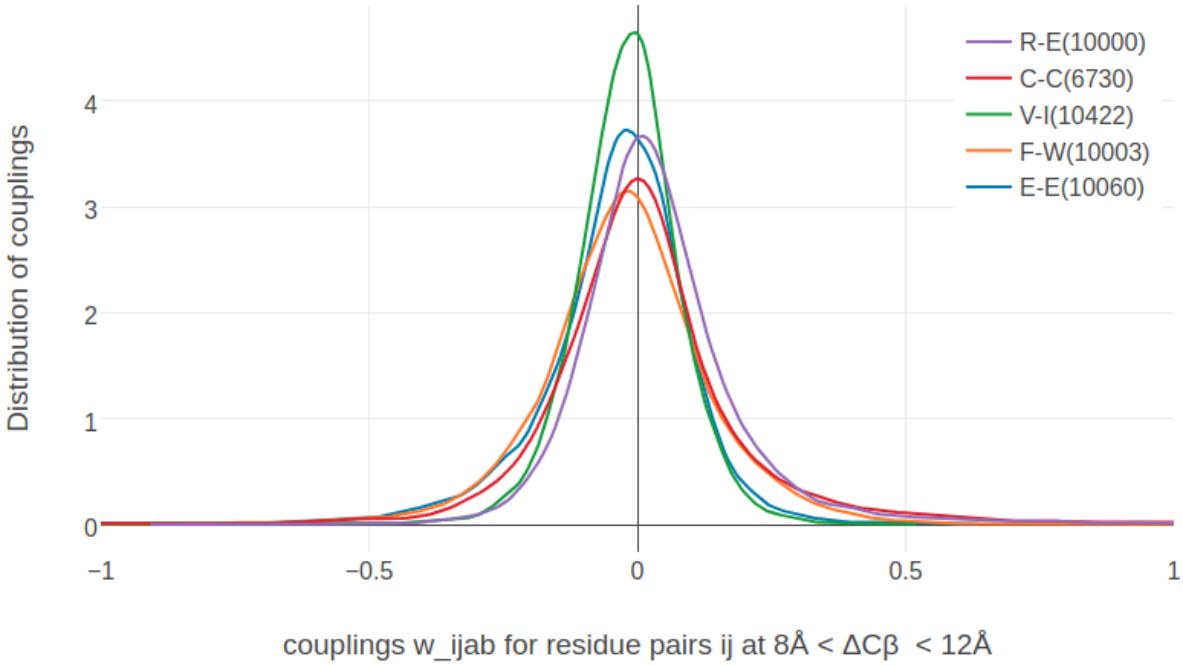

Figure 3.7: Distribution of selected couplings for filtered residue pairs with $C_\beta - C_\beta$ distances between $8\mathring{A}$ and $12\mathring{A}$ (see methods section 7.7.2 for details). Number of coupling values used to determine the distribution is given in brackets in the legend. Couplings are the same as in Figure 3.6.

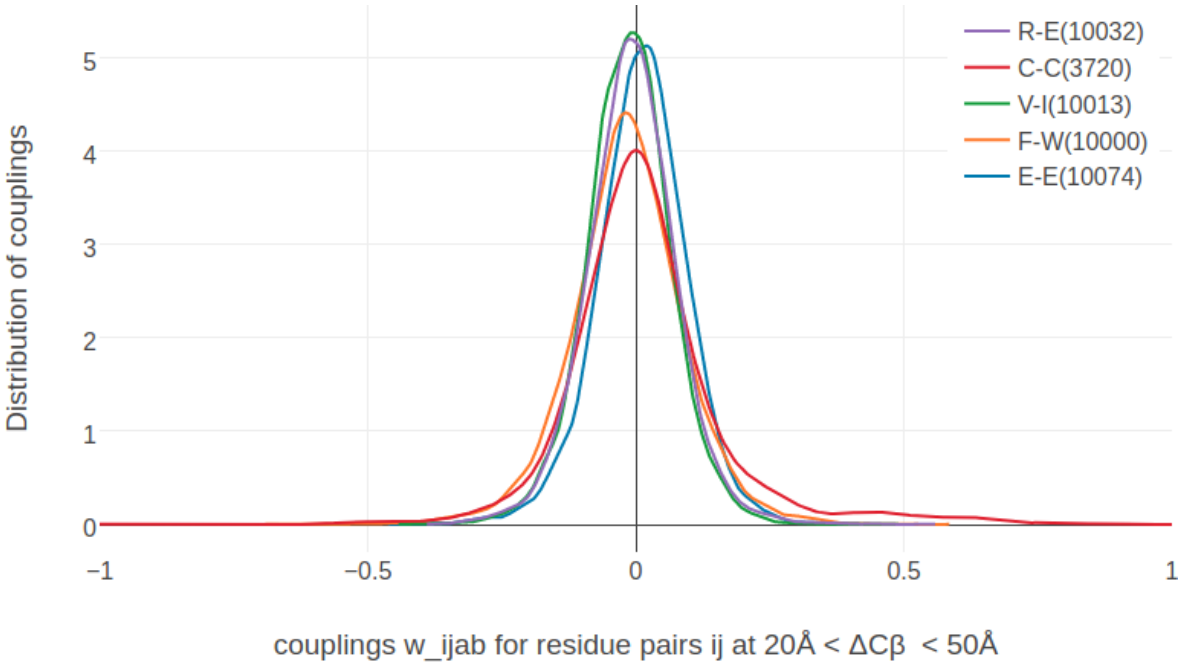

Figure 3.8: Distribution of selected couplings for filtered residue pairs with $C_\beta - C_\beta$ distances between $20\mathring{A}$ and $50\mathring{A}$ (see methods section 7.7.2 for details). Number of coupling values used to determine the distribution is given in brackets in the legend. Couplings are the same as in Figure 3.6.

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 the contact class. The strong coevolution signals for aromatic pairs even at high distance ranges might result from some kind of cooperative effects. Aromatic residues are known to form network-like structures in the protein core that stabilize protein structure [187]. An example is given in Appendix Figure E.4. A possible explanation might be that the *Potts model* is limited to learning single positions and pairwise correlations. An extension to higher order couplings might resolve these cooperative effects observed between residues in the protein core.

3.4 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 dimensions might be misleading when variables are dependent on each other and further insights might be concealed in higher order relationships. Unfortunately, it is not possible to reasonably visualize high dimensional coupling matrices.

Exploring two dimensional coupling scatter plots strengthens the observation that couplings matrices contain signals that reflect biological relevant amino acid interactions. The plots in the top row in Figure 3.9 show the distribution of couplings for filtered residue pairs with $C_\beta - C_\beta$ distances $< 8\mathring{A}$ between the ionic pairings of E-R and R-E and between the ionic pairing R-E and the equally charged residues E-E, respectively. Coupling values for R-E and E-R are positively correlated with predominantly positive values. This means when the amino acid pair R-E is frequently observed at two positions i and j, then it also likely that the amino acid pair E-R can be frequently observed. This situation indicates an important ionic interaction whereby the location of the positively and negatively charged residue at position

i or j is irrelevant.

On the contrary, coupling values for R-E and E-E are negatively correlated, with positive values for R-E and negative values for E-E. This distribution can be interpreted with frequently occuring amino acid pairs R-E at two positions i and j while at the same time the amino acid pair E-E cannot be observed. Again, this situation coincides with amino acid pairings that would be expected for an ionic interaction.

The bottom left plot in Figure 3.9 shows the distribution between couplings for the hydrophobic pairings I-L and V-I that is almost symmetric and broadly centered around zero. Coupling distributions for residue pairs that are not physically interacting ($C_\beta \gg 8\mathring{A}$) resemble the distribution for hydrophobic pairings in that there is no correlation, but at high distance the distributions are much tighter centered around zero (bottom right plot in Figure 3.9).

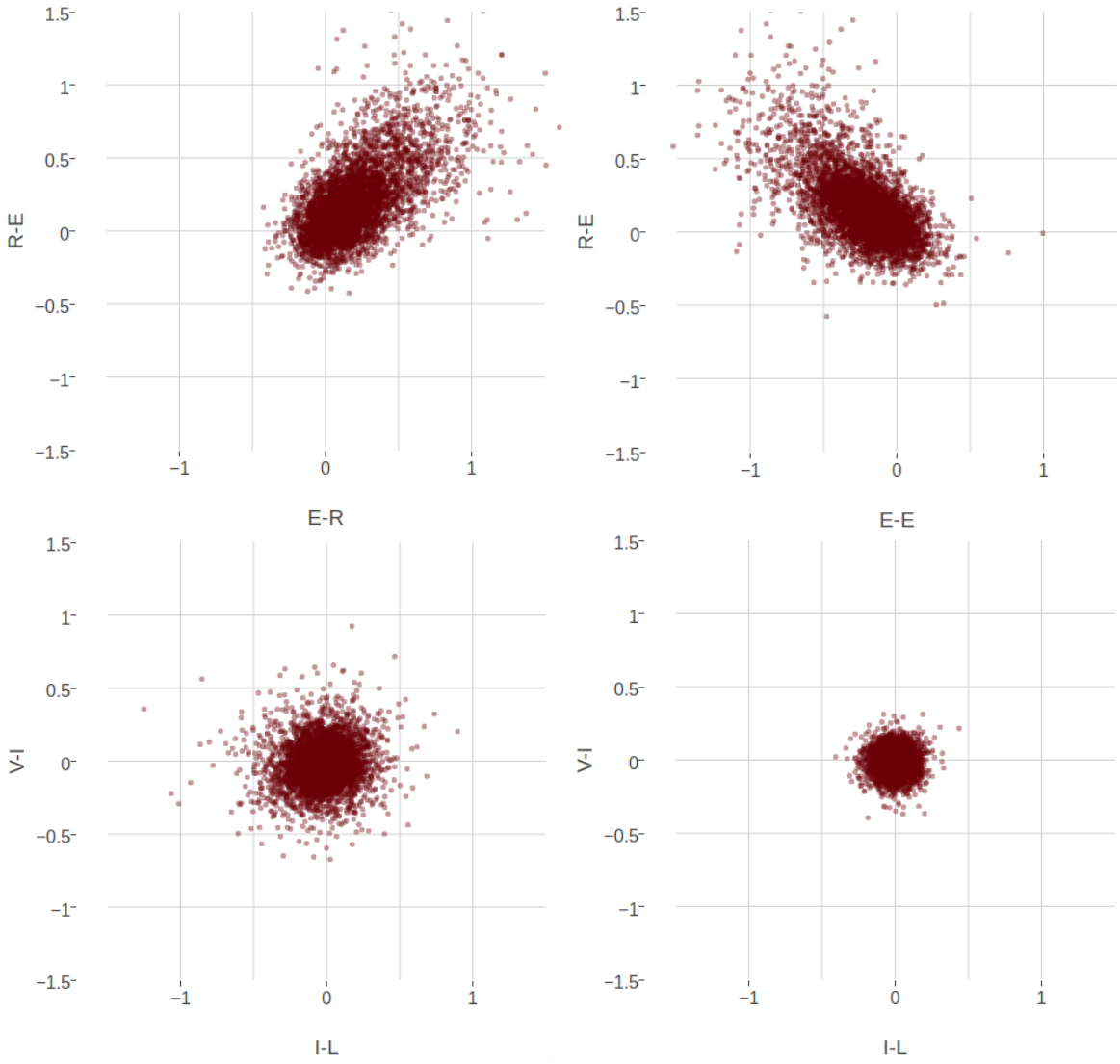

Figure 3.9: Two-dimensional distribution of approximately 10000 coupling values computed with pseudo-likelihood. **Top Left** The 2-dimensional distribution of couplings E-R and R-E for residue pairs with $C_\beta - C_\beta$ distances $< 8\mathring{A}$ is almost symmetric and the coupling values are positively correlated. **Top Right** The 2-dimensional distribution of couplings E-R and E-E for residue pairs with $C_\beta - C_\beta$ distances $< 8\mathring{A}$ is almost symmetric and the coupling values are negatively correlated. **Bottom Left** The 2-dimensional distribution of couplings I-L and V-I for residue pairs with $C_\beta - C_\beta$ distances $< 8\mathring{A}$ is symmetrically distributed around zero without visible correlation. **Bottom Right** The 2-dimensional distribution of couplings I-L and V-I for residue pairs with $C_\beta - C_\beta$ distances $> 20\mathring{A}$ is tighly distributed around zero.

4

Optimizing the Full Likelihood

Section 2.4 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 (see section 2.4.3.1). 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 the most interesting targets for contact prediction (see Figure 2.9).

While the partition function of the full likelihood cannot be efficiently 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 [188]. This section elaborates on how *contrastive divergence* can be used to optimize the full likelihood with gradient descent techniques. Furthermore, two aspects of the underlying *Potts model*, namely gap treatment and the choice of regularization, have been refined which is explained in detail in methods section 7.6.

4.1 Approximating the Gradient of the Full Likelihood with Contrastive Divergence

The gradient of the regularized full log likelihood with respect to the couplings w_{ijab} can be written as

$$\frac{\partial LL_{\text{reg}}}{\partial w_{ijab}} = N_{ij}q(x_i{=}a, x_j = b) - N_{ij}\, p(x_i{=}a, x_j{=}b|\mathbf{v}, \mathbf{w}) - \lambda_w w_{ijab}\,, \tag{4.1}$$

where $N_{ij}q(x_i = a, x_j = b)$ are the empirical pairwise amino acid counts, $p(x_i = a, x_j = b|\mathbf{v}, \mathbf{w})$ corresponds to the marginal distribution of the *Potts model* and $\lambda_w w_{ijab}$ is the partial derivative of the L2-regularizer used to constrain the couplings $\mathbf{w}$. The empirical amino acid counts are constant and need to be computed only once from the alignment. The model

probability term cannot be computed analytically as it involves the partition function that has exponential complexity.

MCMC algorithms are predominantly used in Bayesian statistics to generate samples from probability distributions that involve the computation of complex integrals and therefore cannot be computed analytically [94,189]. Samples are generated from a probability distribution as the current state of a running Markov chain. If the Markov chain is run long enough, the equilibrium statistics of the samples will be identical to the true probability distribution statistics. In 2002, Lapedes et al. applied MCMC sampling to approximate the probability terms in the gradient of the full likelihood [103]. They obtained sequence samples from a Markov chain that was run for 4,000,000 steps by keeping every tenth configuration of the chain. Optimization converged after 10,000 - 15,000 epochs when the gradient had become zero. The expected amino acid counts according to the model distribution, $N_{ij}\ p(x_i = a, x_j = b|\mathbf{v}, \mathbf{w})$, were estimated from the generated samples. Their approach was successfull but is computationally feasible only for small proteins and points out the limits of applying MCMC algorithms. Typically, they require many sampling steps to obtain unbiased estimates from the stationary distribution which comes at high computational costs.

In 2002, Hinton invented CD as an approximation to MCMC methods [188]. It was originally developed for training products of experts models but it can generally be applied to maximizing log likelihoods and has become overly popular for training restricted Boltzmann machines [94,190,191]. The idea is simple: instead of starting a Markov chain from a random point and running it until it has reached the stationary distribution, it is initialized with a data sample and evolved for only a small number of steps. Obviously the chain has not yet converged to its stationary distribution and the data sample obtained from the current configuration of the chain presents a biased estimate. The intuition behind CD is that eventhough the gradient estimate is very noisy and biased, it points roughly into a similar direction as the true gradient of the full likelihood. Therefore the approximate CD gradient should become zero approximately where the true gradient of the likelihood becomes zero. Once the parameters are close to the optimum, starting a Gibbs chain from a data sample should reproduce the empirical distribution and not lead away from it, because the parameters already describe the empirical distribution correctly.

The approximation of the likelihood gradient with CD according to the *Potts* model for modelling protein families is visualized in Figure 4.1. 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. One full step of Gibbs sampling updates every sequence position $i \in \{1, \ldots, L\}$ subsequently by randomly selecting an amino acid based on the conditional probabilities for observing an amino acid a at position i given the model parameters and all other (arady updated) sequence positions:

$$p(\mathbf{x}_i = a|(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_L), \mathbf{v}, \mathbf{w}) \propto \exp\left(v_i(a) + \sum_{j=1;i\neq j}^{L} \mathbf{w}_{ij}(a, x_j)\right) \tag{4.2}$$

The generated sample sequences are then used to compute the pairwise amino acid frequencies that correspond to rough estimates of the marginal probabilities of the *Potts* model. Finally, an approximate gradient of the full likelihood is obtained by subtracting the sampled amino acid counts from the empirical amino acid counts as denoted in eq. (4.1).

The next sections elucidate the optimization of the *Potts* model full likelihood with CD to obtain an approximation to the gradient.

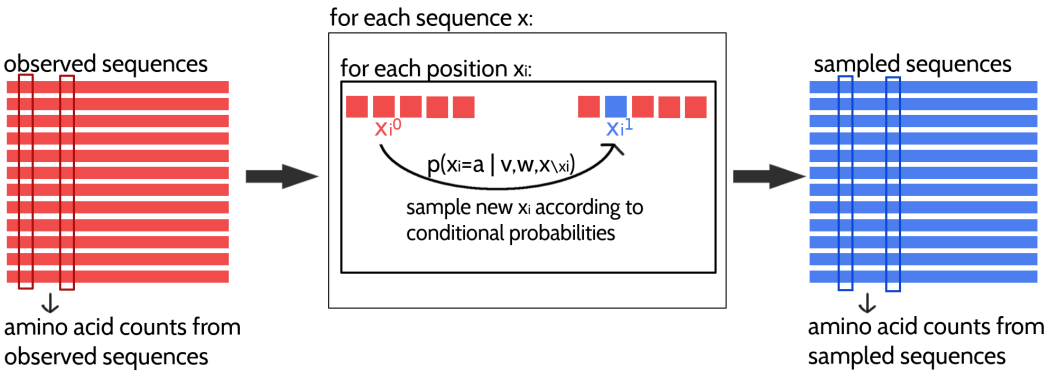

Figure 4.1: Approximating the full likelihood gradient of the *Potts* model with CD. Pairwise amino acid counts are computed from the observed sequences of the input alignment shown in red on the left. Expected amino acid frequencies according to the model distribution are computed from a sampled alignment shown in blue on the right. The CD approximation of the likelihood gradient is obtained by computing the difference in amino acid counts of the observed and sampled alignment. A newly sampled sequence is obtained by evolving a Markov chain, that is initialized with an observed sequence, for one full Gibbs step. The Gibbs step involves updating every position in the sequence (unless it is a gap) according to the conditional probabilities for the 20 amino acids at this position.

4.2 Optimizing the Full Likelihood

Given the likelihood gradient estimates obtained with CD, the full negative log likelihood can now be minimized using a gradient descent optimization algorithm. Gradient descent algorithms are used to find the minimum of an objective function with respect to its parametrization by iteratively updating the parameters values in the opposite direction of the gradient of the objective function with respect to these parameters. SGD is a variant thereof that uses an stochastic estimate of the gradient whose average over many updates approaches the true gradient. The stochasticity is commonly obtained by evaluating a random subsample of the data at each iteration. For CD stochasticity additionally arises from the Gibbs sampling process in order to obtain a gradient estimate in the first place.

As a consequence of stochasticity, the gradient estimates are noisy, resulting in parameter updates with high variance and strong fluctuations of the objective function. These fluctuations 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 [192–194]. However, choosing an optimal step size for parameter updates as well as finding the optimal annealing schedule offers a challenge and needs manual tuning [195,196]. If the step size is chosen too small, progress will be unnecessarily slow, if it is chosen too large, the optimum will be overshot and can cause the system to diverge (see Figure 4.2). Further complications arise from the fact that different parameters often require different optimal step sizes, because the magnitude of gradients might vary considerably for different parameters, e.g. because of sparse data.

Unfortunately, it is neither possible to use second order optimization algorithms nor sophisticated first order algorithms like conjugate gradients to optimize the full likelihood. While the former class of algorithms requires (approximate) computation of the second partial derivatives, the latter requires evaluating the objective function in order to identify the optimal step size via linesearch, both being computationally too demanding.

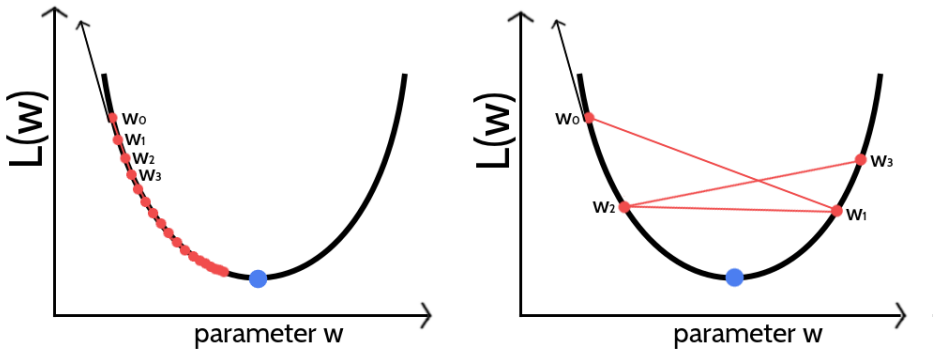

Figure 4.2: Visualization of gradient descent optimization of an objective function $L(w)$ for different step sizes α. The blue dot marks the minimum of the objective function. The direction of the gradient at the initial parameter estimate w_0 is given as black arrow. The updated parameter estimate w_1 is obtained by taking a step of size α into the opposite direction of the gradient. **Left** If the step size is too small the algorithm will require too many iterations to converge. **Right** If the step size is too large, gradient descent will overshoot the minimum and can cause the system to diverge.

The next subsections describe the hyperparameter tuning for stochastic gradient descent, covering the choice of the convergence criterion and finding the optimal learning rate annealing schedule.

4.2.1 Convergence Criterion for Stochastic Gradient Descent

In theory the gradient descent algorithm has converged and the optimum of the objective function has been reached when the gradient becomes zero. In practice the gradients will never be exactly zero, especially due to the stochasticity of the gradient estimates when using stochastic gradient descent with CD. For this reason, it is crucial to define a suitable convergence criterion that can be tested during optimization and once the criterion is met, convergence is assumed and the algorithm is stopped. Typically, the objective function (or a related loss function) is periodically evaluated on a validation set and the optimizer is halted whenever the function value saturates or starts to increase. This technique is called early stopping and additionally prevents overfitting [197,198]. Unfortunately, we cannot compute the full likelihood function due to its complexity and need to define a different convergence criterion.

One possibility is to stop learning when the L2 norm of the gradient for the coupling parameters $||\nabla_{\mathbf{w}} LL(\mathbf{v}^*, \mathbf{w})||_2$ is close to zero [199]. However, when using a finite number of sequences for sampling, the norm of the gradient does not converge to zero but saturates at a certain offset as it is described in section 4.4. Convergence can also be monitored as the relative change of the norm of gradients within a certain number of iterations and optimization can be stopped when the norm of gradient has reached a certain plateau. As gradient estimates are very noisy with stochastic gradient descent, gradient fluctuations complicate the proper assessment of this criterion.

Instead of the gradient, it is also possible to observe the relative change of the norm of parameter estimates $||\mathbf{w}||_2$ over several iterations and stop learning when it falls below a small threshold ϵ,

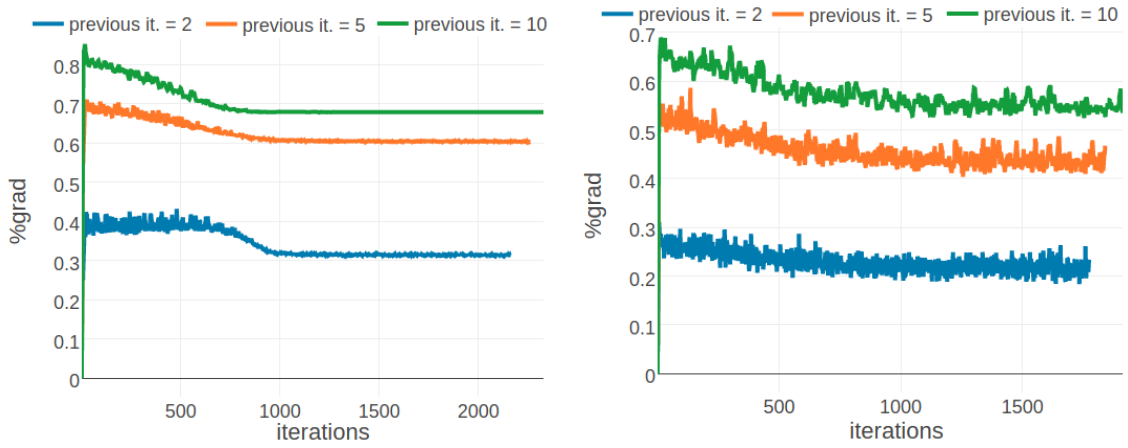

Figure 4.3: Development of the percentage of dimensions for which the derivate has changed its direction over the last iterations. Change in the direction (i.e. the sign) of the partial derivates has been evaluated over varying number of previous iterations as it is specified by *previous it.* in the legend. Optimization is performed with SGD using the optimal hyperparameters described in section 4.2.2 using regularization ocefficient $\lambda_w = 0.1L$ (see section 4.3) and using one step of Gibbs sampling. Optimization is stopped when the relative change over the L2-norm of parameter estimates $||\mathbf{w}||_2$ over the last iterations, as specified in the legend, falls below the threshold of $\epsilon = 1e-8$. Development has been monitored for two different proteins, **Left** 1c75A00 (protein length = 71, Neff = 16808) **Right** 1ahoA00 (protein length = 64, Neff = 229).

$$\frac{||\mathbf{w}_{t-x}||_2 - ||\mathbf{w}_t||_2}{||\mathbf{w}_{t-x}||_2} < \epsilon \,. \tag{4.3}$$

This measure is less noisy than subsequent gradient estimates because the magnitude of parameter updates is bounded by the learning rate.

Another idea is to monitor the direction of the partial derivatives since for stochastic gradient descent the optimum is a moving target and the gradient will start oscillating when approaching the optimum. However, this theoretical assumption is complicated by the fact that gradient oscillations are also typically observed when the parameter surface contains narrow valleys or generally when the learning rate is too big, as it is visualized in the right plot in Figure 4.2. When optimizing high-dimensional problems using the same learning rate for all dimensions, it is likely that parameters converge at different speeds [192] leading to oscillations that could either originate from convergence or yet too large learning rates. As can be seen in Figure 4.3, the percentage of dimensions for which the derivate changes direction within the last iterations is usually high and varies for different proteins. Therefore it is not a good indicator of convergence. When using the adaptive learning rate optimizer *ADAM*, the momentum term is an interfering factor for assessing the direction of partial derivatives. Parameters will be updated into the direction of a smoothed historical gradient and oscillations, regardless of which origin, will be dampened. It is therefore hard to define a general convergence criteria based on the direction of derivatives that can distinguish these different scenarios.

Of course, the simplest strategy to assume convergence is to specify a maximum number of iterations for the optimization procedure, which also ensures that the algorithm will stop eventually if none of the other convergence criteria is met.

A necessary but not sufficient criterion for convergence for the full likelihood is given by

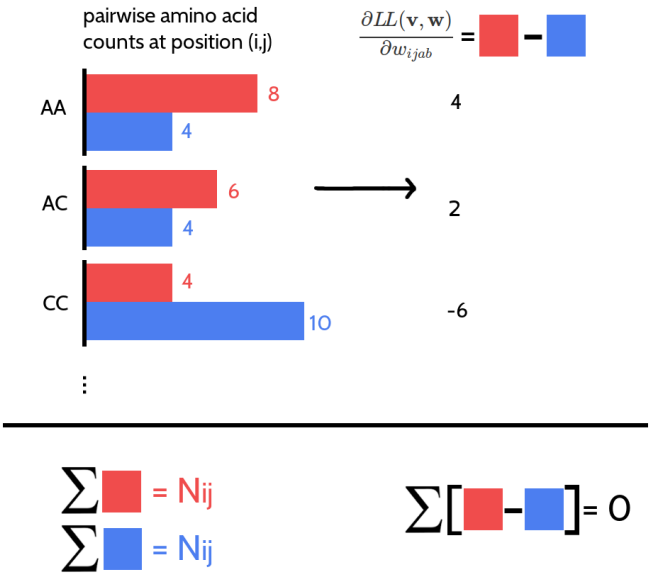

Figure 4.4: The 400 partial derivatives $\frac{\partial LL_{\text{reg}}(\mathbf{v},\mathbf{w})}{\partial w_{ijab}}$ at position (i,j) for $a,b \in \{1,\ldots,20\}$ are not independent. Red bars represent pairwise amino acid counts at position (i,j) for the sampled alignment. Blue bars represent pairwise amino acid counts at position (i,j) for the input alignment. The sum over pairwise amino acid counts at position (i,j) for both alignments is N_{ij}, which is the number of ungapped sequences. The partial derivative for w_{ijab} is computed as the difference of pairwise amino acid counts for amino acids a and b at position (i,j). The sum over the partial derivatives $\frac{\partial LL_{\text{reg}}(\mathbf{v},\mathbf{w})}{\partial w_{ijab}}$ at position (i,j) for all $a,b \in \{1,\ldots,20\}$ is zero.

$\sum_{a,b=1}^{20} w_{ijab} = 0$. This requirement is derived in section 7.6.3. When using plain stochastic gradient descent without momentum and without adaptive learning rates, this criterion is never violated when parameters are initialized uniformly. This is due to the fact that the 400 partial derivatives for w_{ijab} given a pair (i,j) and for $a,b \in \{1,\ldots,20\}$ are not independent because the sum over the 400 pairwise amino acid counts at positions i and j is identical for the observed and the sampled alignment and amounts to,

$$\sum_{a,b=1}^{20} N_{ij} q(x_i = a, q_j = b) = N_{ij} \ . \tag{4.4}$$

Considering a residue pair (i,j) and assuming amino acid pair (a,b) has higher counts in the sampled alignment than in the observed input alignment, then this difference in counts must be compensated by other amino acid pairs (c,d) having less counts in the sampled alignment compared to the true alignment (see Figure 4.4). Therefore it holds, $\sum_{a,b=1}^{20} \nabla_{w_{ijab}} LL_{\text{reg}}(\mathbf{v},\mathbf{w}) = 0$. This symmetry is translated into parameter updates as long as the same learning rate is used to update all parameters. However, when using adaptive learning rates, this symmetry is broken and the condition $\sum_{a,b=1}^{20} w_{ijab} = 0$ can be violated during the optimization processs. It is therefore interesting to monitor $\sum_{1\leq 1<j\leq L} \sum_{a,b=1}^{20} w_{ijab}$.

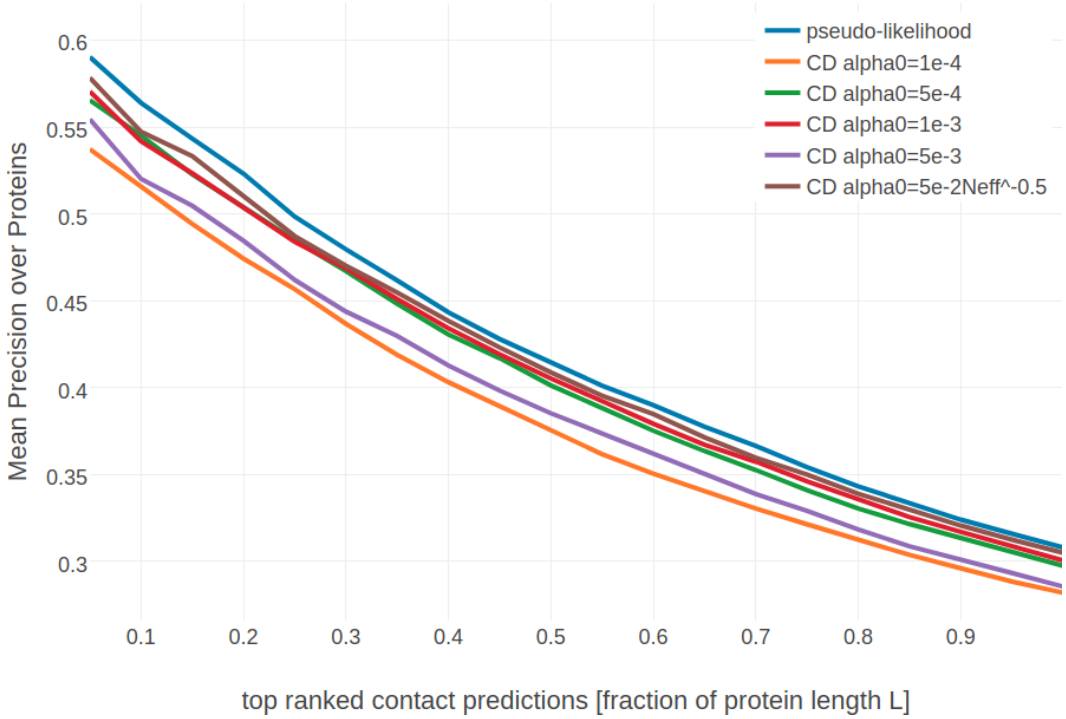

Figure 4.5: Mean precision for top ranked contact predictions over 286 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. **pseudo-likelihood**: couplings computed with pseudo-likelihood. **CD**: couplings computed with CD using stochastic gradient descent with different initial learning rates α_0 as specified in the legend.

4.2.2 Tuning Hyperparameters of Stochastic Gradient Descent Optimizer

The coupling parameters $\mathbf{w}$ will be updated at each time step t by taking a step of size α along the direction of the negative gradient of the regularized full log likelihood, $-\nabla_w LL_{\text{reg}}(\mathbf{v}, \mathbf{w})$, that has been approximated with CD,

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \cdot \nabla_w LL_{\text{reg}}(\mathbf{v}, \mathbf{w}) \; . \tag{4.5}$$

In order to get a first intuition of the optimization problem, I tested initial learning rates $\alpha_0 \in \{1\text{e}{-4}, 5\text{e}{-4}, 1\text{e}{-3}, 5\text{e}{-3}\}$ with a standard learning rate annealing schedule, $\alpha = \frac{\alpha_0}{1+\gamma \cdot t}$ where t is the time step and γ is the decay rate that is set to 0.01[193].

Figure 4.5 shows the mean precision for top ranked contacts computed from pseudo-likelihood couplings and from CD couplings optimized with stochastic gradient descent using the four different learning rates. Overall, mean precision for CD contacts is lower than for pseudo-likelihood contacts, especially when using the smallest ($\alpha_0 = 1\text{e}{-4}$) and biggest ($\alpha_0 = 5\text{e}{-3}$) learning rate.

Looking at individual proteins it turns out that the optimal learning rate depends on alignment size. The left plot in Figure 4.6 shows a convergence plot of SGD optimization using different learning rates for a protein with a small alignment. With a small initial learning rate $\alpha_0 = 1\text{e} - 4$ the optimization runs very slowly and does not reach convergence within 5000 iterations. Using a large initial learning rate $\alpha_0 = 5\text{e} - 3$ will result in slighly overshooting the optimum at the beginning of the optimization but with the learning rate decaying over time the parameter estimates converge. In contrast, for a protein with a big alignment (right

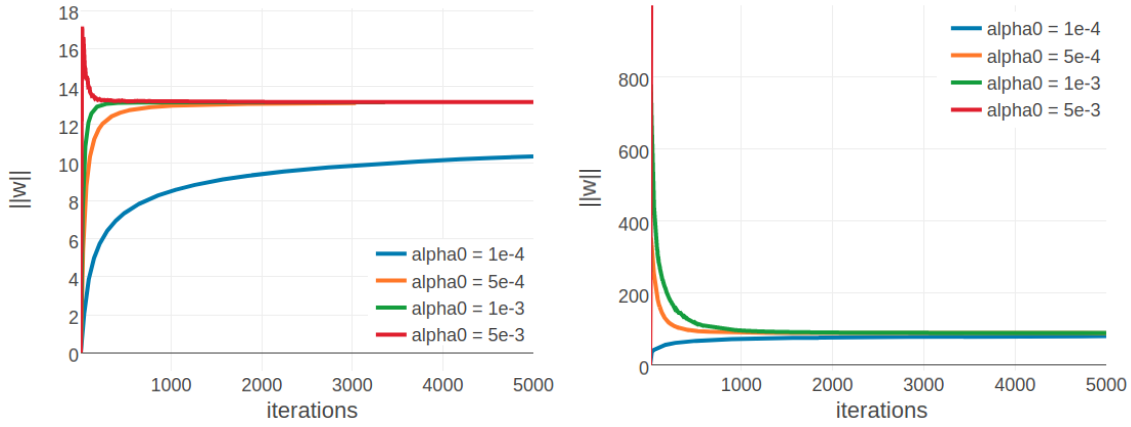

Figure 4.6: Convergence plots for two proteins during SGD optimization with different learning rates and convergence measured as L2-norm of the coupling parameters $||\mathbf{w}||_2$. Linear learning rate annealing schedule has been used with decay rate $\gamma = 0.01$ and initial learning rates α_0 have been set as specified in the legend. **Left** Convergence plot for protein 1aho_A_00 having protein length L=64 and 378 sequences in the alignment (Neff=229). **Right** Convergence plot for protein 1c75_A_00 having protein length L=71 and 28078 sequences in the alignment (Neff=16808). Figure is cut at the yaxis at $||\mathbf{w}||_2 = 1000$, but learning rate of 5e−3 reaches $||\mathbf{w}||_2 \approx 9000$.

plot in Figure 4.6) the choice of learning rate has a more pronounced effect. With a small initial learning rate $\alpha_0 = 1\text{e}-4$ the optimization runs slowly but almost converges within 5000 iterations. A large initial learning rate $\alpha_0 = 5\text{e}-3$ lets the parameters diverge quickly and the optimum cannot be revovered. With learning rates $\alpha_0 = 5\text{e}-4$ and $\alpha_0 = 1\text{e}-3$, the optimum is well overshot at the beginning of the optimization but the parameter estimates eventually converge as the learning rate decreases over time.

These observations can be explained by the fact that the magnitude of the gradient scales with the number of sequences in the alignment. The gradient is computed from amino acid counts as explained in section 4.1. Therefore, alignments with many sequences will generally produce larger gradients than 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 largest. Following these observations, I defined the initial learning rate α_0 as a function of Neff, aiming at values for α_0 around 5e-3 for small Neff and values for α_0 around 1e-4 for large Neff,

$$\alpha_0 = \frac{5\text{e}-2}{\sqrt{N_{\text{eff}}}} \,. \tag{4.6}$$

For small Neff $\approx$ 50 this definition of the learning rate yields $\alpha_0 \approx 7\text{e}-3$ and for big Neff $\approx$ 20000 this yields $\alpha_0 \approx 3.5\text{e}-4$. Using this learning rate defined as a function of Neff, precision improves over the previous fixed learning rates (see Figure 4.5). All following analyses are conducted using the Neff-dependent learning rate.

In a next step, I evaluated the following learning rate annealing schedules and decay rates using the Neff-dependent initial learning rate given in eq. (4.6):

- default linear learning rate schedule $\alpha = \frac{\alpha_0}{1+\gamma t}$ with $\gamma \in \{1\text{e}-3, 1\text{e}-2, 1\text{e}-1, 1\}$
- square root learning rate schedule $\alpha = \frac{\alpha_0}{\sqrt{1+\gamma t}}$ with $\gamma \in \{1\text{e}-2, 1\text{e}-1, 1\}$
- sigmoidal learning rate schedule $\alpha_{t+1} = \frac{\alpha_t}{1+\gamma t}$ with $\gamma \in \{1\text{e}-6, 1\text{e}-5, 1\text{e}-4, 1\text{e}-3\}$

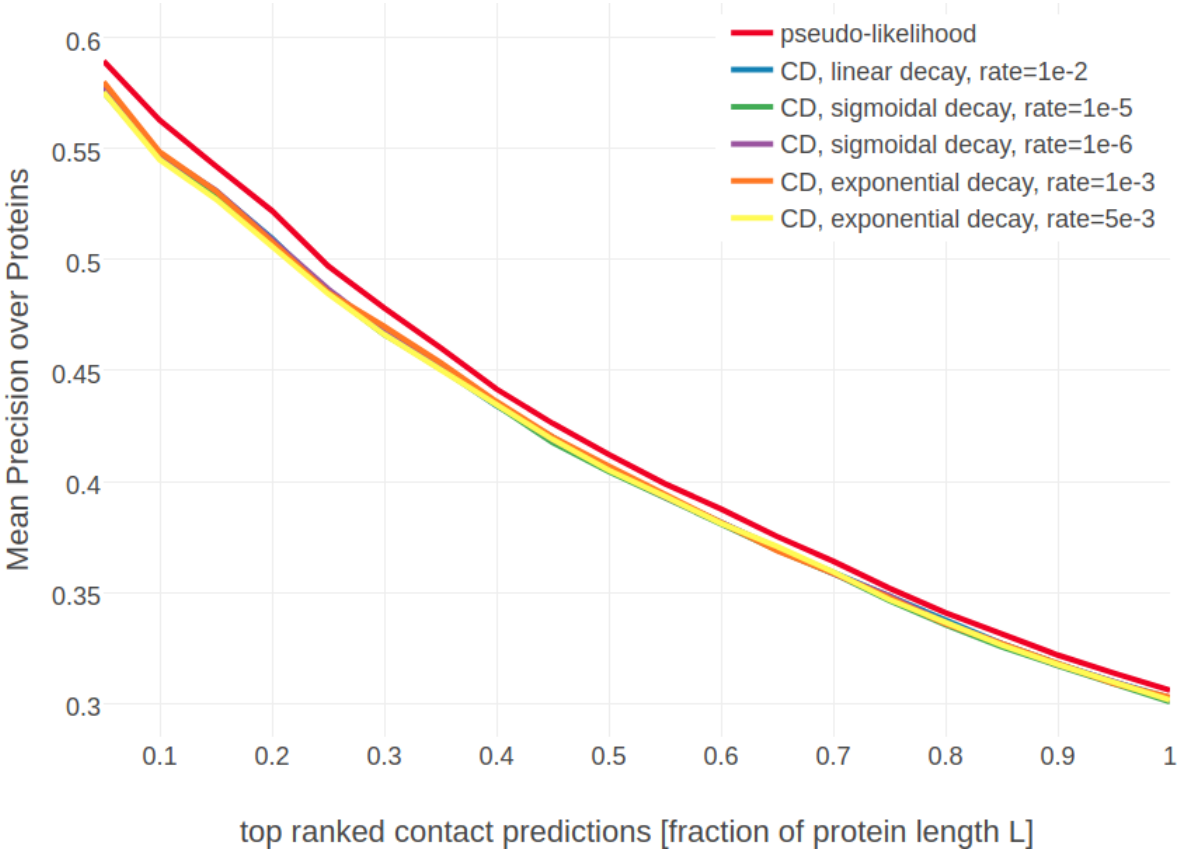

Figure 4.7: Mean precision for top ranked contact predictions over 288 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. **pseudo-likelihood**: couplings computed with pseudo-likelihood. **CD**: couplings computed with CD using stochastic gradient descent with an initial learning rate defined with respect to Neff. Learning rate annealing schedules and decay rates as specified in the legend.

- exponential learning rate schedule $\alpha_{t+1} = \alpha_0 \cdot \exp(-\gamma t)$ with $\gamma \in \{5\text{e}{-4}, 1\text{e}{-4}, 5\text{e}{-3}\}$

The learning rate annealing schedules are visualized for different decay rates in Appendix Figure F.1 and the respective benchmark plots can be found in Appendix F.2. Optimizing CD with SGD using any of the learning rate schedules listed above yields on average lower precision for the top ranked contacts than the pseudo-likelihood contact score. Several learning rate schedules perform almost equally well as can be seen in Figure 4.7. The highest precision, being one to two percentage points below the mean precision for the pseudo-likelihood contact score, is obtained with a linear learning rate schedule and decay rate $\gamma = 1\text{e}{-2}$, with a sigmoidal learning rate schedule and decay rates $\gamma = 1\text{e}{-5}$ and $\gamma = 1\text{e}{-6}$ and with an exponential learning rate schedule and decay rate $\gamma = 1\text{e}{-3}$ and $\gamma = 1\text{e}{-5}$. The square root learning rate schedule gives ovarally bad results and does not lead to convergence because the learning rate decays slowly at later time steps.

In contrast to the findings regarding the initial learning rate earlier, an optimal decay rate can be defined independent of the alignment size. Figure 4.8 shows convergence plots for the same two exemplary proteins as before. Proteins with low Neff are robust against the particular choice of learning rate schedule and decay rate (see left plot in Figure 4.8). The presumed optimum at $||w||_2 \approx 13.2$ is almost always reached. Proteins with high Neff are stronger adversely affected by quickly decaying learning rates. For example, the learning rate decays quickly and then converges when using a linear learning rate, which effectively prevents

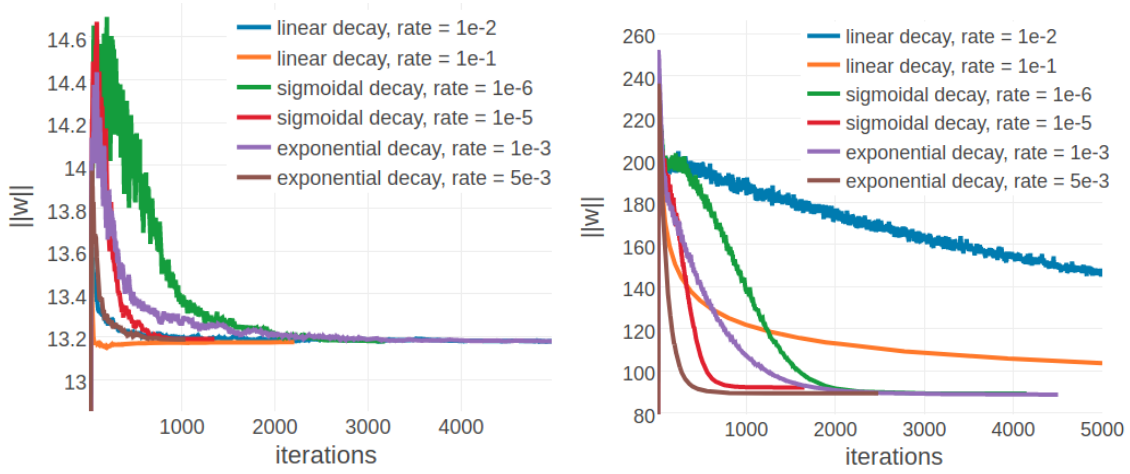

Figure 4.8: L2-norm of the coupling parameters $||\mathbf{w}||_2$ during stochastic gradient descent optimization with different learning rates schedules. The initial learning rate α_0 is defined with respect to Neff as given in eq. (4.6). Learning rate schedules and decay rates are used according to the legend. **Left** Convergence plot for protein 1aho_A_00 having protein length L=64 and 378 sequences in the alignment (Neff=229). **Right** Convergence plot for protein 1c75_A_00 having protein length L=71 and 28078 sequences in the alignment (Neff=16808).

further optimization progress and the presumed optimum at $||w||_2 \approx 90$ is not reached within 5000 iterations. Less quickly decaying learning rates, such as $\gamma = 1e{-}3$ with an exponential schedule or $\gamma = 1e{-}6$ with a sigmoidal schedule, guide the parameter estimates close to the expected optimum and can be used with proteins having low Neffs as well as having high Neffs.

Several different learning rate annealing schedules yield almost identical mean precision for top ranked contacts, as was shown earlier (see Figure 4.7). But it can be found that they differ in convergence speed. Figure 4.9 shows the distribution over the number of iterations until convergence for SGD optimizations with five different learning rate schedules that yield similar performance. The optimization converges on average within less than 2000 iterations only when using either a sigmoidal learning rate annealing schedule with decay rate $\gamma = 1e{-}5$ or an exponential learning rate annealing schedule with decay rate $\gamma = 5e{-}3$, On the contrary, the distribution of iterations until convergence has a median of 5000 when using a linear learning rate annealing schedule with $\gamma = 1e{-}2$ or an exponential schedule with decay rate $\gamma = 1e{-}3$. Under these considerations, I chose a sigmoidal learning rate schedule with $\gamma = 5e{-}6$ for all further analysis.

Finally, I checked whether altering the convergence criteria has notable impact on performance. Per default, optimization is stopped whenever the relative change of the L2 norm over coupling parameters $||\mathbf{w}||_2$ falls below a small value $\epsilon < 1e-8$ within the last $t = 5$ iterations as given in eq. (4.3). As can be seen in Figure 4.10 the mean precision over proteins is robust to different settings of the number of iterations t over which the relative change of the norm of couplings is measured. The mean number of iterations until convergence increases slighty when increasing t, from 1697 iterations for $t = 2$, to 1782 iterations for $t = 5$, to 1917 iterations for $t = 10$ as can be sees in Appendix Figure F.10.

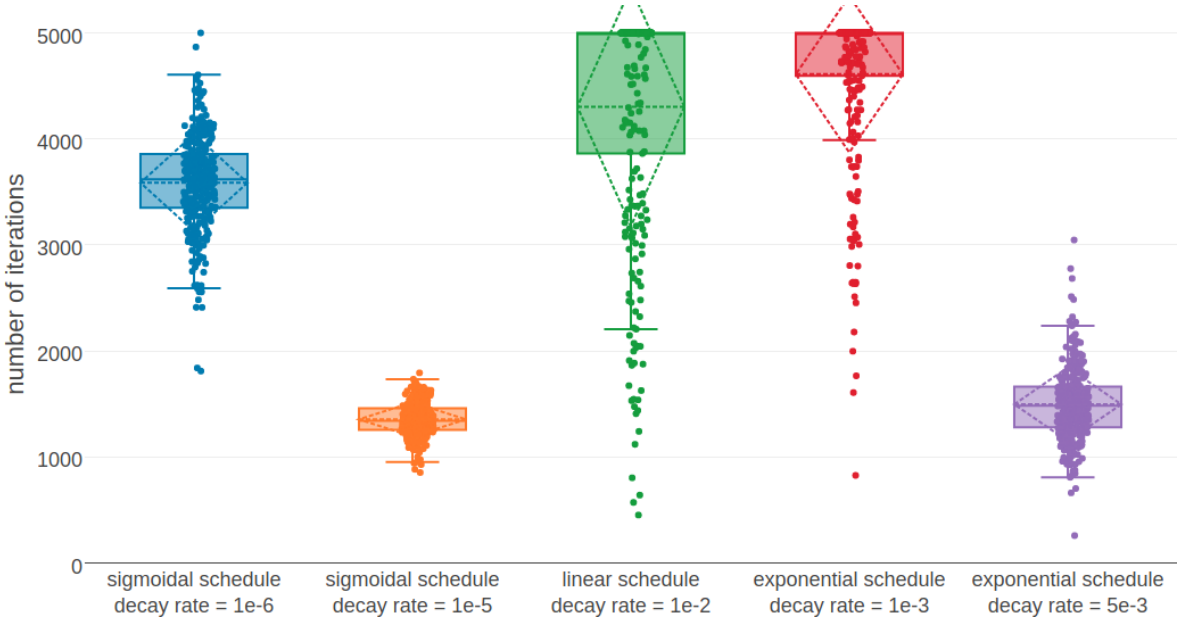

Figure 4.9: Distribution of the number of iterations until convergence for SGD optimizations of the full likelihood for different learning rate schedules. Convergence is reached when the relative difference of parameter norms $||\mathbf{w}||_2$ falls below $\epsilon = 1e-8$. Initial learning rate α_0 is defined with respect to Neff as given in eq. (4.6) and maximum number of iterations is set to 5000. Learning rate schedules and decay rates are used as specified in the legend.

4.3 Tuning Regularization Coefficients for Contrastive Divergence

For tuning the hyperparameters of the stochastic gradient descent optimizer in the last section 4.2.2, the coupling parameters $\mathbf{w}$ were constrained by a Gaussian prior $\mathcal{N}(\mathbf{w}|0, \lambda_w^{-1} I)$ using the default pseudo-likelihood regularization coefficient $\lambda_w = 1\text{e}{-2}L$ as decscribed in methods section 7.5. It is conceivable that CD achieves optimal performance using stronger or weaker regularization than used for pseudo-likelihood optimization. Therefore, I evaluated performance of CD using the previously identified hyperparamters for SGD and different regularization coefficients $\lambda_w \in \{1\text{e}{-2}L, 5\text{e}{-2}L, 1\text{e}{-1}L, 1\text{e}{-2}L, L\}$. The single potentials $\mathbf{v}$ are not subject to optimization and are kept fixed at their maximum-likelihood estimate v^* given in eq. (7.28).

As can be seen in Figure 4.11, using strong regularization for the couplings, $\lambda_w = L$, results in a drastic drop of mean precision. Using weaker regularization such as $\lambda_w = 1\text{e}{-2}L$ or $\lambda_w = 5\text{e}{-2}L$ improves precision for the top $L/10$ and $L/5$ predicted contacts but decreases precision when including lower ranked predictions. As a matter of fact, a slightly weaker regularization $\lambda_w = 1\text{e}{-1}L$ than the default $\lambda_w = 1\text{e}{-2}L$ improves mean precision especially for the top $L/2$ contacts in such a way, that it is comparable to the pseudo-likelihood performance.

As mentioned before, a difference compared to pseudo-likelihood optimization is that the single potentials $\mathbf{v}$ are not optimized with CD but rather set to their maximum-likelihood estimate as it is obtained in a single position model that is discussed in methods section (7.28). When the single potentials $\mathbf{v}$ are optimized with CD using the same regularization coefficient $\lambda_v = 10$ as it is used when optimizing the pseudo-likelihood, performance is almost indistinguishable compared to keeping the single potentials $\mathbf{v}$ fixed as can be seen in appendix Figure F.12.

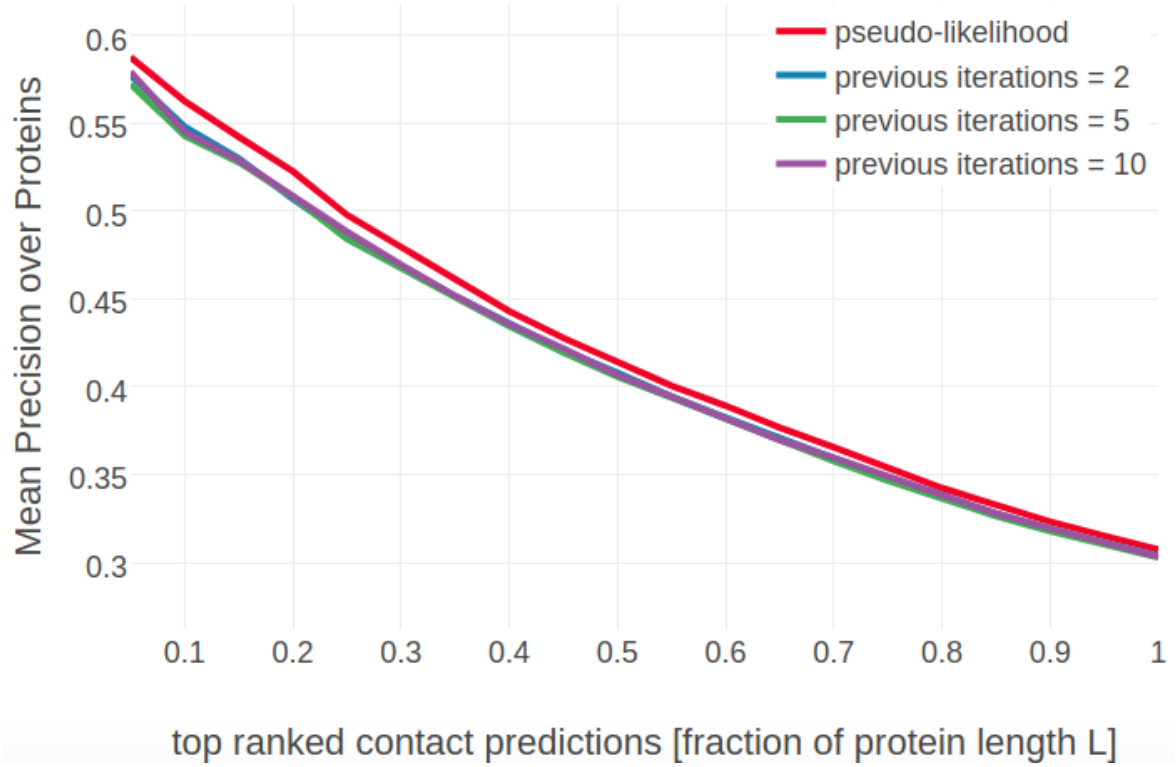

Figure 4.10: Mean precision for top ranked contact predictions over 288 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. **pseudo-likelihood**: couplings computed with pseudo-likelihood. **#previous iterations = X**: couplings computed with CD using stochastic gradient descent with an initial learning rate defined with respect to Neff and the sigmoidal learning rate schedule with $\gamma = 5e{-}6$. Different values for the number of previous iterations (as specified in the legend) have been evaluated over which the relative change of the norm of coupling parameter is evaluated, which defines the exact convergence criterion.

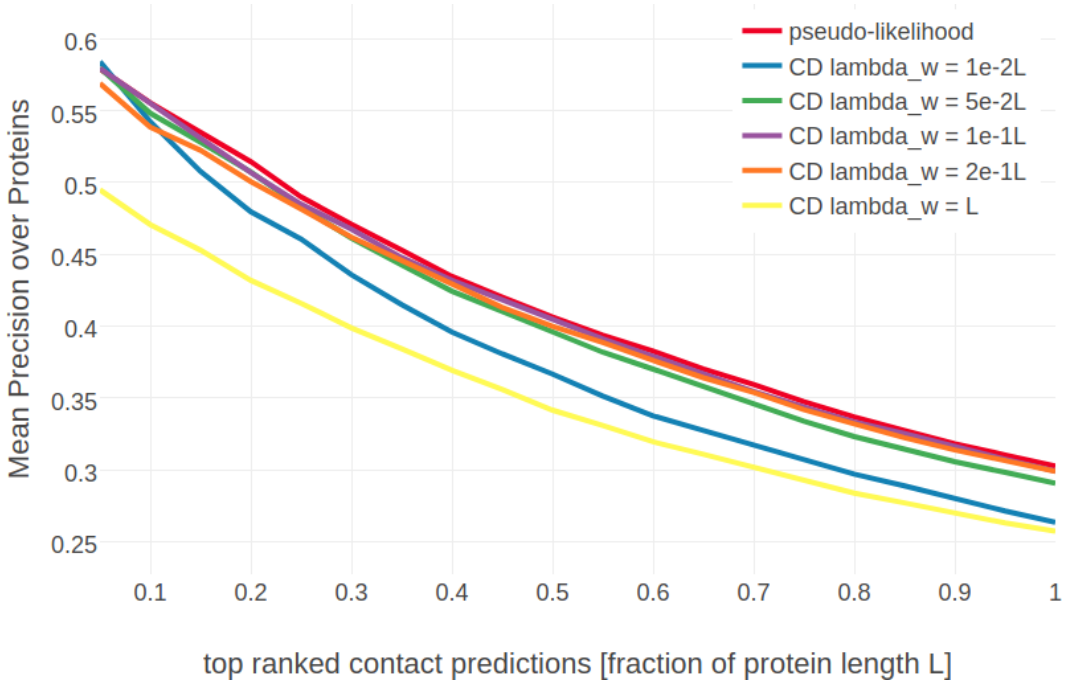

Figure 4.11: Mean precision for top ranked contact predictions over 288 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. Subsets are defined according to quantiles of Neff values. Upper left: Subset of proteins with Neff < Q1. Upper right: 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. **pseudo-likelihood**: couplings computed with pseudo-likelihood. **CD lambda_w = X**: couplings computed with CD using L2-regularization on the couplings $\mathbf{w}$ with regularization coefficient λ_w chosen as specified in the legend and keeping the single potentials v_i fixed at their MLE optimum v_i^* given in eq. (7.28).

4.4 Tuning the Gibbs Sampling Scheme for Contrastive Divergence

The original CD-k algorithm described by Hinton in 2002 evolves the Markov chains by k=1 Gibbs steps [188]. As described earlier, CD-1 provides a biased estimate of the true gradient because the Markov chains have not reached the stationary distribution [190]. Bengio and Delalleau show that the bias for CD-k kan be understood as a residual term when expressing the log likelihood gradient as an expansion that involves the k-th sample of the Gibbs chain [191]. As k goes to infinity the residual term and hence the bias converges to zero and the CD gradient estimate converges to a stochastic estimation of the true likelihood gradient. Indeed, eventhough surprising results have been obtained by evolving the Markov chains for only one Gibbs step, typically CD-k for k>>1 gives more precise results [191]. Furthermore it has been shown, that bias also depends on the mixing rate or the rate of convergence of the chains whereby the mixing rate decreases when model parameters increase [200]. This can lead to divergence of the CD-k solution from the maximum-likelihood solution in a sense the model systematically gets worse as optimization progresses [201]. Regularization of the parameters offers a solution to this problem, constraining the magnitude of the parameters. A different solution suggested by Bengio and Delalleau is to dynamically increase k when the model parameters increase [191]. These studies analysing the convergence properties and the expected approximation error for CD-k have mainly been conducted for Restricted Boltzmann Machines. It is therefore not clear, whether and to what extent these findings apply to the *Potts* model.

Several connections of CD to other well known approximation algorithms have been drawn. For example, it can be shown that CD by sampling only one random variable according to the conditional probability is exactly equivalent to optimising the pseudo-likelihood [202,203]. Asuncion and colleagues showed further that an arbitrary good approximation to the full likelihood can be reached by applying blocked-Gibbs sampling [204]. Thereby, CD by varying the number variables that is randomly samples, has an equivalent composite likelihoods, which is a higher-order generalization of the pseudo-likelihood.

PCD is a variation of CD such that the Markov chain is not reinitialized at a data sample every time a new gradient is computed [200]. Instead, the Markov chains are kept *persistent* that is, they are evolved between successive gradient computations. The fundamental idea behind PCD is that the model changes only slowly between parameter updates given a sufficiently small learning rate and the Markov chains will not be pushed too far from equilibrium after each update but rather stay close to the stationary distribution [94,190,200]. Tieleman and others observed that PCD often works better than CD, eventhough CD can be faster in the early stages of learning and thus should be preferred when runtime is the limiting factor [94,200].

persistent CD performs better than CD in all practical cases testet and it performs similarly to CD10, a version of contrastive divergence using 10 Gibbs sampling steps instead of just one [johannes]

The next sections discuss various modifications of the CD algorithm, such as varying the regularizatio strength λ_w for constraining the coupling parameters $\mathbf{w}$, increasing the number of Gibbs sampling steps and varying the number of Markov chains used for sampling. Persistent contrastive divergence is analysed for various combinations of the above mentioned settings and eventully combined with CD-k. Unless noted otherwise, all optimizations will be performed using stochastic gradient descent with the tuned hyperparameters described in the last sections.

4.4.1 Varying the Sample Size

The default Gibbs sampling scheme explained in method section 7.9 involves the random selection of $10L$ sequences from the input alignment, with L being protein length, at every iteration of the optimization procedure. These sequences are used to initialize the Markov chains for Gibbs sampling new sequences to estimate the gradient with CD. The particular choice of $10L$ sequences was motivated by the fact that there is a relationship between the precision of contacts predicted from pseudo-likelihood and protein length as long as the alignment has less than 10^3 diverse sequences [174]. It has been argued that roughly $5L$ nonredundant sequences are required to obtain confident predictions that can bet used for protein structure prediction [102].

I analysed whether varying the number of sequences used for the approximation of the gradient via Gibbs sampling affects performance. Randomly selecting only a subset of sequences S from the N sequences of the input alignment corresponds to the stochastic gradient descent idea of a minibatch and introduces additional stochasticity over the CD Gibbs sampling process. Using $S < N$ sequences for Gibbs sampling has the further advantage of decreasing the runtime at each iteration. I evaluated different schemes for randomly selecting sequences for Gibbs sampling at every iteration of the optimization:

- sampling $x \cdot L$ sequences with $x \in \{1, 5, 10, 50\}$ without replacement enforcing $S = \min(N, xL)$
- sampling $x \cdot N_{\text{eff}}$ sequences with $x \in \{0.2, 0.3, 0.4\}$ without replacement

As can be seen in Figure 4.12, randomly selecting L sequences for sampling, results in a visible drop in performance. Using $5L$ sequences for sampling results in slighlty decreased performance over using $10L$ or $50L$ sequences. There is no benefit in using more than $10L$ sequences, especially as sampling more sequences increases runtime per iteration. Specifying the number of sequences for sampling as fractions of Neff generally improves precision slightly over selecting $10L$ or $50L$ sequences for sampling. And by sampling $0.3N_{\text{eff}}$ and $0.4N_{\text{eff}}$ sequences, CD does even slighty improve over pseudo-likelihood.

When evaluating performance with respect to the number of effective sequences Neff, it can clearly be noted that the optimal samplings size must depend on Neff. Selecting too many sequences, e.g. $50L$ for small alignments (upper left plot in Figure 4.13), or selecting too few sequences, e.g $1L$ for big alignments (lower right plot in Figure 4.13), results in a decrease in precision compared to defining sampling size as fractions of Neff. Especially small alignments benefit from sample sizes defined as a fraction of Neff with improvements of about three percentage points in precision over pseudo-likelihood.

To understand the effect of different choices of sample size it is necessary to look at single proteins. The left plot in Figure 4.14 shows the development of the L2 norm of the gradient for couplings, $||\nabla_{\mathbf{w}} LL(\mathbf{v}^*, \mathbf{w})||_2$, for protein chain 1c75_A_00 that is of length 71 and has Neff = 16808. The norm of the gradient decreases during optimization and saturates at decreasing levels for increasing choices of sample size. Increasing the sample size by a factor 100 (from L to $100L$) leads to an approximately 10-fold reduction of the norm of gradient (1.4e+5 compared to 1.45e+4) at convergence, which corresponds to a typical reduction of statistical noise as the square root of the number of samples. It is not feasible to sample the number of sequences at each iteration that would be necessary to reduce the norm of the gradient to near zero.

In any case, precision of the top ranked contacts does not improve to the same amount as the norm of the gradient decreases when using larger sample sizes as could be seen in the previous

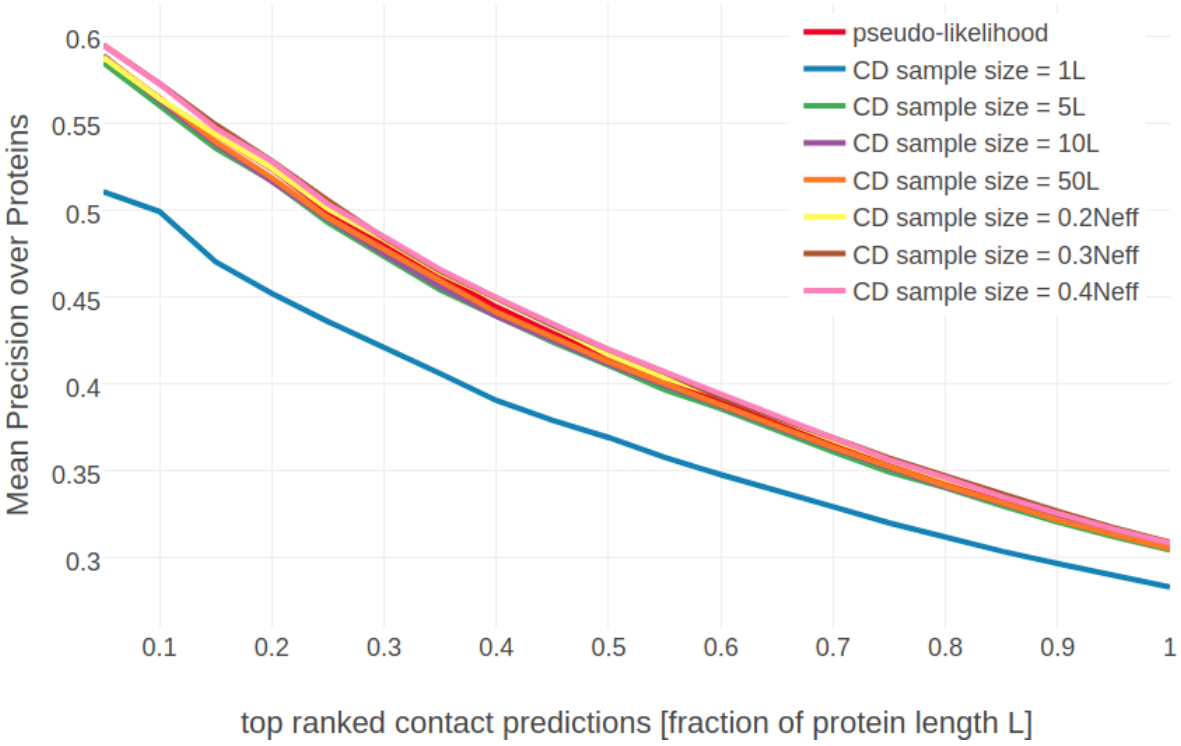

Figure 4.12: Mean precision for top ranked contact predictions over 288 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. **pseudo-likelihood**: couplings computed with pseudo-likelihood. **CD sample size = X** : contact scores computed from CD with SGD and varying number of sample size as specified in the legend. Sample size refers to the number of randomly selected sequences for Gibbs sampling. It is defined either as multiples of protein length L or as fraction of the effective number of sequences Neff.

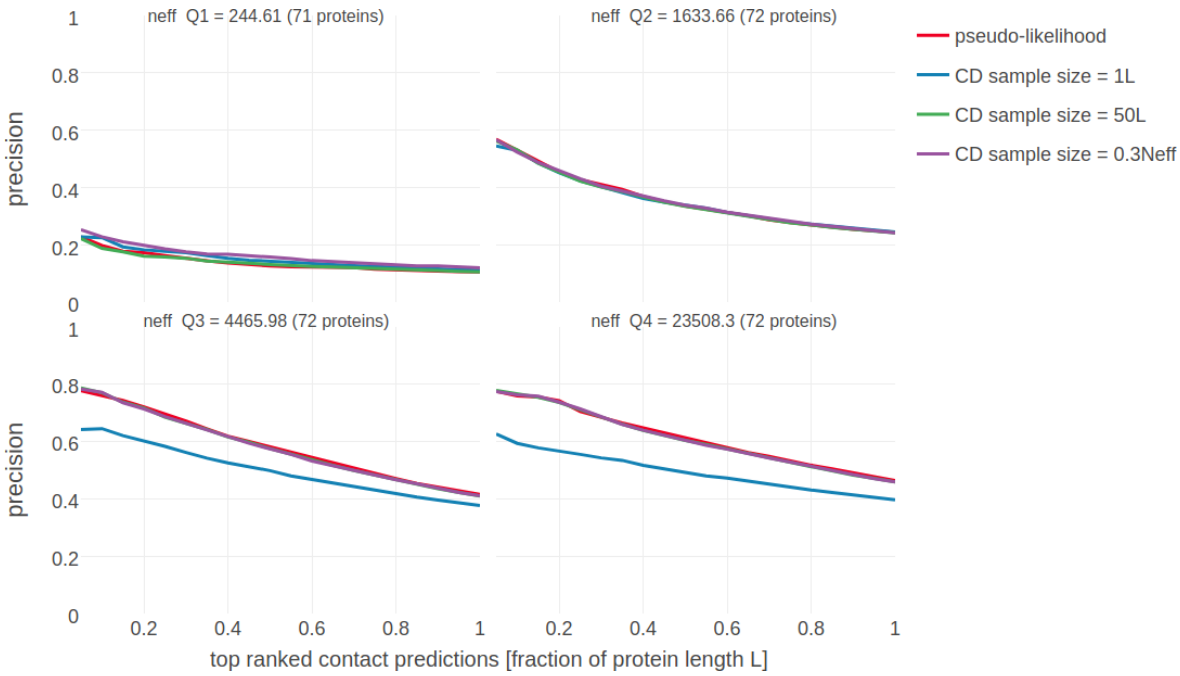

Figure 4.13: Mean precision for top ranked contact predictions over 300 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. **pseudo-likelihood**: contact scores computed from pseudo-likelihood. **CD sample size = X**: contact scores computed from CD optimized with SGD and varying number of sample size as specified in the legend. Sample size refers to the number of randomly selected sequences for Gibbs sampling. It is defined either as multiples of protein length L or as fraction of the effective number of sequences Neff.

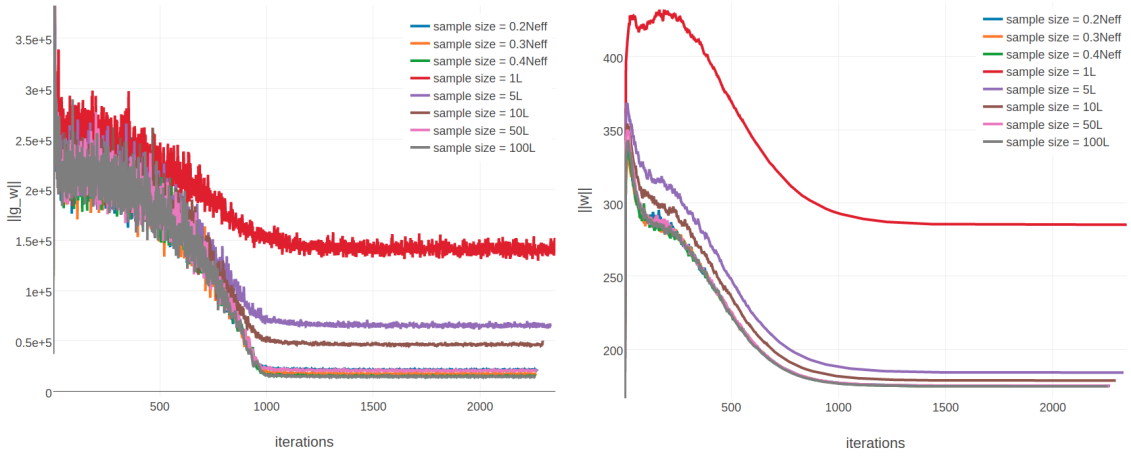

Figure 4.14: Monitoring parameter norm and gradient norm for protein 1c75_A_00 during SGD using different sample sizes. Protein 1c75_A_00 has length L=71 and 28078 sequences in the alignment (Neff=16808) **Left** L2-norm of the gradients for coupling parameters $||\mathbf{w}||_2$ (without contrbution of regularizer). The number of sequences, that is used for Gibbs sampling to approximate the gradient, is given in the legend. **Right** L2-norm of the coupling parameters $||\mathbf{w}||_2$. The number of sequences, that is used for Gibbs sampling to approximate the gradient, is given in the legend.

benchmark. Probably, the improved gradient when using a larger sample size helps to finetune the parameters, which only has a negligible effect on the contact score computed as APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. This hypothesis is confirmed when assessing the development of the L2 norm of coupling parameters $||\mathbf{w}||_2$ over optimization shown in right plot in Figure 4.14. The norm of the coupling parameters is almost indistinguishable after optimization when using a sample size of $50L$, $100L$ or $0.2 - 0.4$Neff.

It is not clear, why an improved gradient estimate due to sampling more sequences results in weaker performance for proteins with small alignments as could be seen in the previous benchmark in Figure 4.13. For protein chain 1aho_A_00 of length 64 and with 378 sequences (Neff=229), setting $S = 10L$ or $S = 50L$ which corresponds to using all $N = 378$ sequences for the sampling procedure to approximate the gradient, results in a mean precision over the top $0.1L$ - L contacts of 0.44, whereas using only $0.3N_{\text{eff}} = 69$ sequences gives a mean precision of 0.62. The left plot in Appendix Figure F.13 shows the development of the norm of the gradient and the norm of coupling parameters for this protein. As before, the gradient estimate improves and the norm of the gradient converges towards smaller values when more sequences are used in the Gibbs sampling process and therefore should lead to a better approximation of the likelihood. One explanation could be that this is some effect of overfitting, eventhough a regularizer is used and the norm of coupling parameters actually is smaller when using higher sampling sizes (see the right plot in Appendix Figure F.13).

4.4.2 Varying the number of Gibbs Steps

The default CD-k algorithm as described by Hinton in 2002 evolves the gibbs chain by only $k = 1$ full step [188]. As it has been pointed out in the literature, the CD-1 sampler represents a biased estimator because samples are not obtained from the stationary distribution of the Markov chain. Furthermore, it was found that sampling $k > 1$ steps gives more precise results at the cost of longer runtimes per gradient evaluation [191,200]. I analysed the impact on performance when the number of Gibbs steps is increased to 5 and 10.

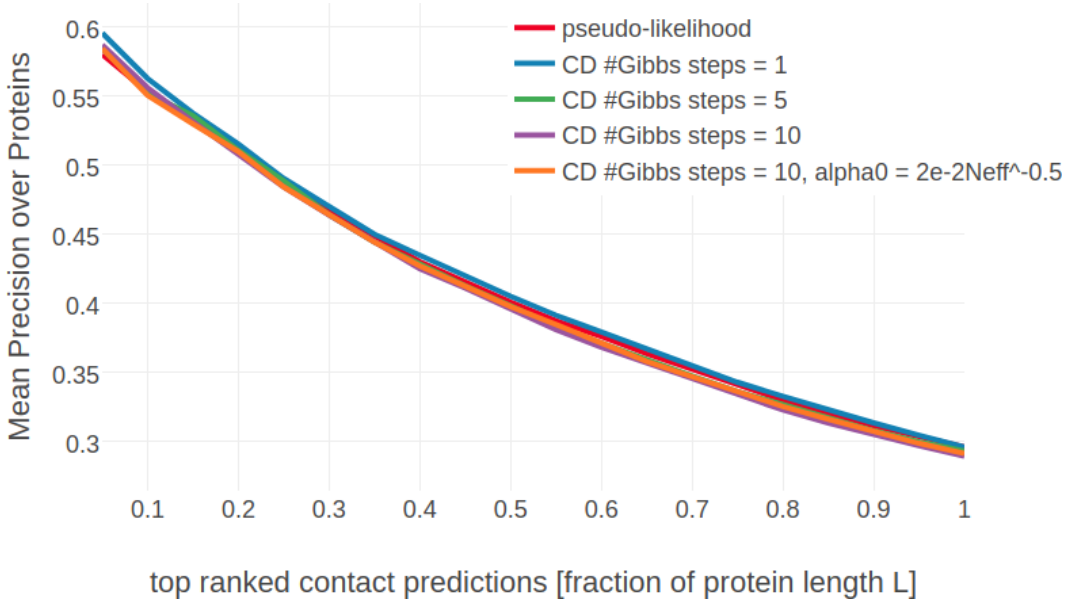

Figure 4.15: Mean precision for top ranked contact predictions over 300 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. **pseudo-likelihood**: contact scores computed from pseudo-likelihood. **CD #Gibbs steps = X**: contact scores computed from CD optimized with SGD and evolving each Markov chain using the number of Gibbs steps specified in the legend.

As can be seen in Figure 4.15, increasing the number of Gibbs steps does result in a slight drop of performance. When evaluating precision with respect to Neff it can be found that using more Gibbs sampling steps is especially disadvantageous for large alignments (see Appendix Figure F.11).

When evaluating single proteins, it can be observed that for proteins with small alignments the L2 norm of the parameters, $||\mathbf{w}||_2$, converges towards a different offset when using more than one Gibbs steps (see left plot in Figure 4.16). Naturally, the Markov chains can wander further away from their initialization when they are evolved over a longer time which results in a stronger gradient at the beginning of the optimization. Therefore and because the initial learning rate has been optimized for one Gibbs step, the parameter norm overshoots the optimum at the beginning. Even when lowering the initial learning rate from $\alpha_0 = \frac{5e-2}{\sqrt{N_{\text{eff}}}}$ to $\alpha_0 \in \left\{\frac{3e-2}{\sqrt{N_{\text{eff}}}}, \frac{2e-2}{\sqrt{N_{\text{eff}}}}, \frac{1e-2}{\sqrt{N_{\text{eff}}}}\right\}$, the SGD optimizer evidently approaches a different optimum. Surprisingly, the different optimum that is found for proteins with small alignments has no substantial impact on precision, as becomes evident from Figure F.11. For proteins with large alignments it can be observed that there is not one alternative solution to the parameters, but depending on the number of Gibbs steps and on the initial learning rate, α_0, the L2 norm over parameters converges towards various different offsets (see right plot in Figure 4.16). It is not clear how these observations can be interpreted, in particular given the fact, that the L2 norm of gradients, $||\nabla_{\mathbf{w}} LL(\mathbf{v}^*, \mathbf{w})||_2$, converges to the identical offset for all settings regardless of alignment size (see Appendix Figure F.14).

4.4.3 Persistent Contrastive Divergence

Finally I analysed PCD since it has often been found to be a better estimator of the likelihood gradient than CD. It has been suggested to use PCD with small learning rates and larger mini-batches because the fundamental assumption is that the model does not change to quickly

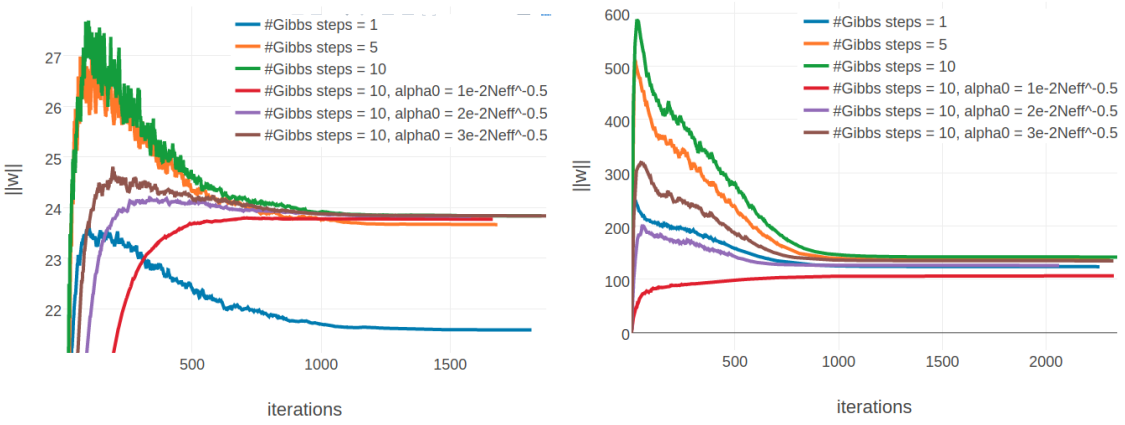

Figure 4.16: Monitoring parameter norm, $||\mathbf{w}||_2$, for protein 1aho_A_00 and 1c75_A_00 during SGD optimization using different number of Gibbs steps and initial learning rates, α_0. Number of Gibbs steps is given in the legend, as well as particular choices for the initial learning rate, when not using the default $\alpha_0 = \frac{5e-2}{\sqrt{N_{\text{eff}}}}$. **Left** Protein 1aho_A_00 has length L=64 and 378 sequences in the alignment (Neff=229) **Right** Protein 1c75_A_00 has length L=71 and 28078 sequences in the alignment (Neff=16808).

over iterations and therefore the Markov chains will stay close to the equilibrium distribition. However, using smaller learning rates and larger minibatches not only increases runtime but also requires tuning of the learning rate and learning rate schedule. Since it has been found, that CD is faster in learning at the beginning of the optimization, I tested a compromise, that uses CD-1 at the beginning of the optimization and when learning slows down, measured as the relative change of the norm of coupling parameters, I start PCD. This also has the advantage that the model has already approached the optimum and the parameters $\mathbf{w}$ will almost be constant over many updates.

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

Discussion: - as could be seen: improved gradients and different solutions do not translate into improved precision of top ranked contacts - APC corrected l2norm might not be an appropriate measure for CD couplings: look at correlation plot of couplings (pll vs cd) and l2norm (pll vs cd) and apc l2norm (pll vs cd) -> differences vanish - ranking of residues might not be influenced by subtle changes in parameters when crude l2norm is computed -> rank plot: merged list of top ranked contacts from both methods what can we see: - generally pll has stronger scores (see also boxplot over all proteins? statistic?) - ranking is very similar, especially for top ranked contacts (thats why benchmark plots so similar)

4.5 Using ADAM to optimize Contrastive Divergence

It is possible that the learning rate schedule and decay rate need to be tuned with respect to the number of Gibbs steps used for sampling because different choices of Gibbs steps influence the size of the gradient. The same might apply to usign persistent contrasticve divergence.

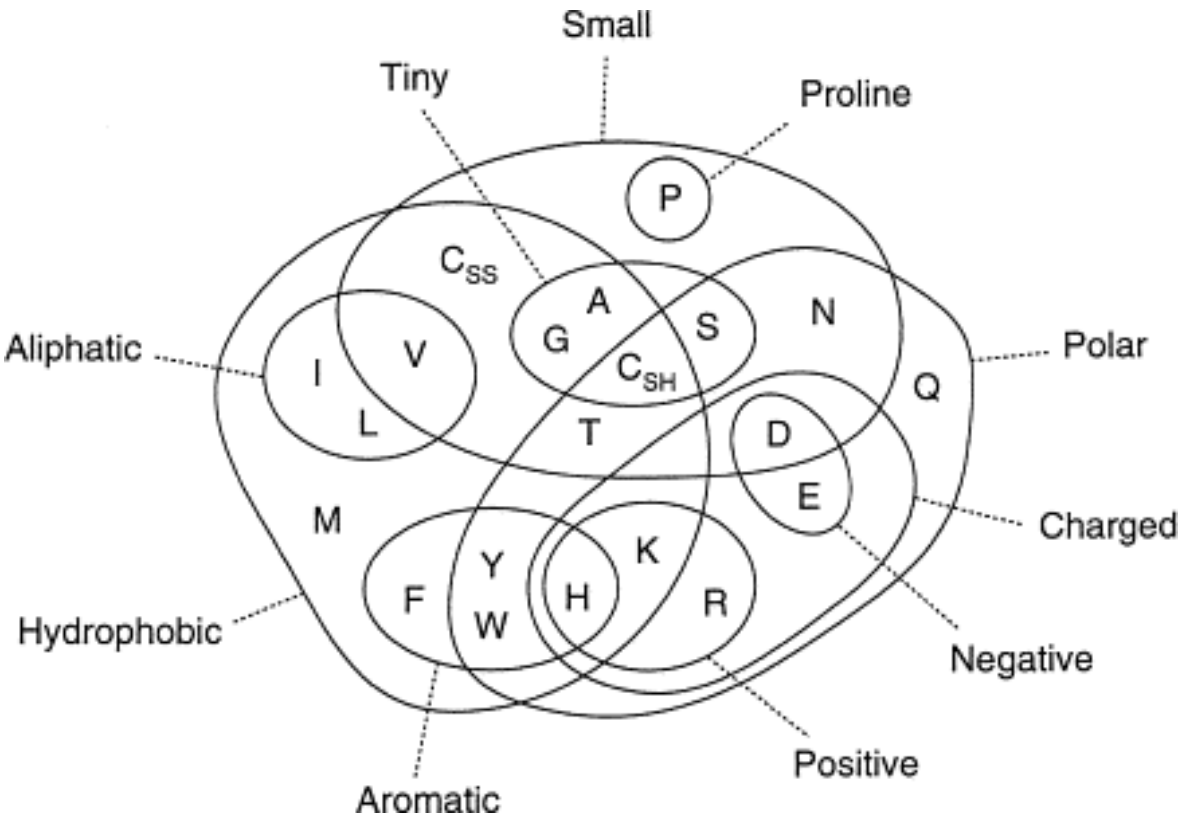

Figure 4.17: dideldum

Hyperparameter optimization with stochastic gradient descent is a time-consuming task requiring manual intervention for each of the different settings. There exist many variants of stochastic gradient descent algorithms that deal with the aforementioned challenges e.g. speeding up convergence rates using momentum or defining adaptive learning rates for each parameter [192]. One of these SGD variants is Adaptive Moment Estimation ($ADAM$) [205], an algorithm that computes per-parameter learning rates including momentum (see methods section 7.8.1 for details). A major advantage of $ADAM$ over pure SGD is that it does not require tuning many hyperparameters as the default values have been found to work quite well. $ADAM$ will be compared to the manually tuned SGD optimizer in section 4.5. I used $ADAM$ to find out whether its automatic adjusting of learning rates can find better parameter estimates than [sgd] for using different gibbs steps and pcd.

A problem when using $ADAM$ is that the necessarty condidtion $\sum_{a,b=1}^{20} w_{ijab} = 0$ is violated (see section 4.2.1).

4.6 Comparing CD couplings to pLL couplings

Scatter Plots for couplings wijab: nice correlation - pearson correlation? gibt 3 outlier und das sind auch noch die staerksten couplings hier sieht man besser als beim boxplot, das CD kleinere couplings hat (absolut)

Boxplots for couplings wijab: sieht ma nicht viel

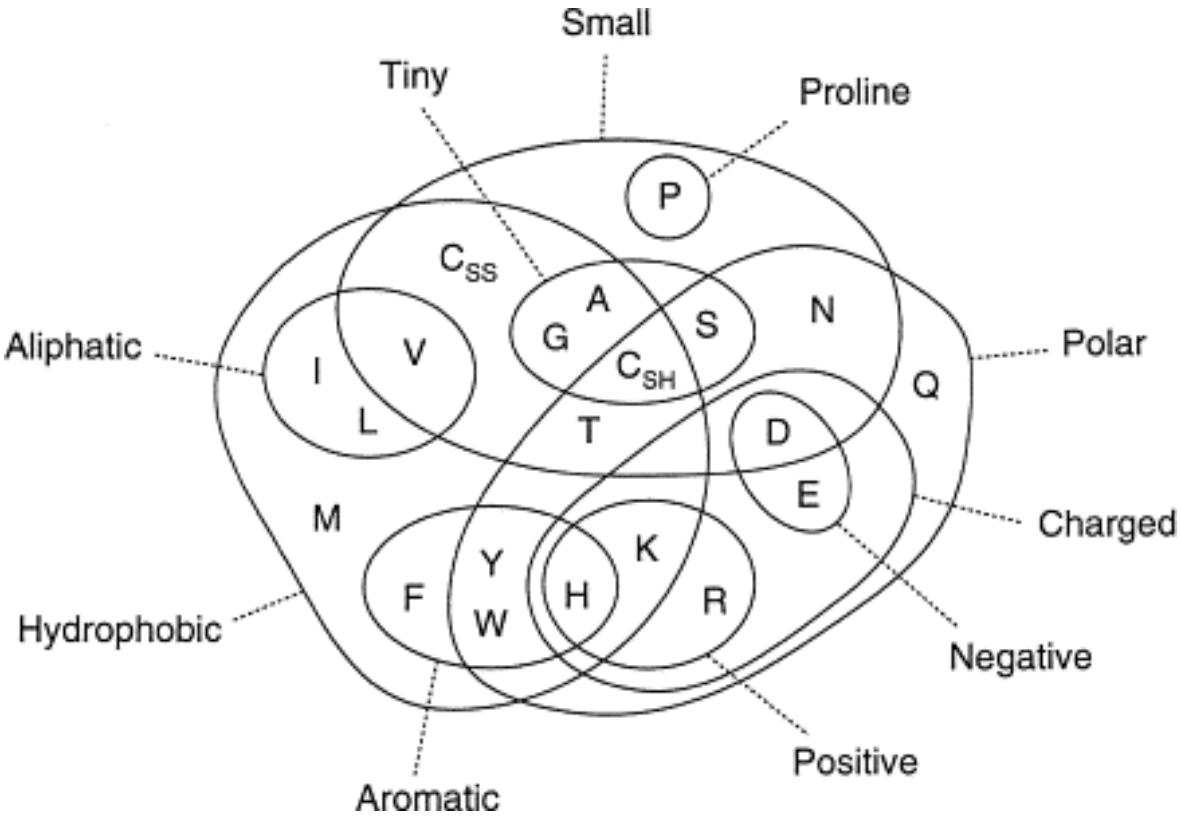

Figure 4.18: dideldum

Scatter for l2norm over couplings ||wij||_2: hier sieht man sehr schoen, dass CD systematisch kleinere scores hat als pLL und man sieht die drei outlier

Boxplots for l2norm over couplings ||wij||_2: contrastive divergence hat kleineren scores signifikant?

Scatter for l2norm-apc over couplings ||wij||_2 - apc: das gleiche mit apc: systematisch kleinere scores (absolut)

Boxplots for l2norm-apc over couplings ||wij||_2 - apc: auch mit apc: es gibt weniger starke scores mit CD signifikant?

scipy.stats.wilcoxon(l2norm_apc_pll, l2norm_apc_cd) #WilcoxonResult(statistic=187035.0, pvalue=0.029710790280912919)

scipy.stats.ranksums(l2norm_apc_pll, l2norm_apc_cd) #RanksumsResult(statistic=0.38719674947266086, pvalue=0.69861055638055758)

scipy.stats.kendalltau(l2norm_apc_pll, l2norm_apc_cd) #KendalltauResult(correlation=0.76655297812416368, pvalue=1.3239009316846667e-260)

scipy.stats.spearmanr(l2norm_apc_pll, l2norm_apc_cd) #SpearmanrResult(correlation=0.92532449605319178, pvalue=0.0)

scipy.stats.ks_2samp(l2norm_apc_pll, l2norm_apc_cd) Ks_2sampResult(statistic=0.067552602436323439, pvalue=0.030922873101286375)

The two score distributions are significantly different (kolmogorov smirnov test, two-sided p-value of 0.03).

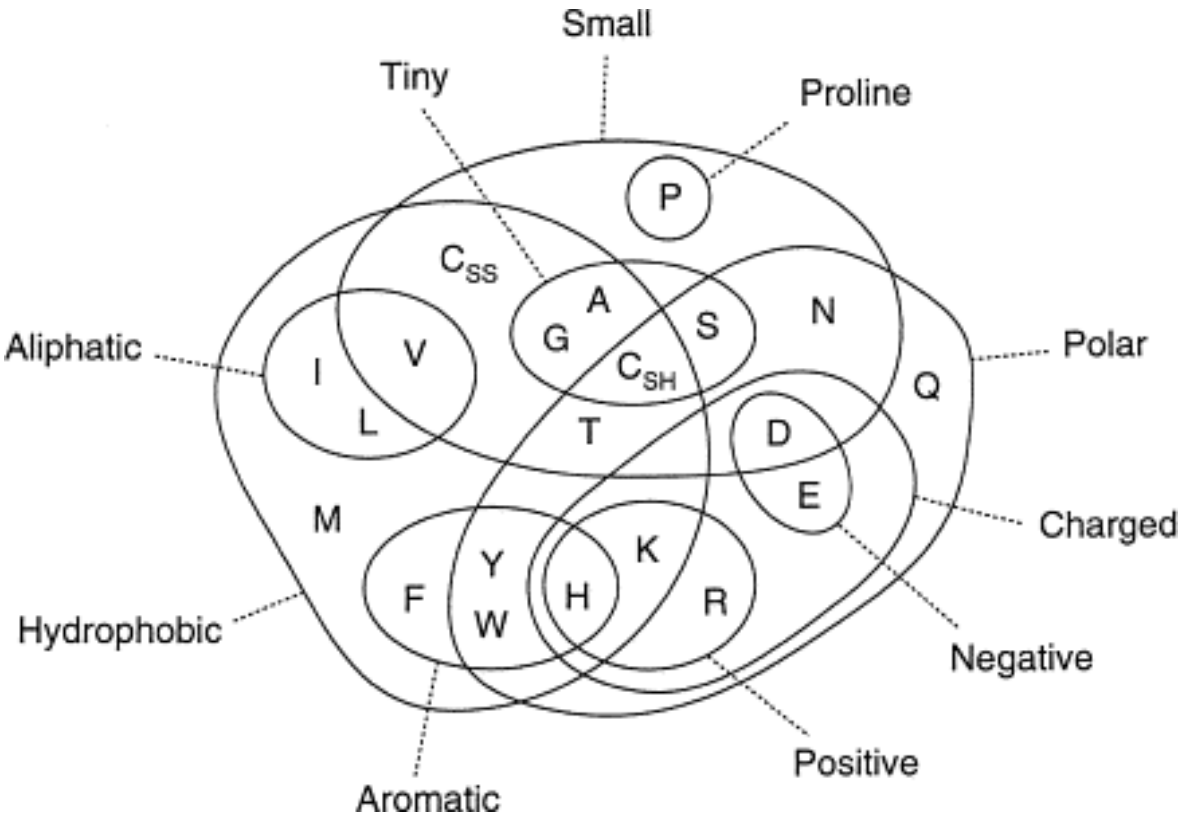

Figure 4.19: dideldum

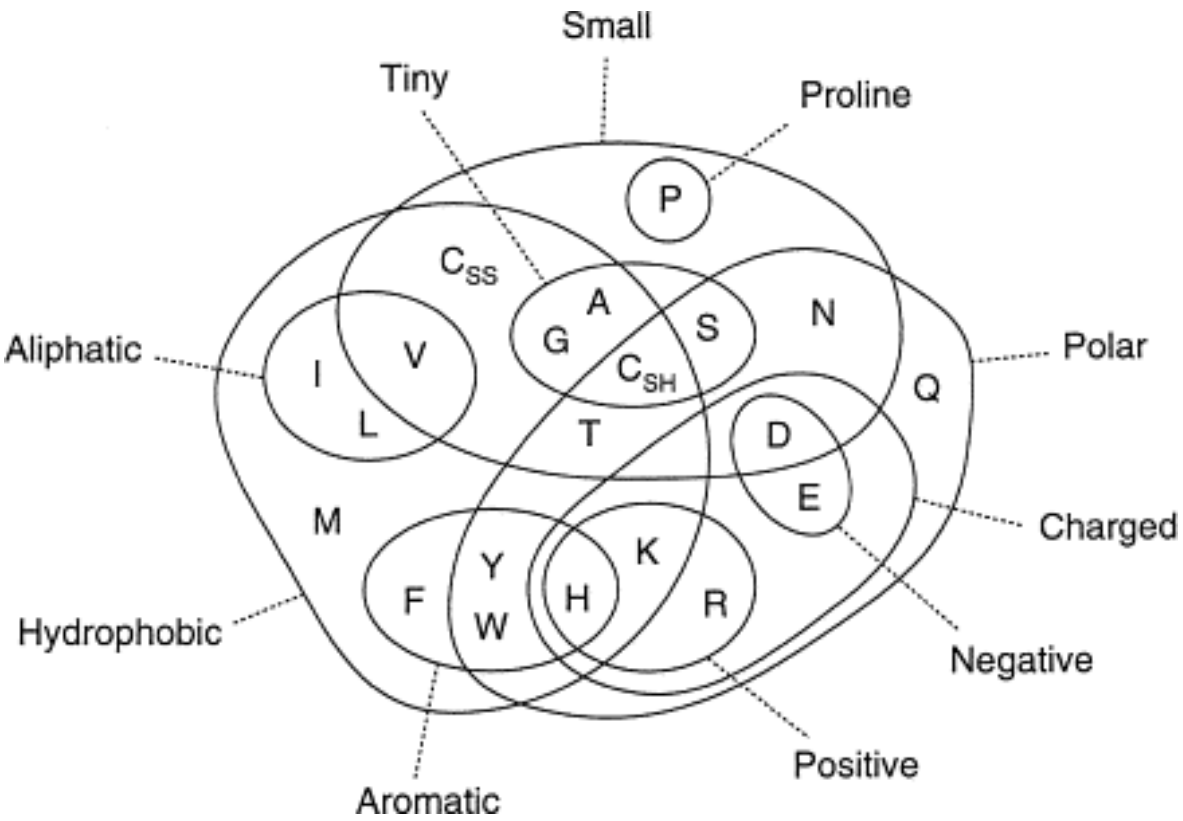

Figure 4.20: dideldum

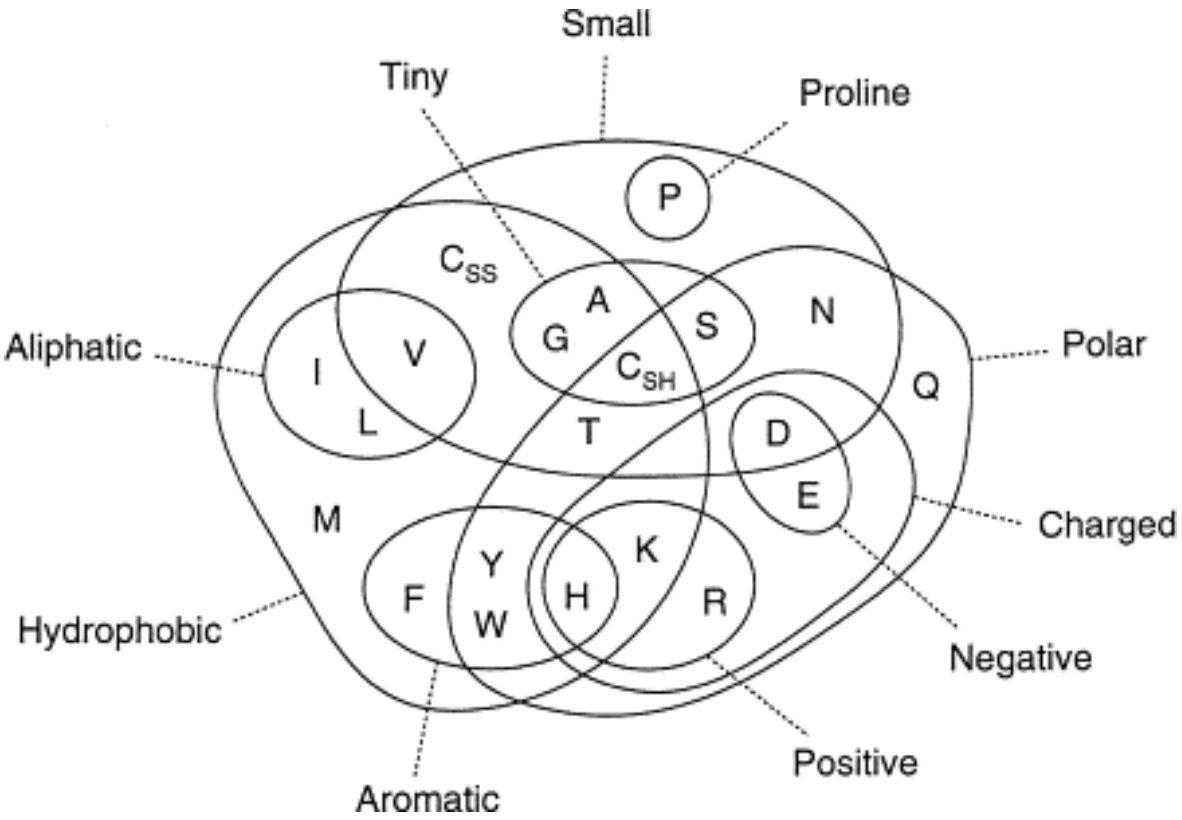

Figure 4.21: dideldum

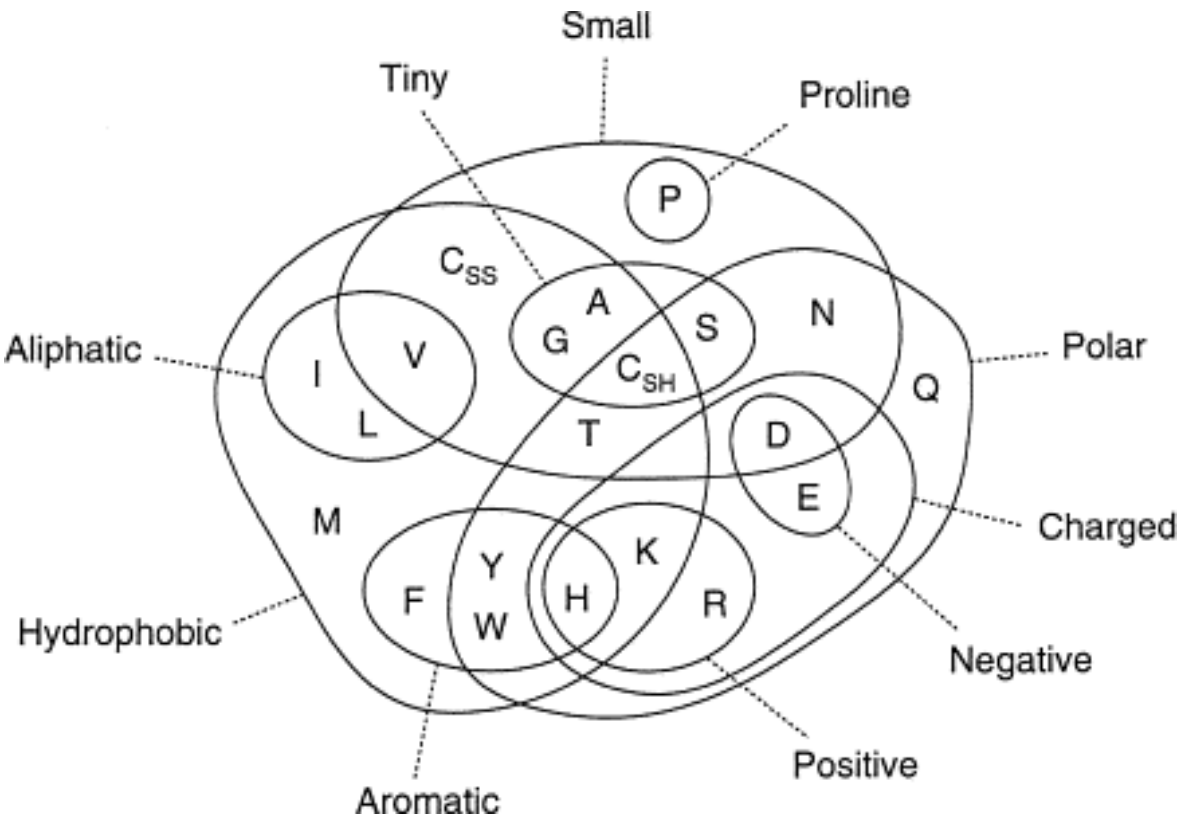

Figure 4.22: dideldum

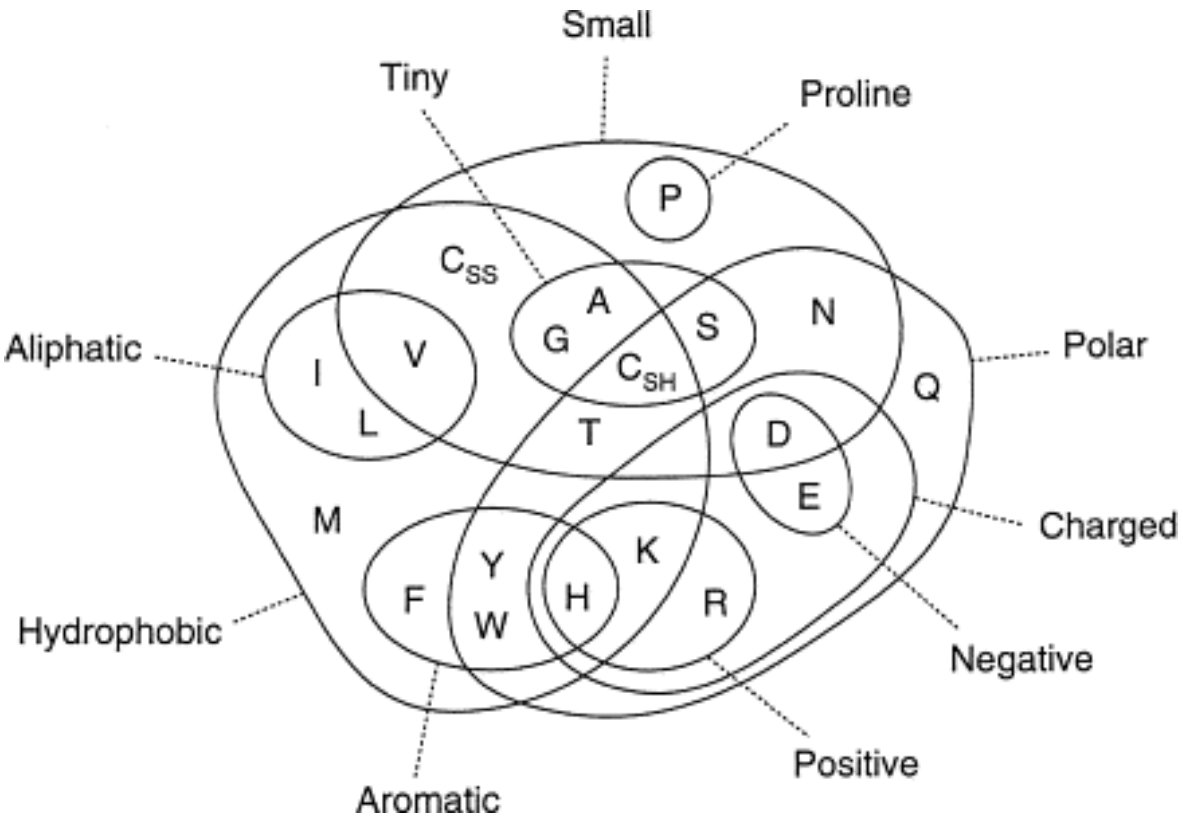

Figure 4.23: dideldum

Comparing the ranking: geiler plot ranking ist sehr aehnlich, besonders fuer top contacts nur diese drei outlier sind bei CD sehr viel hoeher gerankt als bei pLL

Comparing the ranking with q-q plot:

Showing the contact maps: pLL

CD

Showing precision:

- convergence criterion: mention paper [198]

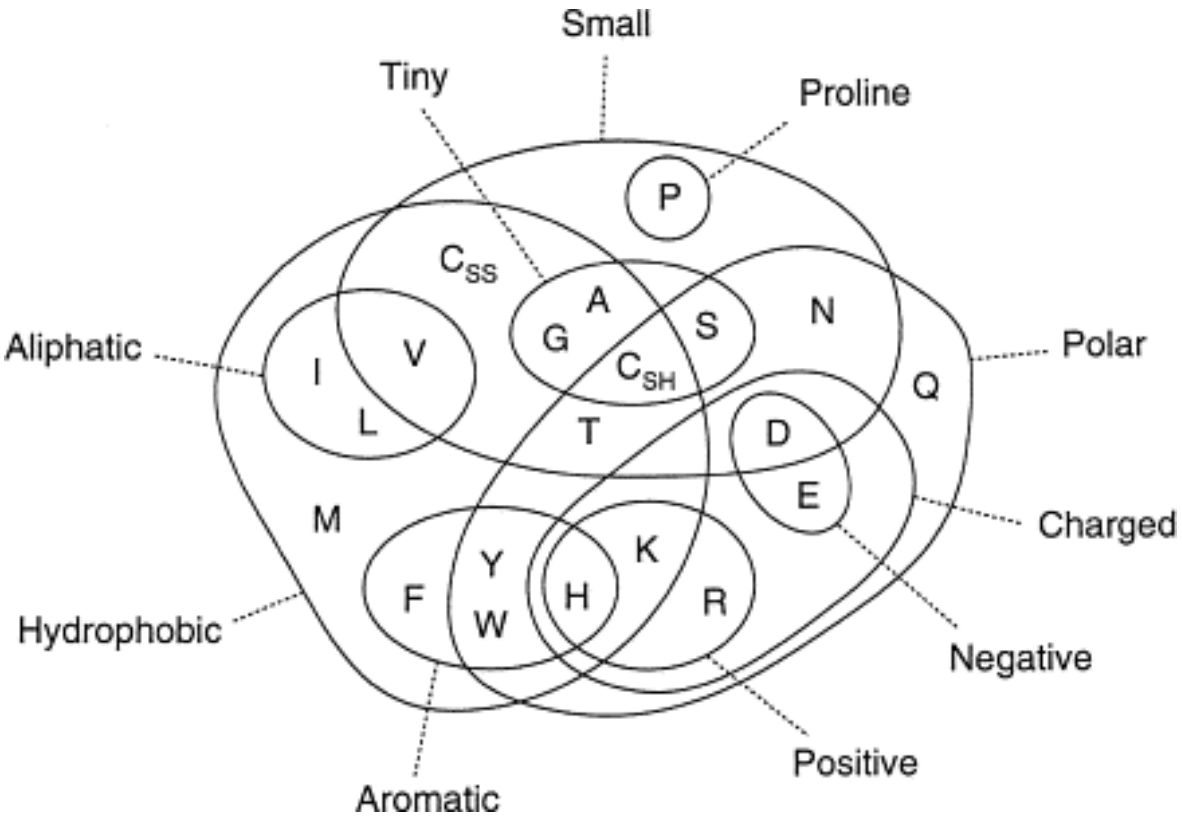

Figure 4.24: dideldum

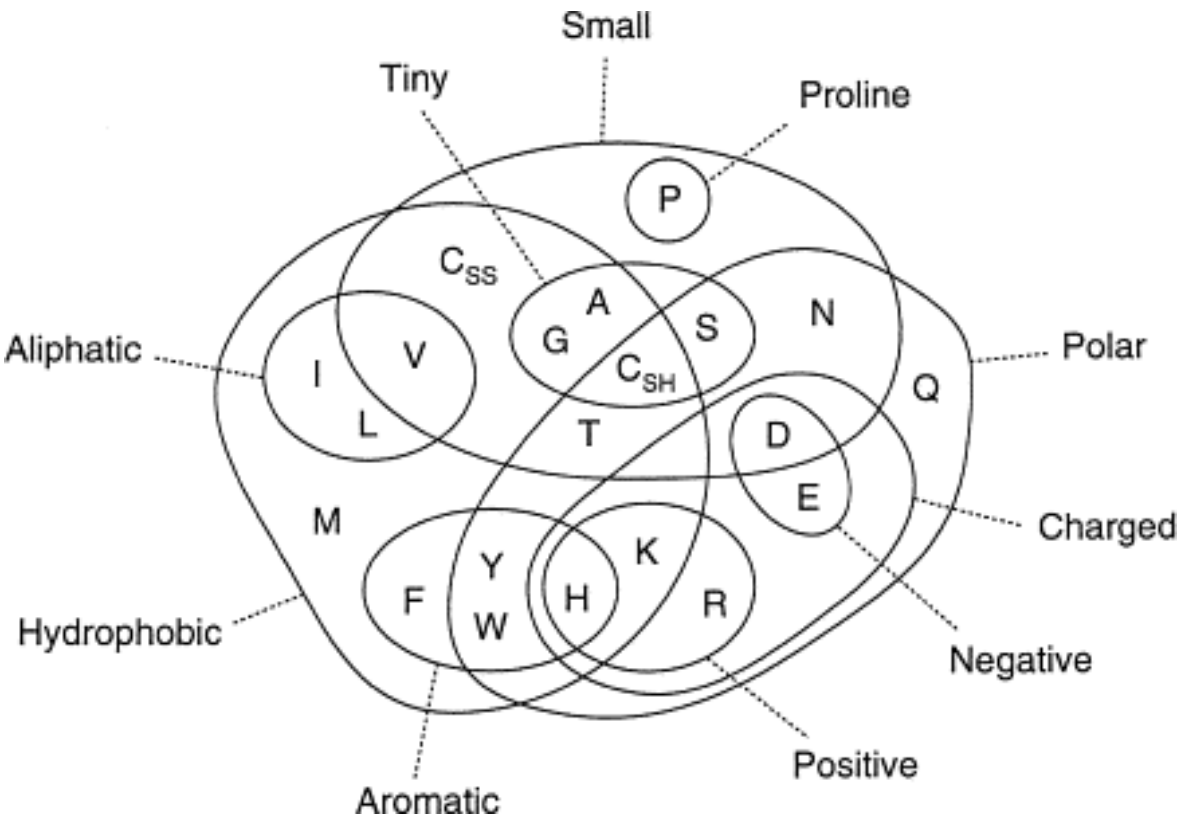

Figure 4.25: dideldum

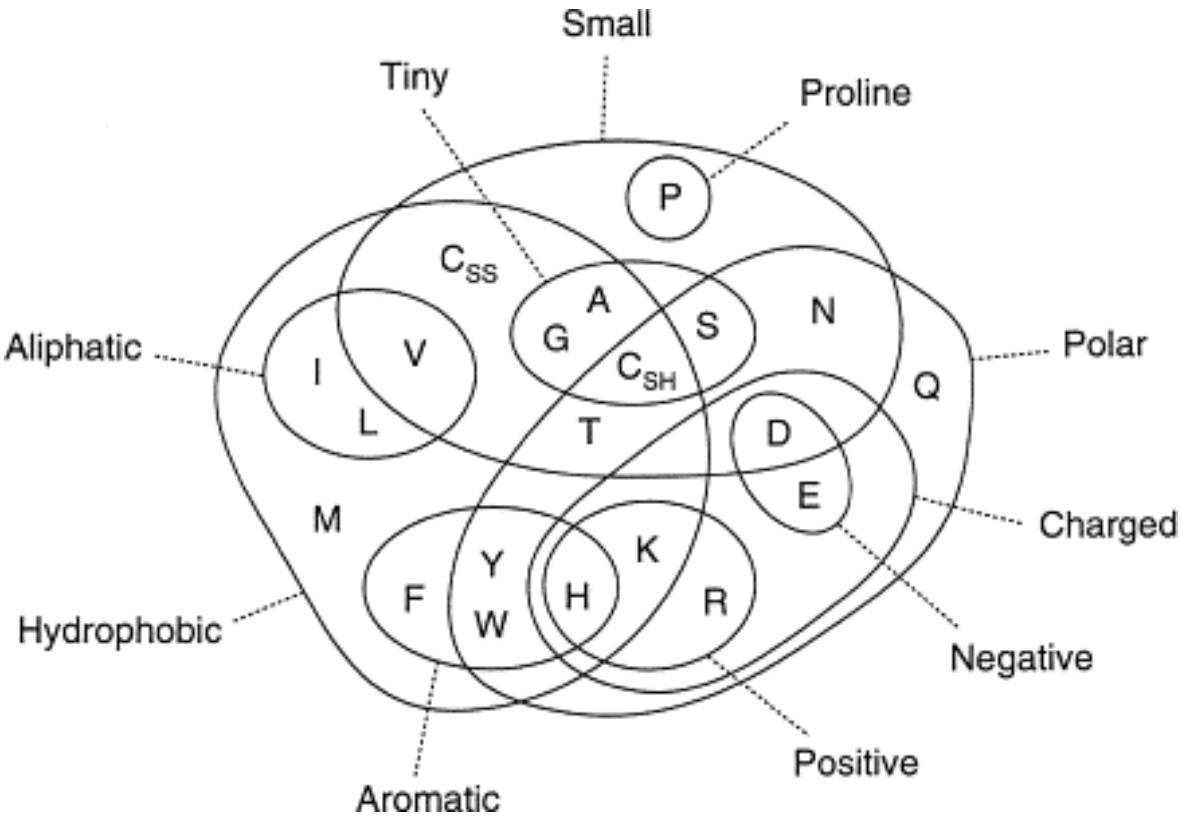

Figure 4.26: dideldum

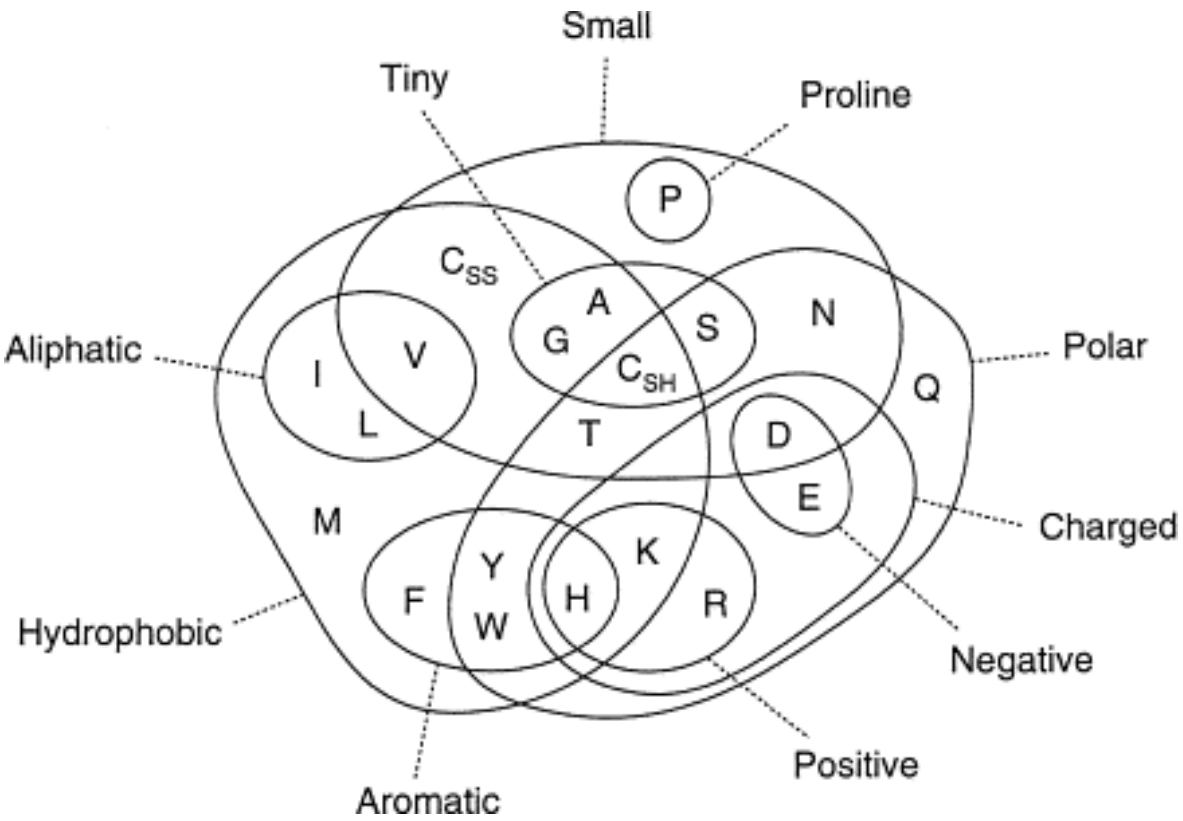

Figure 4.27: dideldum

5

Random Forest Contact Prior

The wealth of successful meta-predictors presented in section 2.3 highlights the importance to exploit other sources of information apart from coevolution statistics. Much information about residue interactions is typically contained in single position features that can be predicted from local sequence profiles, such as secondary structure, solvent accessibility or contact number, and in pairwise features such as the contact prediction scores for residue pairs (i, j) from a simple local statistical methods as presented in section 2.1.

For example, predictions of secondary structure elements and solvent accessibility are used by almost all modern machine learning predictors, such as MetaPsicov [84], NeBCon [87], EPSILON-CP [86], PconsC3 [82]. Other frequently used sequence derived features include pairwise contact potentials, sequence separation and conservation measures such as column entropy [84,87,206].

In the following sections I present a random forest classifier that uses sequence derived features to distinguish contacts from non-contacts. Methods section 7.15 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 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 [207–209]. They are easy to implement, fast to train and can handle large numbers of features due to implicit feature selection [210].

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 drawn with replacement from the training set and has the same size as the training set (i.e., a bootstrap sample)

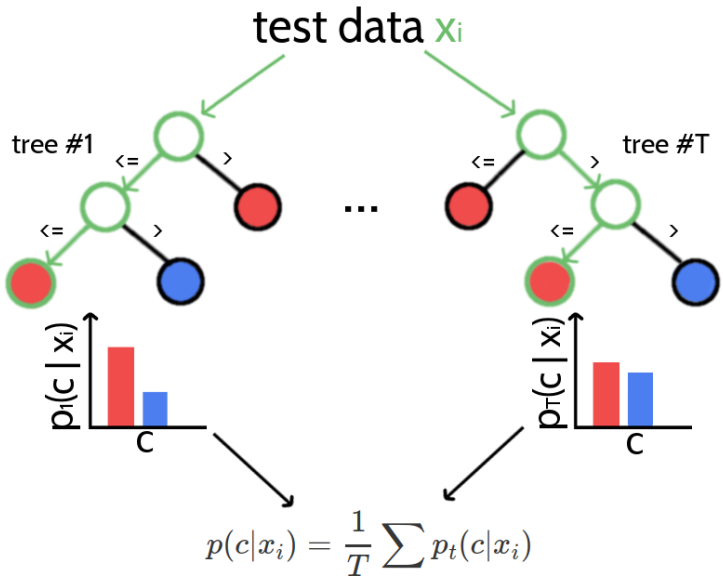

Figure 5.1: Classifying new data with random forests. A new data sample is run down every tree in the forest until it ends up in a leaf node. Every leaf node has associated class probabilities $p(c)$ reflecting the fraction of training samples at this leaf node belonging to every class c. The color of the leaf nodes reflects the class with highest probability. The predictions from all trees in form of the class probabilties are averaged and yield the final prediction.

2. every split of a node is evaluated on a random subset of features

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 [211]. It is still advisable to restrict the depth of single trees in a random forest, not only to counteract overfitting but also to reduce model complexity and to speedup the algorithm.

Random forests are capable of regression and classification tasks. For classification, predictions for new data are obtained by running each 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. Figure 5.1 visualizes the procedure of classifying a new data sample.

Typically, *Gini impurity*, which is a computationally efficient approximation to the entropy, is used as a split criterion to evaluate the quality of a split. 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. For every feature f in the random subset that is considered for splitting a particular node N, the *decrease in Gini impurity* ΔGI_f will be computed as,

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

where p_{left} and p_{right} refers to the fraction of samples ending up in the left and right child node respectively [210]. The feature f with highest ΔGI_f over the two resulting child node subsets will be used to split the data set at the given node N.

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 [212].

5.2 Hyperparameter Optimization for Random Forest

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 [213]. 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 leaf node size regarding the minimal number of data samples ending in a leaf node. Again, a positive side-effect of pruning and requiring minimal leaf node size is a speedup of the algorithm. [211]

In the following, I use 5-fold cross-validation to identify the optimal architecture of the random forest. Details about the training set and he cross-validation procedure can be found in method section 7.16. 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:

- $n\_estimators \in \{100, 500, 1000\}$
- $max\_depth \in \{10, 100, 1000, None\}$

Figure 5.2 shows that the top five parameter combinations perform nearly identical. Random forests with 1000 trees perform slightly better than models constituting 500 trees, irrespective of the depth of the trees. In order to keep model complexity small, I chose `n_estimators=1000` and `max_depth=100` for further analysis.

Next, I optimized the parameters *min_samples_leaf*, defining the minimum number of samples required at a leaf node and *max_features*, defining the number of randomly selected features considered for each split using the following settings:

- $min\_samples\_leaf \in \{1, 10, 100\}$
- $max\_features \in \{8, 16, 38, 75\}$ representing $\sqrt{N}$, $\log 2N$, $0.15N$ and $0.3N$ respectively with $N = 250$ being the number of features listed in method section 7.15.

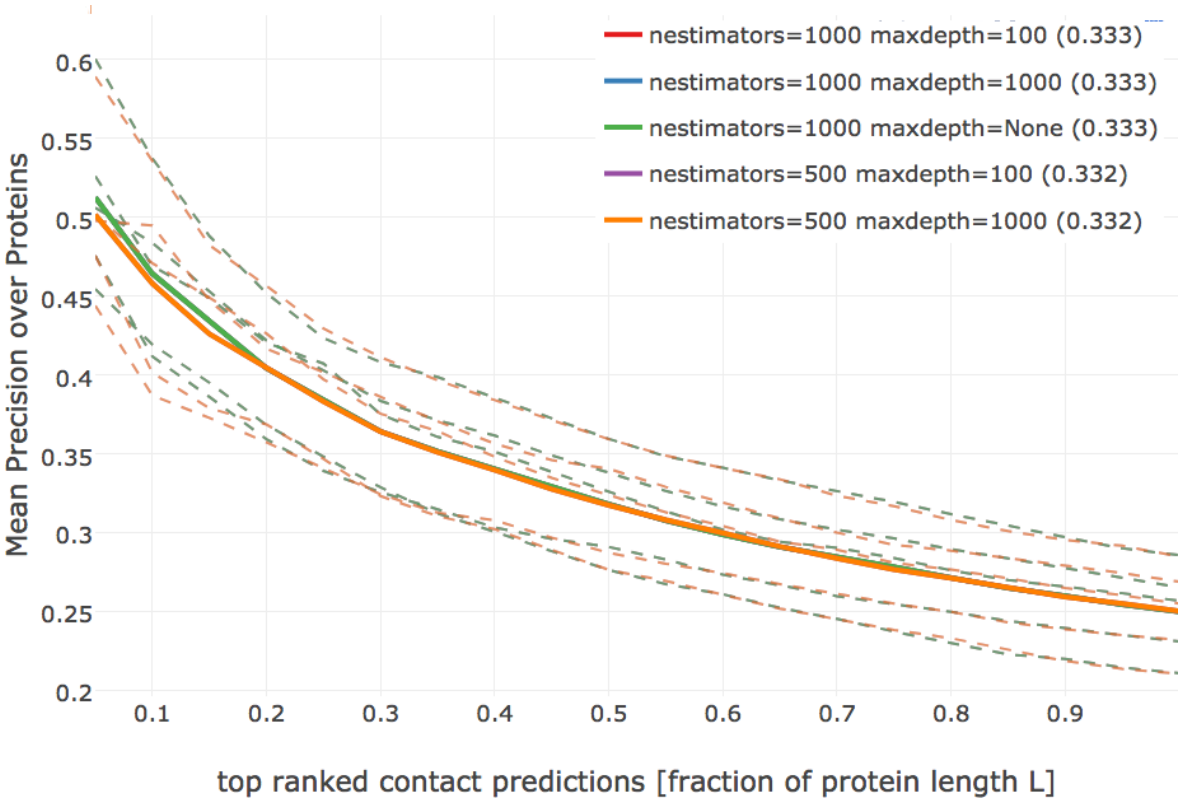

Figure 5.2: Mean precision over 200 proteins against highest scoring contact predictions from random forest models for different settings of *n_estimators* and *max_depth*. Dashed lines show the performance of models that have been learned on the five different subsets of training data. Solid lines give the mean precision over the five models. Only those models are shown that yielded the five highest mean precision values (given in parantheses in the legend). Random forest models with 1000 trees and maximum depth of trees of either 100, 1000 or unrestricted tree depth perform nearly identical (lines overlap). Random forest models with 500 trees and *max_depth*=10 or *max_depth*=100 perform slightly worse.

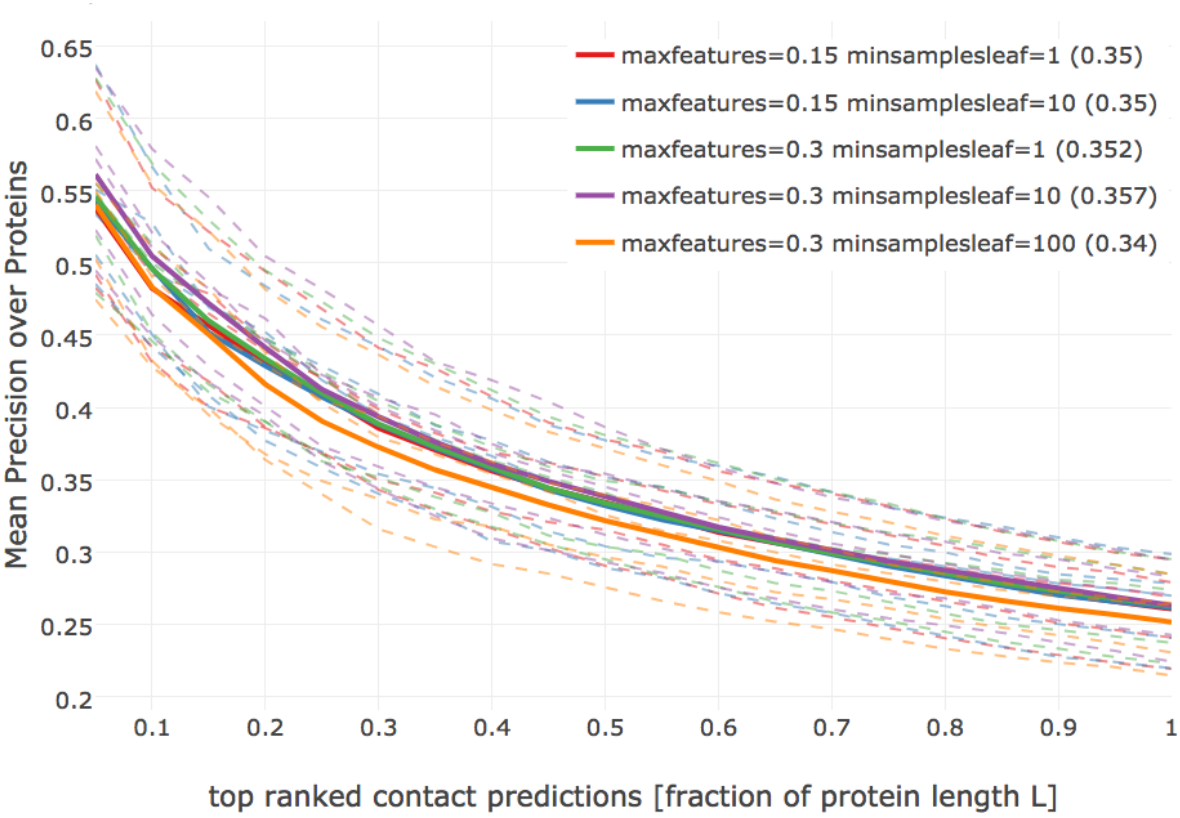

Figure 5.3: Mean precision over 200 proteins against highest scoring contact predictions from random forest models with different settings of *min_samples_leaf* and *max_features*. Dashed lines show the performance of models that have been learned on the five different subsets of training data. Solid lines give the mean precision over the five models. Only those models are shown that yielded the five best mean precision values (given in parantheses in the legend).

Randomly selecting 30% of features (=75 features) and requiring at least 10 samples per leaf gives highest mean precision as can be seen in Figure 5.3. I chose `max_features=0.30` 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=0.30`).

- proportion of contacts/non-contacts $\in \{1:2, 1:5, 1:10, 1:20\}$ while keeping total dataset size fixed at 300,000 residue pairs
- window size: $\in \{5, 7, 9, 11\}$
- non-contact threshold $\in \{8, 15, 20\}$

As can be seen in appendix G.2 and G.3, 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 G.4. These estimates might be biased in a way since the random forest hyperparameters have been optimized on a dataset using exactly these optimal settings.

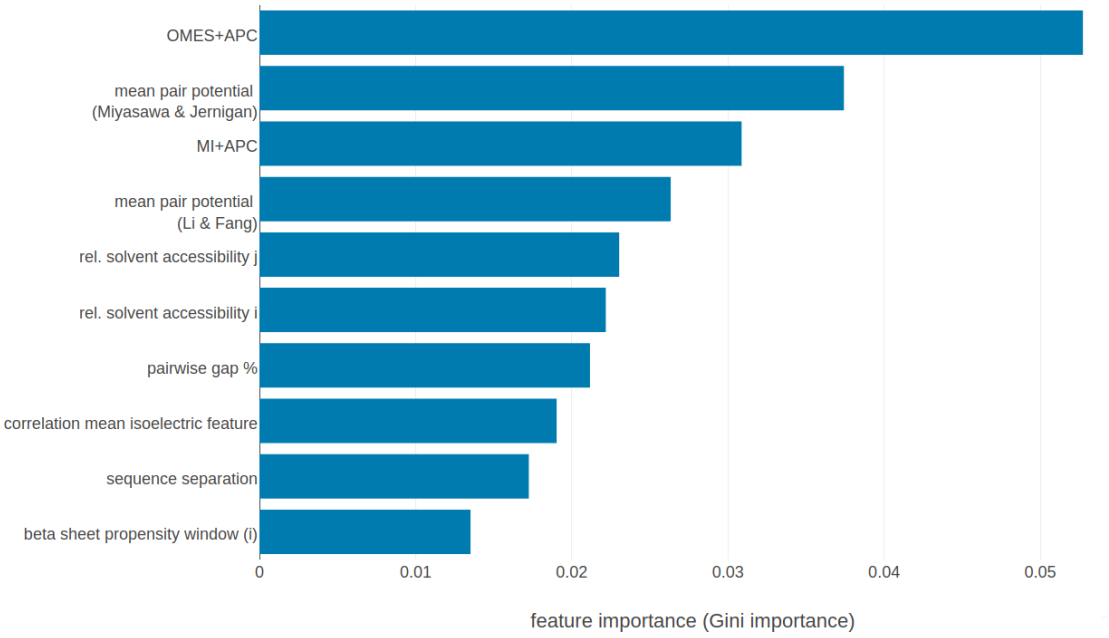

Figure 5.4: Top ten features ranked according to *Gini importance*. **OMES+APC**: APC corrected OMES score according to Fodor&Aldrich [214]. **mean pair potential (Miyasawa & Jernigan)**: average quasi-chemical energy of transfer of amino acids from water to the protein environment [215]. **MI+APC**: APC corrected mutual information between amino acid counts (using pseudo-counts). **mean pair potential (Li&Fang)**: average general contact potential by Li & Fang [69]. **rel. solvent accessibilty i(j)**: RSA score computed with Netsurfp (v1.0) [216] for position i(j). **pairwise gap%**: percentage of gapped sequences at either position i and j. **correlation mean isoelectric feature**: Pearson correlation between the mean isoelectric point feature (according to Zimmermann et al., 1968) for positions i and j. **sequence separation**: |j-i|. **beta sheet propensity window(i)**: beta-sheet propensity according to Psipred [217] computed within a window of five positions around i. eatures are described in detail in methods section 7.15.

5.3 Evaluating Random Forest Model as Contact Predictor

I trained a random forest classifier on the feature set described in methods section 7.15 and using the optimal hyperparameters identified with 5-fold cross-validation as described in the last section.

Figure 5.4 shows the ranking of the ten most important features according to *Gini importance*. Both local statistical contact scores, *OMES* [214] and MI (mutual information between amino acid counts), constitute the most important features besides the mean pair potentials acording to Miyazawa & Jernigan [215] and Li&Fang[69]. Further important features include the relative solvent accessibility at both pair positions, the total percentage of gaps at both positions, the correlation between mean isoelectric point property at both positions, sequence separation and the beta-sheet propensity in a window of size five around position i.

Many features have low *Gini importance* scores which means they are rarely considered for splitting a node and can most likely be removed from the dataset. Removing irrelevant features from the dataset is a convenient procedure to reduce model complexity. It has been found, that prediction performance might even increase after removing the most irrelevant features [210]. 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 features related to amino

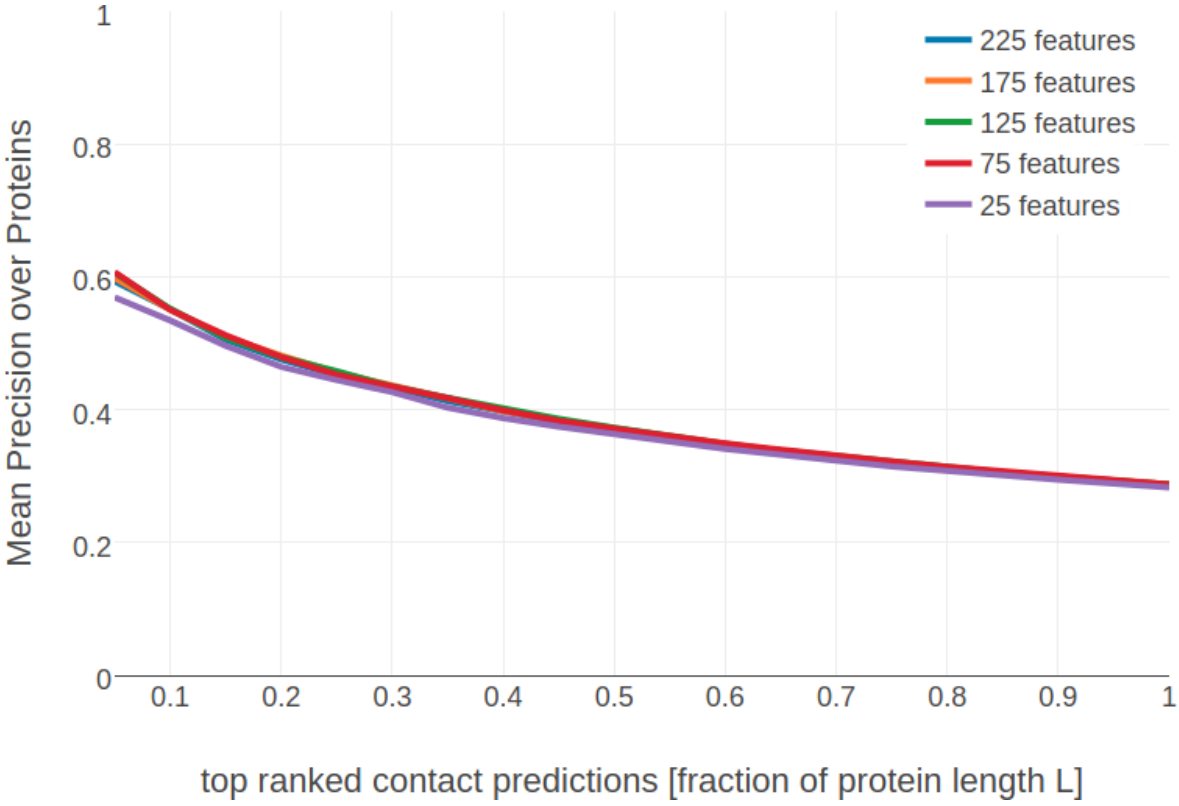

Figure 5.5: Mean precision of top ranked predictions over 200 proteins for random forest models trained on subsets of features of decreasing importance. Subsets of features have been selected as described in methods section 7.16.1.

acid composition), the performance of their predictor increased slightly [86]. Other studies have also emphasized the importance of feature selection to improve performance and reduce model complexity [67,69].

As described in methods section 7.16.1, I performed feature selection by evaluating model performance on subsets of features of decreasing importance. Most models trained on subsets of the total feature space perform nearly identical compared to the model trained on all features, as can be seen in Figure 5.5. Performance of the random forest models drops noticeably when using only the 25 most important features. For the further analysis I am using the random forest model trained on the 75 most important features as this model constitutes the smallest set of features while performing nearly identical compared to the model trained on the complete feature set.

Figure 5.6 shows the mean precision for the random forest model trained on the 75 most important features. The random forest model has a mean precision of 0.33 for the top $0.5 \cdot L$ contacts compared to a precision of 0.47 for pseudo-likelihood. Furthermore, the random forest model improves approximately ten percentage points in precision over the local statistical contact scores, *OMES* and mutual information (MI). Both methods comprise important features of the random forest model as can be seen in Figure 5.4.

When analysing performance with respect to alignment size it can be found that the random forest model outperforms the pseudo-likelihood score for small alignments (see Figure 5.7). Both, local statistial models *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

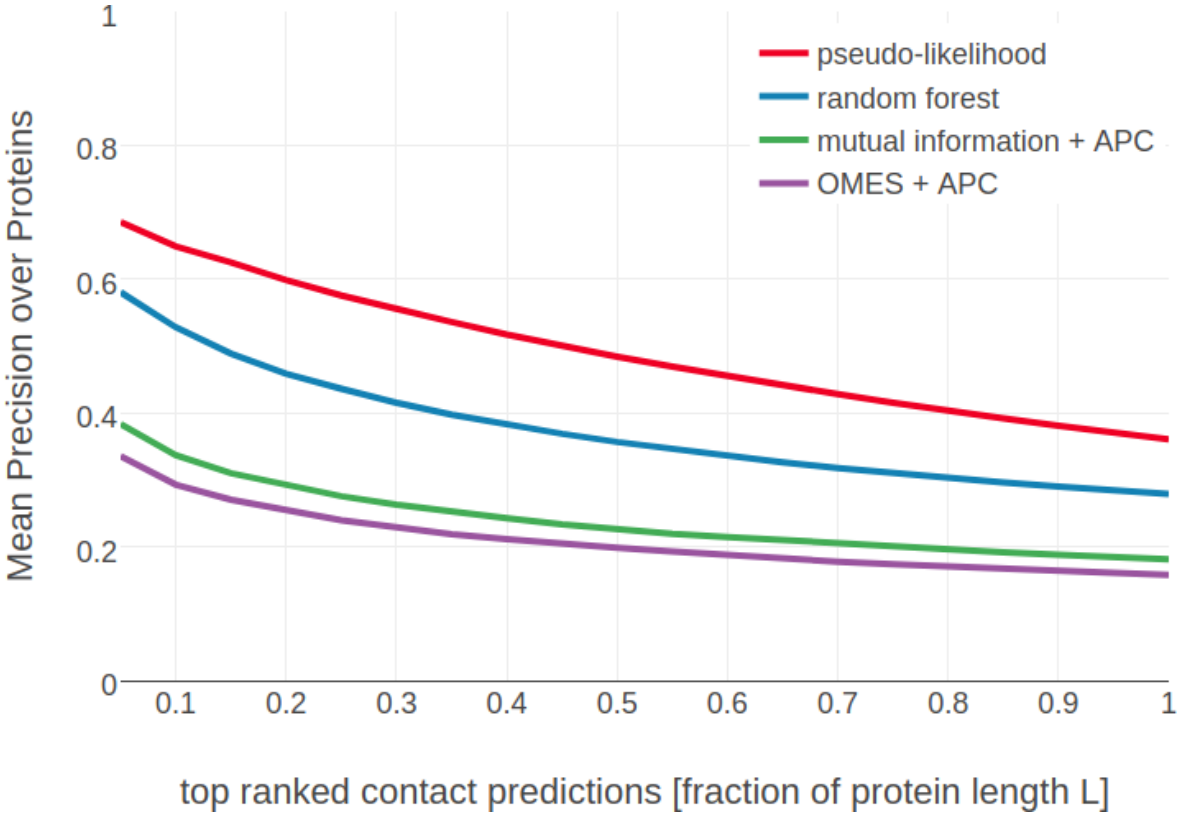

Figure 5.6: Mean precision for top ranked contacts on a test set of 1000 proteins. **pseudo-likelihood** = APC corrected Frobenius norm of couplings computed with pseudo-likelihood. **random forest** = random forest model trained on 75 sequence derived features. **OMES** = APC corrected *OMES* contact score according to Fodor&Aldrich [214]. **mutual information** = APC corrected mutual information between amino acid counts (using pseudo-counts).

[82,86].

5.4 Using Contact Scores as Additional Features

Figure 5.7 shows that the random forest predictor improves over the pseudo-likelihood coevolution method when the alignment consists of only few sequences. In order to assess this improvement in a more direct manner, it is possible to build a combined random forest predictor that is not only trained on the sequence derived features but also on the pseudo-likelihood contact score as an additional feature. As expected, the pseudo-likelihood score comprises the most important feature in the model (see Appendix Figure G.1) followed by the same sequence features that were found in the previous analysis in Figure 5.4. Models trained on subsets of features as described in method section 7.16.1 perform equally well as the model trained on the complete set of features (see Appendix Figure G.2). Only the model trained on the 26 most important features has slighlty decreased precision for the top L/10 ranked contacts. The model trained on 76 features was selected as the final model and its performance can be examined in Figure 5.8. The combination of simple sequence features with the coevolution pseudo-likelihood contact score indeed improves predictive power for the random forest model over both single approaches. Especially for small alignments, the improvement is substantial as can be seen in in the left upper plot in Figure 5.9. In contrast, the improvement on large alignments (right lower plot in Figure 5.9) is small, as the gain from simple sequence features compared to the much more powerful coevolution signals is

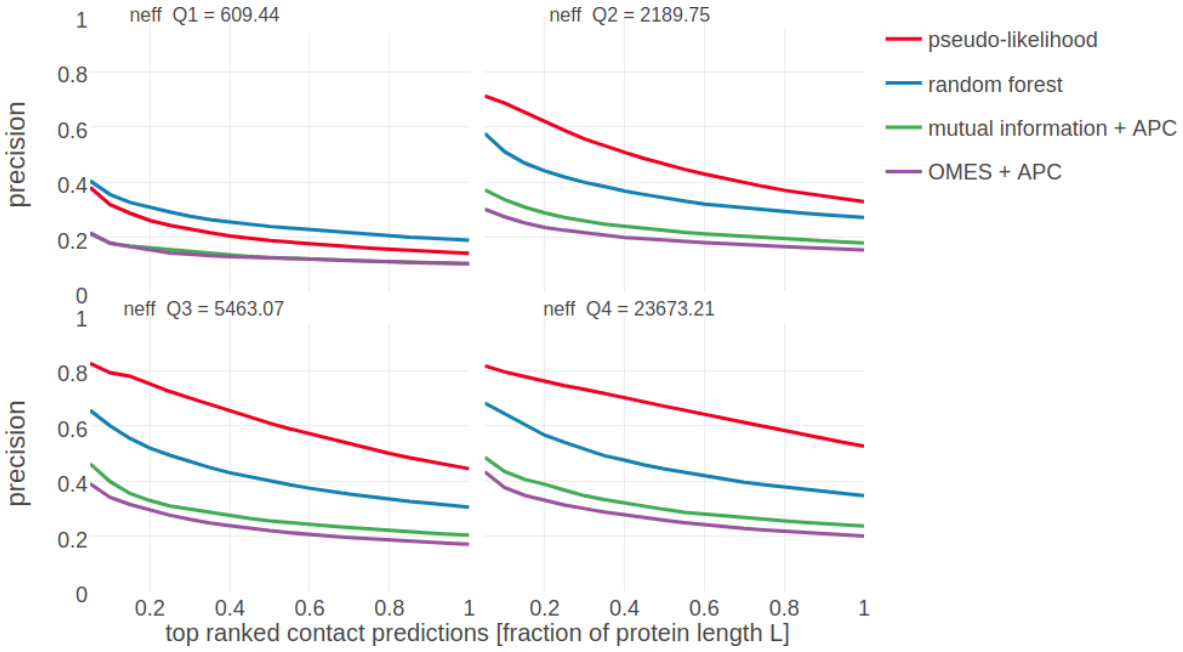

Figure 5.7: Mean precision for top ranked contacts on a test set of 1000 proteins splitted into four equally sized subsets with respect to Neff. Subsets are defined according to quantiles of Neff values. Upper left: Subset of proteins with Neff < Q1. Upper right: Subset of proteins with Q1 <= Neff < Q2. Lower left: Subset of proteins with Q2 <= Neff < Q3. Lower right: Subset of proteins with Q3 <= Neff < Q4. **pseudo-likelihood** = APC corrected Frobenius norm of couplings computed with pseudo-likelihood. **random forest** = random forest model trained on 75 sequence derived features. **OMES** = APC corrected *OMES* contact score according to Fodor&Aldrich [214]. **mutual information** = APC corrected mutual information between amino acid counts (using pseudo-counts).

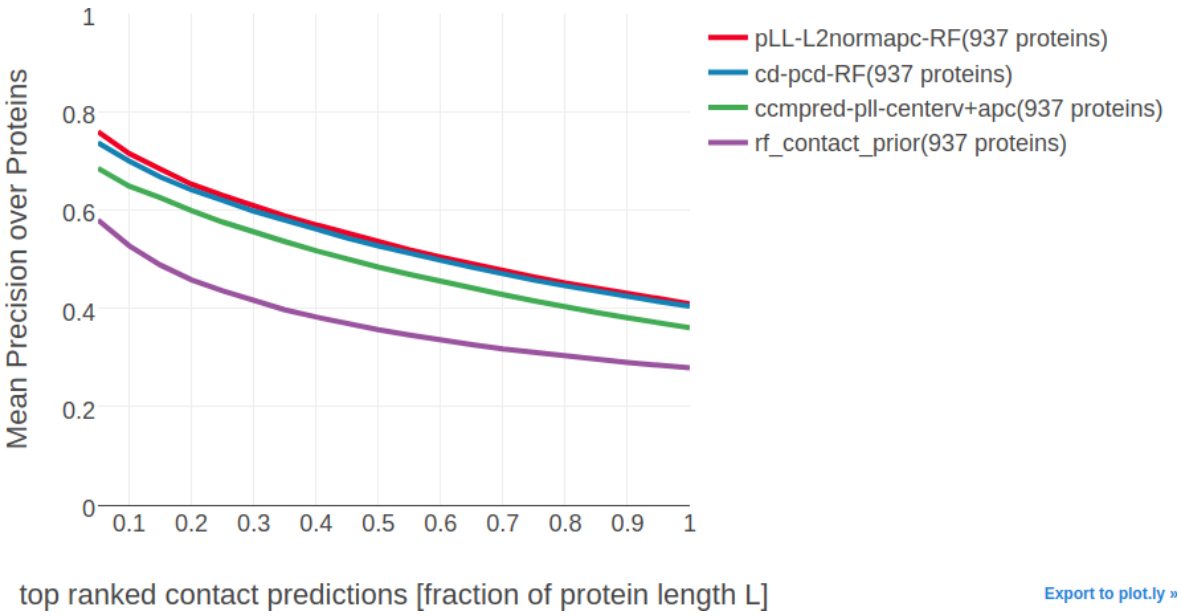

Figure 5.8: Mean precision for top ranked contacts on a test set of 1000 proteins. **random forest pLL** random forest model trained on 75 sequence derived features and the pseudo-likelihood contact score (see method *pseudo-likelihood*). **random forest CD** random forest model trained on 75 sequence derived features and the contrastive divergence contact score (see method *contrastive divergence*). **random forest pLL CD** random forest model trained on 75 sequence derived features and the pseudo-likelihood contact score (see method *pseudo-likelihood*) and the contrastive divergence contact score (see method *contrastive divergence*). **contrastive divergence** APC corrected Frobenius norm of couplings computed with contrastive divergence. **pseudo-likelihood** = APC corrected Frobenius norm of couplings computed with pseudo-likelihood. **random forest** = random forest model trained on 75 sequence derived features.

neglectable.

Similarly, the contact score derived from couplings computed with CD in chapter 4 can be added as a feature instead of the pseudo-likelihood score or besides the pseudo-likelihood contact score. All three models perform comparably, as can be seen in the benchmark in Figure 5.8. The random forest model using both contact scores as features improves only slightly in precision over the random forest model that uses solely one of both contact scores. Since it has been shown in section 4.6 that pseudo-likelihood and contrastive divergence contact scores are highly correlated, resulting in very similar rankings for residue pairs, this result is not surprising. Apparantly, there is minor information gain by adding both contact scores.

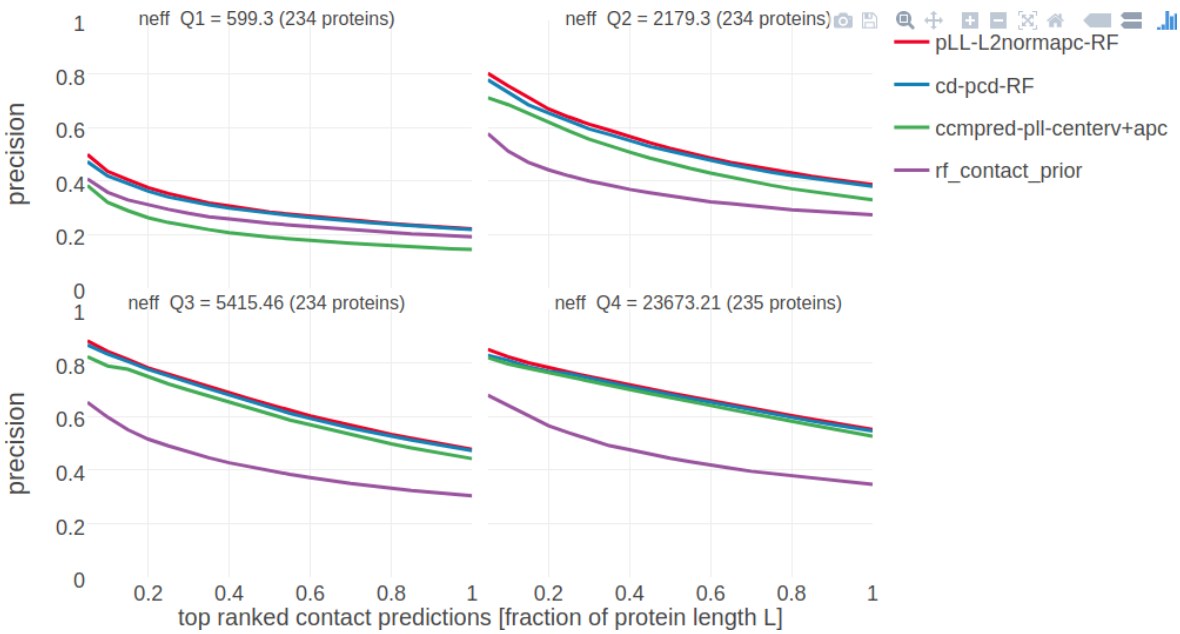

Figure 5.9: Mean precision for top ranked contacts on a test set of 1000 proteins splitted into four equally sized subsets with respect to Neff. Subsets are defined according to quantiles of Neff values. Upper left: Subset of proteins with Neff < Q1. Upper right: Subset of proteins with Q1 <= Neff < Q2. Lower left: Subset of proteins with Q2 <= Neff < Q3. Lower right: Subset of proteins with Q3 <= Neff < Q4. Methods are the same as in Figure 5.8

6

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 2.4.3.1). Apart from the heuristic transformation that omits meaningful information comprised in the coupling matrices $\mathbf{w}_{ij}$ as discussed in section 3, 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 posterior probability distributions of contact states c_{ij} for all residues pairs i and j, given the MSA $\mathbf{X}$. A true contact, having contact state $c=1$, is defined as two residues whose C_β-C_β distance $\leq 8\mathring{A}$, whereas a residue pair with C_β-C_β distance $> 8\mathring{A}$ is considered not to be in physical contact and is assigned the contact state $\mathbf{c}=0$. 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.

6.1 Computing the Posterior Probabiilty of a Contact $p(\mathbf{c}=1|\mathbf{X})$

The joint probability of contact states $\mathbf{c}$ and MRF model parameters $(\mathbf{v},\mathbf{w})$ given the MSA $\mathbf{X}$ and a set of sequence derived features ϕ (such as listed in method section 7.15), can be written as a hierarchical Bayesian model of the form:

$$p(\mathbf{c},\mathbf{v},\mathbf{w}|\mathbf{X},\phi) \propto p(\mathbf{X}|\mathbf{v},\mathbf{w})p(\mathbf{v},\mathbf{w}|\mathbf{c})\,p(\mathbf{c}|\phi)\,. \tag{6.1}$$

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

$$p(\mathbf{c}|\mathbf{X},\phi) \propto p(\mathbf{X}|\mathbf{c})p(\mathbf{c}|\phi) \tag{6.2}$$

$$p(\mathbf{X}|\mathbf{c}) = \int\int p(\mathbf{X}|\mathbf{v},\mathbf{w})\, p(\mathbf{v},\mathbf{w}|\mathbf{c})\, d\mathbf{v}\, d\mathbf{w}\ . \tag{6.3}$$

The single potentials $\mathbf{v}$ will be fixed at their best estimate $\mathbf{v}^*$ (see method section 7.6.3) 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 contact state c_{ij}, which is described in detail in the next section 6.3. The prior over MRF model parameters then yields,

$$p(\mathbf{v},\mathbf{w}|\mathbf{c}) = \mathcal{N}(\mathbf{v}|\mathbf{v}^*, \lambda_v^{-1}\mathbf{I}) \prod_{1\le i<j\le L} p(\mathbf{w}_{ij}|c_{ij})\ . \tag{6.4}$$

Furthermore, section 6.2 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. (6.3) and is covered in section 6.4.

Finally, the marginals $p(c_{ij}|\mathbf{X},\phi) = \int p(\mathbf{c}|\mathbf{X},\phi) d\mathbf{c}_{\setminus ij}$, where $\mathbf{c}_{\setminus ij}$ is the vector containing all coordinates of $\mathbf{c}$ except c_{ij} will be computed in 6.6.

6.2 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

$$p(\mathbf{w}|\mathbf{X},\mathbf{v}^*) = p(\mathbf{X}|\mathbf{v}^*,\mathbf{w})\mathcal{N}(\mathbf{w}|\mathbf{0},\lambda_w^{-1}\mathbf{I}) \tag{6.5}$$

where the single potentials $\mathbf{v}$ are set to the target vector $\mathbf{v}^*$ as discussed in section 6.1.

The posterior distribution can be approximated with a so called "Laplace Approximation" [94] as follows. By performing a second order Taylor expansion around the mode $\mathbf{w}^*$ of the log posterior it can be written as

$$\log p(\mathbf{w}|\mathbf{X},\mathbf{v}^*) \overset{!}{\approx} \log p(\mathbf{w}^*|\mathbf{X},\mathbf{v}^*) + \nabla_{\mathbf{w}} \log p(\mathbf{w}|\mathbf{X},\mathbf{v}^*)|_{\mathbf{w}^*}(\mathbf{w}-\mathbf{w}^*) - \frac{1}{2}(\mathbf{w}-\mathbf{w}^*)^{\mathrm{T}}\mathbf{H}(\mathbf{w}-\mathbf{w}^*)\,. \tag{6.6}$$

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

$$(\mathbf{H})_{klcd,ijab} = -\left.\frac{\partial^2 \log p(\mathbf{w}|\mathbf{X},\mathbf{v}^*)}{\partial \mathbf{w}_{klcd}\,\partial w_{ijab}}\right|_{(\mathbf{w}^*)}\,. \tag{6.7}$$

The mode $\mathbf{w}^*$ will be determined with the CD approach described in detail in section 4. 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

$$\log p(\mathbf{w}|\mathbf{X},\mathbf{v}^*) \approx \log p(\mathbf{w}^*|\mathbf{X},\mathbf{v}^*) - \frac{1}{2}(\mathbf{w}-\mathbf{w}^*)^{\mathrm{T}}\,\mathbf{H}\,(\mathbf{w}-\mathbf{w}^*)\,. \tag{6.8}$$

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

$$\begin{aligned} p(\mathbf{w}|\mathbf{X},\mathbf{v}^*) &\approx p(\mathbf{w}^*|\mathbf{X},\mathbf{v}^*)\exp\left(-\frac{1}{2}(\mathbf{w}-\mathbf{w}^*)^{\mathrm{T}}\mathbf{H}(\mathbf{w}-\mathbf{w}^*)\right) \\ &= p(\mathbf{w}^*|\mathbf{X},\mathbf{v}^*)\frac{(2\pi)^{\frac{D}{2}}}{|\mathbf{H}|^{\frac{D}{2}}} \times \mathcal{N}(\mathbf{w}|\mathbf{w}^*,\mathbf{H}^{-1}) \\ &\propto \mathcal{N}(\mathbf{w}|\mathbf{w}^*,\mathbf{H}^{-1})\,, \end{aligned} \tag{6.9}$$

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 7.11.

6.2.1 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 7.13, 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:

$$\mu(r_{ij}) = \mathrm{E}_{p(\mathbf{w}_{ij}|r_{ij},\tilde{\mu},\tilde{\Lambda})}\left[\mathbf{w}_{ij}\right] \tag{6.10}$$

$$= \int \mathbf{w}_{ij}\, p(\mathbf{w}_{ij}|r_{ij},\tilde{\mu},\tilde{\Lambda}) d\mathbf{w} \tag{6.11}$$

$$= \int \mathbf{w}_{ij} \sum_{k=0}^{K} g_k(r_{ij})\, \mathcal{N}(\mathbf{w}_{ij}|\tilde{\mu_k},\tilde{\Lambda}_k^{-1}) d\mathbf{w} \tag{6.12}$$

$$= \sum_{k=0}^{K} g_k(r_{ij}) \int \mathbf{w}_{ij}\, \mathcal{N}(\mathbf{w}_{ij}|\tilde{\mu_k},\tilde{\Lambda}_k^{-1}) d\mathbf{w} \tag{6.13}$$

$$\mu(r_{ij}) = \sum_{k=0}^{K} g_k(r_{ij})\, \tilde{\mu_k} \tag{6.14}$$

and similarly,

$$\mathbf{\Sigma}(r_{ij}) = \mathrm{var}_{p(\mathbf{w}_{ij}|r_{ij},\tilde{\mu},\tilde{\Lambda})}\left[\mathbf{w}_{ij}\right] \tag{6.15}$$

$$= \int (\mathbf{w}_{ij} - \mu(r_{ij}))(\mathbf{w}_{ij} - \mu(r_{ij}))^{\mathrm{T}}\, p(\mathbf{w}_{ij}|r_{ij},\tilde{\mu},\tilde{\Lambda}) d\mathbf{w} \tag{6.16}$$

$$= \sum_{k=0}^{K} g_k(r_{ij}) \int (\mathbf{w}_{ij} - \mu(r_{ij}))(\mathbf{w}_{ij} - \mu(r_{ij}))^{\mathrm{T}}\, \mathcal{N}(\mathbf{w}_{ij}|\tilde{\mu_k},\tilde{\mathbf{\Lambda}_k}^{-1}) d\mathbf{w} \tag{6.17}$$

$$= \sum_{k=0}^{K} g_k(r_{ij}) \int (\mathbf{w}_{ij} - \mu(r_{ij}) + \tilde{\mu_k})(\mathbf{w}_{ij} - \mu(r_{ij}) + \tilde{\mu_k})^{\mathrm{T}}\, \mathcal{N}(\mathbf{w}_{ij}|\mathbf{0},\tilde{\mathbf{\Lambda}_k}^{-1}) d\mathbf{w} \tag{6.18}$$

$$\mathbf{\Sigma}(r_{ij}) = \sum_{k=0}^{K} g_k(r_{ij}) \left(\tilde{\mathbf{\Lambda}_k}^{-1} + (\mu(r_{ij}) - \tilde{\mu_k})(\mu(r_{ij}) - \tilde{\mu_k})^{\mathrm{T}}\right) . \tag{6.19}$$

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.

6.3 Modelling the prior over couplings with dependence on c_{ij}

The prior over couplings $p(\mathbf{w}_{ij}|c_{ij})$ 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 normalised weights $g_k(c_{ij})$ that depend on the contact state,

$$p(\mathbf{w}_{ij}|c_{ij}) = \sum_{k=0}^{K} g_k(c_{ij})\, \mathcal{N}(\mathbf{w}_{ij}|\mu_k,\mathbf{\Lambda}_k^{-1}) . \tag{6.20}$$

The mixture weights $g_k(c_{ij})$ in eq. (6.20) are modelled as softmax:

$$g_k(c_{ij}) = \frac{\exp \gamma_k(c_{ij})}{\sum_{k'=0}^{K} \exp \gamma_{k'}(c_{ij})} \tag{6.21}$$

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

The assumption that the prior over couplings $p(\mathbf{w}_{ij}|c_{ij})$ can be modelled as a multivariate Gaussian is justified by the analysis of single and 2-dimensional coupling distributions presented in section 3.3 and in section 3.4. The couplings w_{ijab} for this analysis have been filtered such that they have sufficient evidence in the alignment (see method section 7.7.1 for details). Therefore, the presented distributions should resemble the posterior distribution of couplings $p(\mathbf{w}|\mathbf{X}, \mathbf{v}^*) \propto \mathcal{N}(\mathbf{w}|\mathbf{w}^*, \mathbf{H}^{-1})$ in the case that the diagonal elements $(\mathbf{H})_{ijab,ijab}$ have non-negligible values.

6.4 Computing the likelihood function of contact states $p(\mathbf{X}|\mathbf{c})$

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

$$p(\mathbf{X}|\mathbf{c}) = \int \int p(\mathbf{X}|\mathbf{v}, \mathbf{w})\, p(\mathbf{v}, \mathbf{w}|\mathbf{c})\, d\mathbf{v}\, d\mathbf{w}\ . \tag{6.22}$$

Inserting the prior over parameters $p(\mathbf{v}, \mathbf{w}|\mathbf{c})$ from eq. (6.4) into the previous equation and performing the integral over $\mathbf{v}$, as discussed earlier in section 6.1, yields

$$p(\mathbf{X}|\mathbf{c}) = \int \left(\int p(\mathbf{X}|\mathbf{v}, \mathbf{w})\, \mathcal{N}(\mathbf{v}|\mathbf{v}^*, \lambda_v^{-1}\mathbf{I})\, d\mathbf{v} \right) \prod_{1 \le i < j \le L} p(\mathbf{w}_{ij}|c_{ij})\, d\mathbf{w} \tag{6.23}$$

$$p(\mathbf{X}|\mathbf{c}) = \int p(\mathbf{X}|\mathbf{v}^*, \mathbf{w}) \prod_{1 \le i < j \le L} p(\mathbf{w}_{ij}|c_{ij})\, d\mathbf{w} \tag{6.24}$$

Next, the likelihood of sequences, $p(\mathbf{X}|\mathbf{v}^*, \mathbf{w})$, will be multiplied with the regularisation prior $\mathcal{N}(\mathbf{w}|\mathbf{0}, \lambda_w^{-1}\mathbf{I})$ and at the same time the coupling prior, which depends on the contact states, will be divided by the regularisation prior again:

$$p(\mathbf{X}|\mathbf{c}) = \int p(\mathbf{X}|\mathbf{v}^*, \mathbf{w})\, \mathcal{N}(\mathbf{w}|\mathbf{0}, \lambda_w^{-1}\mathbf{I}) \prod_{1 \le i < j \le L} \frac{p(\mathbf{w}_{ij}|c_{ij})}{\mathcal{N}(\mathbf{w}_{ij}|\mathbf{0}, \lambda_w^{-1}\mathbf{I})}\, d\mathbf{w}\ . \tag{6.25}$$

Now the crucial advantage of the likelihood regularisation is borne out: the strength of the regularisation prior, λ_w, can be chosen 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 coupling prior $\prod_{1 \le i < j \le L} p(\mathbf{w}_{ij}|c_{ij})$ that depends on the contact state. This allows to approximate the regularised likelihood with a Gaussian distribution (eq. (6.9)), 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:

$$p(\mathbf{X}|\mathbf{c}) \propto \int \mathcal{N}(\mathbf{w}|\mathbf{w}^*, \mathbf{H}^{-1}) \prod_{1\leq i<j\leq L} \frac{p(\mathbf{w}_{ij}|c_{ij})}{\mathcal{N}(\mathbf{w}_{ij}|\mathbf{0}, \lambda_w^{-1}\mathbf{I})} d\mathbf{w}\,. \tag{6.26}$$

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 7.10 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. (6.26), it is therefore a good approximation to simply set the off-diagonal block matrices (case 3 in eq. (7.43)) to zero! The first term in the integrand of eq. (6.26) now factorizes over (i,j),

$$\mathcal{N}(\mathbf{w}|\mathbf{w}^*, \mathbf{H}^{-1}) \approx \prod_{1\leq i<j\leq L} \mathcal{N}(\mathbf{w}_{ij}|\mathbf{w}_{ij}^*, \mathbf{H}_{ij}^{-1}), \tag{6.27}$$

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

$$p(\mathbf{X}|\mathbf{c}) \propto \int \prod_{1\leq i<j\leq L} \mathcal{N}(\mathbf{w}_{ij}|\mathbf{w}_{ij}^*, \mathbf{H}_{ij}^{-1}) \prod_{1\leq i<j\leq L} \frac{p(\mathbf{w}_{ij}|c_{ij})}{\mathcal{N}(\mathbf{w}_{ij}|\mathbf{0}, \lambda_w^{-1}\mathbf{I})} d\mathbf{w} \tag{6.28}$$

$$p(\mathbf{X}|\mathbf{c}) \propto \int \prod_{1\leq i<j\leq L} \left(\mathcal{N}(\mathbf{w}_{ij}|\mathbf{w}_{ij}^*, \mathbf{H}_{ij}^{-1}) \frac{p(\mathbf{w}_{ij}|c_{ij})}{\mathcal{N}(\mathbf{w}_{ij}|\mathbf{0}, \lambda_w^{-1}\mathbf{I})} \right) d\mathbf{w} \tag{6.29}$$

$$p(\mathbf{X}|\mathbf{c}) \propto \prod_{1\leq i<j\leq L} \int \mathcal{N}(\mathbf{w}_{ij}|\mathbf{w}_{ij}^*, \mathbf{H}_{ij}^{-1}) \frac{p(\mathbf{w}_{ij}|c_{ij})}{\mathcal{N}(\mathbf{w}_{ij}|\mathbf{0}, \lambda_w^{-1}\mathbf{I})} d\mathbf{w}_{ij} \tag{6.30}$$

Inserting the coupling prior defined in eq. (6.20) yields

$$p(\mathbf{X}|\mathbf{c}) \propto \prod_{1\leq i<j\leq L} \int \mathcal{N}(\mathbf{w}_{ij}|\mathbf{w}_{ij}^*, \mathbf{H}_{ij}^{-1}) \frac{\sum_{k=0}^{K} g_k(c_{ij}) \mathcal{N}(\mathbf{w}_{ij}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{w}_{ij}|\mathbf{0}, \lambda_w^{-1}\mathbf{I})} d\mathbf{w}_{ij} \tag{6.31}$$

$$p(\mathbf{X}|\mathbf{c}) \propto \prod_{1\leq i<j\leq L} \sum_{k=0}^{K} g_k(c_{ij}) \int \frac{\mathcal{N}(\mathbf{w}_{ij}|\mathbf{w}_{ij}^*, \mathbf{H}_{ij}^{-1})}{\mathcal{N}(\mathbf{w}_{ij}|\mathbf{0}, \lambda_w^{-1}\mathbf{I})} \mathcal{N}(\mathbf{w}_{ij}|\mu_k, \mathbf{\Lambda}_k^{-1}) d\mathbf{w}_{ij}\,. \tag{6.32}$$

The integral can be carried out using the following formula:

$$\int d\mathbf{x} \frac{\mathcal{N}(\mathbf{x}|\mu_1, \mathbf{\Lambda}_1^{-1})}{\mathcal{N}(\mathbf{x}|\mathbf{0}, \mathbf{\Lambda}3^{-1})} \mathcal{N}(\mathbf{x}|\mu_2, \mathbf{\Lambda}_2^{-1}) = \frac{\mathcal{N}(\mathbf{0}|\mu_1, \mathbf{\Lambda}_1^{-1}) \mathcal{N}(\mathbf{0}|\mu_2, \mathbf{\Lambda}_2^{-1})}{\mathcal{N}(\mathbf{0}|\mathbf{0}, \mathbf{\Lambda}_3^{-1}) \mathcal{N}(\mathbf{0}|\mu_{12}, \mathbf{\Lambda}_{123}^{-1})} \tag{6.33}$$

with

$$\mathbf{\Lambda}_{123} := \mathbf{\Lambda}_1 - \mathbf{\Lambda}_3 + \mathbf{\Lambda}_2 \tag{6.34}$$

$$\mu_{12} := \mathbf{\Lambda}_{123}^{-1}(\mathbf{\Lambda}_1\mu_1 + \mathbf{\Lambda}_2\mu_2). \tag{6.35}$$

We define

$$\mathbf{\Lambda}_{ij,k} := \mathbf{H}_{ij} - \lambda_w \mathbf{I} + \mathbf{\Lambda}_k \tag{6.36}$$

$$\mu_{ij,k} := \mathbf{\Lambda}_{ij,k}^{-1}(\mathbf{H}_{ij}\mathbf{w}_{ij}^* + \mathbf{\Lambda}_k \mu_k)\,. \tag{6.37}$$

and obtain

$$p(\mathbf{X}|\mathbf{c}) \propto \prod_{1 \le i < j \le L} \sum_{k=0}^{K} g_k(c_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})}\,. \tag{6.38}$$

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

6.5 Training the Hyperparameters in the Likelihood Function of Contact States

The likelihood function of contact states given in eq. (6.38) contains the hyperparameters $\gamma_k(c_{ij})$, μ_k and $\mathbf{\Lambda}_k$ representing the weights, mean vectors and precision matrices of the K Gaussian components of the contact state dependent coupling prior, respectively. The hyperparameters will be trained by maximizing the logarithm of the likelihood over a set of training MSAs $\mathbf{X}^1, \ldots, \mathbf{X}^M$ and associated structures with contact state vectors $\mathbf{c}^1, \ldots, \mathbf{c}^M$ plus a regularizer $R(\mu, \mathbf{\Lambda})$ as described in detail in method section 7.13.

I training models with $K \in \{1, 3, 5\}$ Gaussian mixture components with diagonal precision matrices $\mathbf{\Lambda}_k$ and a zero-component that is fixed at $\mu_0 = 0$. The number of model parameters assembles as follows:

- $(K-1) \times 400$ parameters for μ_k with $k \in \{1, \ldots, K\}$ ($\mu_0 = 0$)
- $K \times 400$ parameters for the diagonal $(\mathbf{\Lambda}_k)_{ab,ab}$ with $k \in \{0, \ldots, K\}$
- $2 \times (K-1)$ parameters for $\gamma_k(c_{ij})$ for $k \in \{1, 2\}$ and $c_{ij} \in \{0, 1\}$ ($\gamma_0(c_{ij}) = 1$).

This yields 2004 parameters for $K = 3$, 3608 parameters for $K = 5$ and 7618 parameters for $K = 10$ components. The models have been trained on datasets of varying sizes, consisting of 10000, 100000, 300000 and 500000 residue pairs for contacts and non-contacts respectively.

Convergence according to the criterion, ... has only been observed for the smallest datasets with 10000 and 100000 residue pairs per contact class. CONVERGENCE: REL_REDUCTION_OF_F_<=_FACTR*EPSMCH

6.6 The posterior probability distribution for contact states c_{ij}

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

$$
\begin{aligned}
p(c_{ij}|\mathbf{X},\phi) &= \int d\mathbf{c}_{\setminus ij}\, p(\mathbf{c}|\mathbf{X},\phi) \\
&\propto \int d\mathbf{c}_{\setminus ij}\, p(\mathbf{X}|\mathbf{c})\, p(\mathbf{c}|\phi) \\
&\propto \int d\mathbf{c}_{\setminus ij} \prod_{i'<j'} \sum_{k=0}^{K} g_k(c_{i'j'})\, \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} \prod_{i'<j'} p(c_{i'j'}|\phi_{i'j'})\,, \qquad (6.39)
\end{aligned}
$$

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

$$
\begin{aligned}
p(c_{ij}|\mathbf{X},\phi) &\propto p(c_{ij}|\phi_{ij}) \sum_{k=0}^{K} g_k(c_{ij})\, \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} \\
&\times \prod_{i'<j',(i',j')\neq(i,j)} \int dc_{i'j'}\, p(c_{i'j'}|\phi_{i'j'}) \sum_{k=0}^{K} g_k(c_{i'j'})\, \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} \qquad (6.40)
\end{aligned}
$$

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

$$
p(c_{ij}|\mathbf{X},\phi) \propto p(c_{ij}|\phi_{ij}) \sum_{k=0}^{K} g_k(c_{ij})\, \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})}\,. \qquad (6.41)
$$

7 Methods

7.1 Dataset

A protein dataset has been constructed from the CATH (v4.1) [218] 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[179] 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 [179] 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 ($30 \leq L \leq 600$)
- less than 80% gaps in the multiple sequence alignment (percent gaps < 0.8)
- at least one residue-pair in contact at $C_\beta < 8\mathring{A}$ 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 7.1).

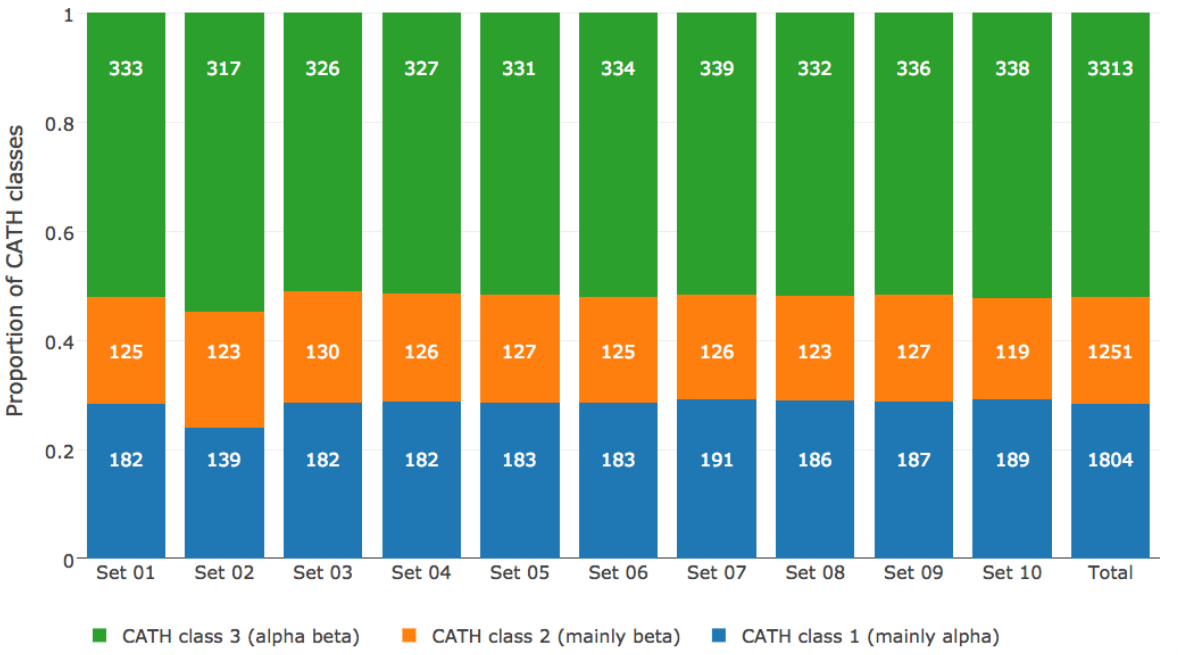

Figure 7.1: Distribution of CATH classes (1=mainly α, 2=mainly β, 3=$\alpha-\beta$) in the dataset and the ten subsets.

7.2 Computing Pseudo-Likelihood Couplings

Dr Stefan Seemayer has reimplementated the open-source software CCMpred [100] in Python. CCMpred optimizes the regularized negative pseudo-log-likelihood using a conjugate gradients optimizer. 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.

7.2.1 Differences between CCMpred and CCMpredpy

CCMpredPy differs from CCMpred [100] 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 7.6.3) using amino acid frequencies computed as described in section 7.4. 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 7.6.3) using amino acid frequencies computed

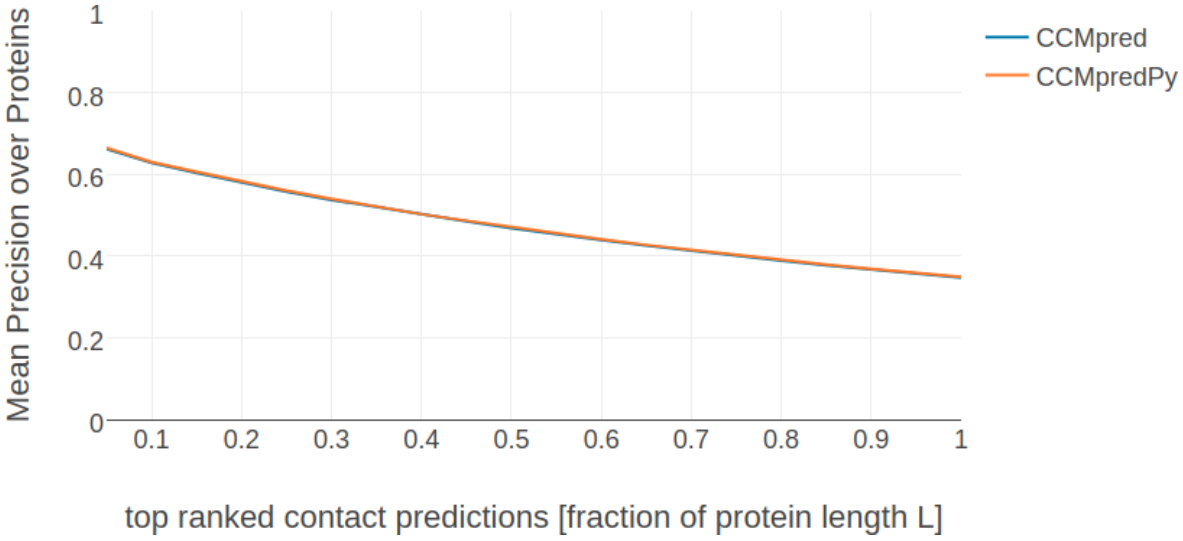

Figure 7.2: Mean precision over 3124 proteins of top ranked contacts computed as APC corrected Frobenius norm of couplings. Couplings have been computed with CCMpred [100] and CCMpredPy as specified in the legend. Specific flags that have been used to run both methods are described in detail in the text (see section 7.2.1).

as described in section 7.4. 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 7.2).

The benchmark in Figure 7.2 as well as all contacts predicted with CCMpred and CCMPredPy (using pseudo-likelihood) in my thesis have been computed using the following flags:

flags for computing pseudo-likelihood couplings with CCMpredPy:

```
--maxit 250                    # Compute a maximum of MAXIT operations
--center-v                     # Use a Gaussian prior for single potentials
                               # centered at ML estimate v*
--reg-l2-lambda-single 10      # regularization coefficient for
                               # single potentials
--reg-l2-lambda-pair-factor 0.2   # regularization coefficient for
                                  # pairwise potentials computed as
                                  # reg-l2-lambda-pair-factor * (L-1)
--pc-uniform          # use uniform pseudocounts
                      # (1/21 for 20 amino acids + 1 gap state)
--pc-count 1          # defining pseudo count admixture coefficient
                      # rho = pc-count/( pc-count+ Neff)
--epsilon 1e-5        # convergence criterion for minimum decrease
                      # in the last K iterations
--ofn-pll             # using pseudo-likelihood as objective function
--alg-cg              # using conjugate gradient to optimize
                      # objective function
```

flags for computing pseudo-likelihood couplings 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 K iterations to EPSILON
```

7.3 Sequence Reweighting

As discussed in section 2.7, 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 [99]. It has been found that reweighting improves contact prediction performance [65,95,176] significantly but results are robust against the choice of the identity threshold in a range between 0.7 and 0.9 [95]. An identity threshold of 0.8 has been used for all analyses in this thesis.

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:

$$w_n = \frac{1}{m_n}, m_n = \sum_{m=1}^{N} I\left(ID(x_n, x_m) \geq 0.8\right) ID(x_n, x_m) = \frac{1}{L} \sum_{i=1}^{L} I(x_n^i = x_m^i) \tag{7.1}$$

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

$$N_{\text{eff}} = \sum_{n=1}^{N} w_n \tag{7.2}$$

TODO: Plot Performance for Seq weighting

7.4 Computing Amino Acid Frequencies

Single and pairwise amino acid frequencies are computed from amino acid counts of weighted sequences as described in the last section 7.3 and additional pseudocounts that are added to improve numerical stability.

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

$$q(x_i{=}a) := (1 - \tau)\ q_0(x_i{=}a) + \tau \tilde{q}(x_i{=}a) \tag{7.3}$$

$$q(x_i{=}a, x_j{=}b) := (1 - \tau)^2\ [q_0(x_i{=}a, x_j{=}b) - q_0(x_i{=}a) q_0(x_j{=}b)] + q(x_i{=}a)\ q(x_j{=}b) \tag{7.4}$$

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:

Figure 7.3: Number of contacts ($C_\beta < 8\mathring{A}$) with respect to protein length and sequence separation has a linear relationship.

$$\tau = \frac{N_{\text{pc}}}{(N_{\text{eff}} + N_{\text{pc}})} \tag{7.5}$$

where $N_{pc} > 0$.

The formula for $q(x_i = a, x_j = b)$ in the second line in eq (7.4) 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 $q_0(x_i = a, x_j = b) = q_0(x_i = a)q_0(x_j = b)$.

7.5 Regularization

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

$$\begin{aligned} R(\mathbf{v}, \mathbf{w}) &= \log \left[\mathcal{N}(\mathbf{v}|\mathbf{v}^*, \lambda_v^{-1} I) \mathcal{N}(\mathbf{w}|\mathbf{w}^*, \lambda_w^{-1} I) \right] \\ &= -\frac{\lambda_v}{2} ||\mathbf{v} - \mathbf{v}^*||_2^2 - \frac{\lambda_w}{2} ||\mathbf{w} - w^*||_2^2 + \text{const.} , \end{aligned} \tag{7.6}$$

where the regularization coefficients λ_v and λ_w determine the strength of regularization.

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 7.3.

Most previous pseudo-likelihood approaches using L2-regularization for pseudo-likelihood optimization set $\mathbf{v}^* = \mathbf{w}^* = \mathbf{0}$ [100–102]. A different choice for v^* is discussed in section 7.6.3 that is is used per default with *CCMpredPy*. The single potentials will not be optimized with CD but will be fixed at v^* given in eq. (7.28). Furthermore, *CCMpredPy* uses regularization coefficients $\lambda_v = 10$ and $\lambda_w = 0.2 \cdot (L - 1)$ for pseudo-likelihood optimization and the choice for λ_w used with CD is discussed in section 4.3.

7.6 The Potts Model

The N sequences of the MSA $\mathbf{X}$ of a protein family 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. The likelihood of the sequences in the MSA of the protein family is modelled with a *Potts Model*, as described in detail in section 2.4:

$$\begin{aligned} p(\mathbf{X}|\mathbf{v}, \mathbf{w}) &= \prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{v}, \mathbf{w}) \\ &= \prod_{n=1}^{N} \frac{1}{Z(\mathbf{v}, \mathbf{w})} \exp\left(\sum_{i=1}^{L} v_i(x_{ni}) \sum_{1 \le i < j \le L} w_{ij}(x_{ni}, x_{nj}) \right) \end{aligned} \tag{7.7}$$

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 $p(\mathbf{x}_n|\mathbf{v}, \mathbf{w})$:

$$Z(\mathbf{v}, \mathbf{w}) = \sum_{y_1, ..., y_L = 1}^{20} \exp\left(\sum_{i=1}^{L} v_i(y_i) \sum_{1 \le i < j \le L} w_{ij}(y_i, y_j) \right) \tag{7.8}$$

The log likelihood is

$$\begin{aligned} LL(\mathbf{v}, \mathbf{w}) &= \log p(\mathbf{X}|\mathbf{v}, \mathbf{w}) \\ &= \sum_{n=1}^{N} \left[\sum_{i=1}^{L} v_i(x_{ni}) \sum_{1 \le i < j \le L} w_{ij}(x_{ni}, x_{nj}) \right] - N \log Z(\mathbf{v}, \mathbf{w}). \end{aligned} \tag{7.9}$$

The gradient of the log likelihood has single components

$$\begin{aligned} \frac{\partial LL(\mathbf{v}, \mathbf{w})}{\partial v_{ia}} &= \sum_{n=1}^{N} I(x_{ni} = a) - N \frac{\partial}{\partial v_{ia}} \log Z(\mathbf{v}, \mathbf{w}) \\ &= \sum_{n=1}^{N} I(x_{ni} = a) - N \sum_{y_1, ..., y_L = 1}^{20} \frac{\exp\left(\sum_{i=1}^{L} v_i(y_i) + \sum_{1 \le i < j \le L} w_{ij}(y_i, y_j) \right)}{Z(\mathbf{v}, \mathbf{w})} I(y_i = a) \\ &= N q(x_i = a) - N p(x_i = a|\mathbf{v}, \mathbf{w}) \end{aligned} \tag{7.10}$$

and pair components

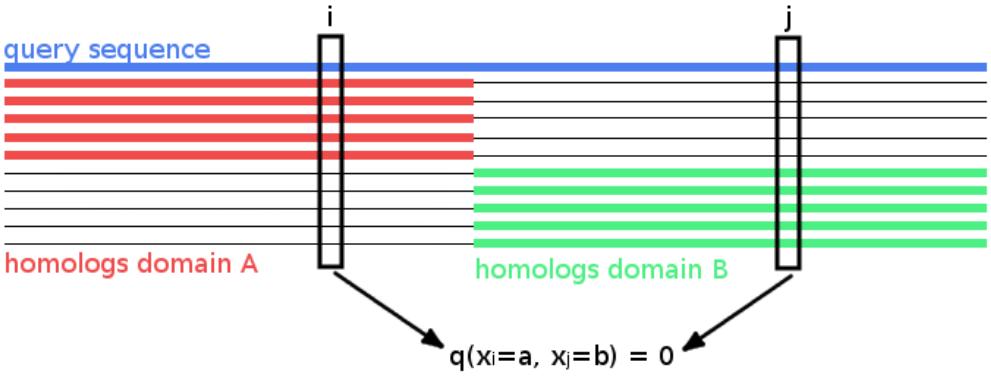

Figure 7.4: Hypothetical MSA consisting of two sets of sequences: the first set has sequences covering only the left half of columns, while the second set has sequences covering only the right half of columns. The two blocks could correspond to protein domains that were aligned to a single query sequence. 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.

$$
\begin{aligned}
\frac{\partial LL(\mathbf{v},\mathbf{w})}{\partial w_{ijab}} &= \sum_{n=1}^{N} I(x_{ni}=a, x_{nj}=b) - N\frac{\partial}{\partial w_{ijab}} \log Z(\mathbf{v},\mathbf{w}) \\
&= \sum_{n=1}^{N} I(x_{ni}=a, x_{nj}=b) \\
&\quad - N \sum_{y_1,\ldots,y_L=1}^{20} \frac{\exp\left(\sum_{i=1}^{L} v_i(y_i) + \sum_{1\le i<j\le L} w_{ij}(y_i,y_j)\right)}{Z(\mathbf{v},\mathbf{w})} I(y_i=a, y_j=b) \\
&= N q(x_i=a, x_j=b) - N \sum_{y_1,\ldots,y_L=1}^{20} p(y_1,\ldots,y_L|\mathbf{v},\mathbf{w})\, I(y_i=a, y_j=b) \\
&= N q(x_i=a, x_j=b) - N p(x_i=a, x_j=b|\mathbf{v},\mathbf{w})
\end{aligned} \tag{7.11}
$$

7.6.1 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 7.4. 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 $a, b \in \{1, ..., L\}$.

According to the gradient of the log likelihood for couplings w_{ijab} given in eq (7.11), 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 at 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:

$$S_n := \{(y_1,...,y_L) : 0 \leq y_i \leq 20 \wedge (y_i{=}0 \text{ iff } x_{ni}{=}0)\} \tag{7.12}$$

and the partition function becomes:

$$Z_n(\mathbf{v},\mathbf{w}) = \sum_{\mathbf{y}\in S_n} \exp\left(\sum_{i=1}^{L} v_i(y_i) \sum_{1\leq i<j\leq L} w_{ij}(y_i,y_j)\right) \tag{7.13}$$

To ensure that the gaps in $y \in S_n$ 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 $\{1,...,20\}$,

$$N_i := \sum_{n=1}^{N} w_n I(x_{ni}{\neq}0) \qquad q_{ia} = q(x_i{=}a) := \frac{1}{N_i}\sum_{n=1}^{N} w_n I(x_{ni}{=}a) \tag{7.14}$$

$$N_{ij} := \sum_{n=1}^{N} w_n I(x_{ni}{\neq}0, x_{nj}{\neq}0) \quad q_{ijab} = q(x_i{=}a, x_j{=}b) := \frac{1}{N_{ij}}\sum_{n=1}^{N} w_n I(x_{ni}{=}a, x_{nj}{=}b) \tag{7.15}$$

with w_n being sequence weights calculated as described in methods section 7.3. With this definition, empirical amino acid frequencies are normalized without gaps, so that

$$\sum_{a=1}^{20} q_{ia} = 1\ ,\ \sum_{a,b=1}^{20} q_{ijab} = 1. \tag{7.16}$$

7.6.2 The Regularized Full Log Likelihood and its Gradient With Gap Treatment

In pseudo-likelihood based methods, a regularisation is commonly used that can be interpreted to arise from a prior probability. The same treatment will be applied to the full likelihood. 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. The choice of v^* is discussed in section 7.6.3. By including the logarithm of this prior into the log likelihood the regularised log likelihood is obtained,

$$LL_{\text{reg}}(\mathbf{v},\mathbf{w}) = \log\left[p(\mathbf{X}|\mathbf{v},\mathbf{w})\,\mathcal{N}(\mathbf{v}|\mathbf{v}^*,\lambda_v^{-1}\mathbf{I})\,\mathcal{N}(\mathbf{w}|\mathbf{0},\lambda_w^{-1}\mathbf{I})\right] \tag{7.17}$$

or explicitely,

$$\begin{aligned} LL_{\text{reg}}(\mathbf{v},\mathbf{w}) = &\sum_{n=1}^{N}\left[\sum_{i=1}^{L} v_i(x_{ni}) + \sum_{1\le i<j\le L} w_{ij}(x_{ni},x_{nj}) - \log Z_n(\mathbf{v},\mathbf{w})\right] \\ &- \frac{\lambda_v}{2}\sum_{i=1}^{L}\sum_{a=1}^{20}(v_{ia}-v_{ia}^*)^2 - \frac{\lambda_w}{2}\sum_{1\le i<j\le L}\sum_{a,b=1}^{20} w_{ijab}^2. \end{aligned} \tag{7.18}$$

The gradient of the regularized log likelihood has single components

$$\begin{aligned} \frac{\partial LL_{\text{reg}}}{\partial v_{ia}} &= \sum_{n=1}^{N} I(x_{ni}=a) - \sum_{n=1}^{N}\frac{\partial}{\partial v_{ia}}\log Z_n(\mathbf{v},\mathbf{w}) - \lambda_v(v_{ia}-v_{ia}^*) \\ &= N_i q(x_i=a) \\ &\quad - \sum_{n=1}^{N}\sum_{\mathbf{y}\in S_n}\frac{\exp\left(\sum_{i=1}^{L} v_i(y_i) + \sum_{1\le i<j\le L} w_{ij}(y_i,y_j)\right)}{Z_n(\mathbf{v},\mathbf{w})} I(y_i=a) \\ &\quad - \lambda_v(v_{ia}-v_{ia}^*) \end{aligned} \tag{7.19}$$

and pair components

$$\begin{aligned} \frac{\partial LL_{\text{reg}}}{\partial w_{ijab}} &= \sum_{n=1}^{N} I(x_{ni}=a, x_{nj}=b) - \sum_{n=1}^{N}\frac{\partial}{\partial w_{ijab}}\log Z_n(\mathbf{v},\mathbf{w}) - \lambda_w w_{ijab} \\ &= N_{ij} q(x_i=a, x_j=b) \\ &\quad - \sum_{n=1}^{N}\sum_{\mathbf{y}\in S_n}\frac{\exp\left(\sum_{i=1}^{L} v_i(y_i) + \sum_{1\le i<j\le L} w_{ij}(y_i,y_j)\right)}{Z_n(\mathbf{v},\mathbf{w})} I(y_i=a, y_j=b) \\ &\quad - \lambda_w w_{ijab} \end{aligned} \tag{7.20}$$

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.

If the proportion of gap positions in $\mathbf{X}$ is small (e.g. $< 5\%$, also compare percentage of gaps in dataset in Appendix Figure C.2), the sums over $\mathbf{y} \in S_n$ in eqs. (7.19) and (7.20) 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

$$\frac{\partial LL_{\text{reg}}}{\partial v_{ia}} = N_i q(x_i=a) - N_i\, p(x_i=a|\mathbf{v},\mathbf{w}) - \lambda_v(v_{ia}-v_{ia}^*) \tag{7.21}$$

$$\frac{\partial LL_{\text{reg}}}{\partial w_{ijab}} = N_{ij} q(x_i=a, x_j=b) - N_{ij}\, p(x_i=a, x_j=b|\mathbf{v},\mathbf{w}) - \lambda_w w_{ijab} \tag{7.22}$$

Note that the couplings between columns i and j in the hypothetical MSA presented in the last section 7.6.1 will now vanish since $N_{ij}=0$ and the gradient with respect to w_{ijab} is equal to $-\lambda_w w_{ijab}$.

7.6.3 The prior on v

Most previous approaches chose a prior around the origin, $p(\mathbf{v}) = \mathcal{N}(\mathbf{v}|\mathbf{0}, \lambda_v^{-1}\mathbf{I})$, i.e., $v^*_{ia}{=}0$. It can be shown that the choice $v^*_{ia} = 0$ leads to undesirable results. Taking the sum over $b = 1, \ldots, 20$ at the optimum of the gradient of couplings in eq. (7.22), yields

$$0 = N_{ij}\, q(x_i{=}a, x_j \neq 0) - N_{ij}\, p(x_i{=}a|\mathbf{v}, \mathbf{w}) - \lambda_w \sum_{b=1}^{20} w_{ijab}\ , \tag{7.23}$$

for all $i, j \in \{1, \ldots, L\}$ and all $a \in \{1, \ldots, 20\}$.

Note, that by taking the sum over $a = 1, \ldots, 20$ it follows that,

$$\sum_{a,b=1}^{20} w_{ijab} = 0. \tag{7.24}$$

At the optimum the gradient with respect to v_{ia} vanishes and according to eq. (7.21), $p(x_i = a|\mathbf{v}, \mathbf{w}) = q(x_i = a) - \lambda_v(v_{ia} - v^*_{ia})/N_i$. This term can be substituted into equation (7.23), yielding

$$0 = N_{ij}\, q(x_i{=}a, x_j \neq 0) - N_{ij}\, q(x_i = a) + \frac{N_{ij}}{N_i}\lambda_v(v_{ia} - v^*_{ia}) - \lambda_w \sum_{b=1}^{20} w_{ijab}\ . \tag{7.25}$$

Considering a MSA without gaps, the terms $N_{ij}\, q(x_i{=}a, x_j \neq 0) - N_{ij}\, q(x_i = a)$ cancel out, leaving

$$0 = \lambda_v(v_{ia} - v^*_{ia}) - \lambda_w \sum_{b=1}^{20} w_{ijab}. \tag{7.26}$$

Now, 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. (7.26), 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

$$\frac{\exp(v^*_{ia})}{\sum_{a'=1}^{20} \exp(v^*_{ia'})} = q(x_i = a). \tag{7.27}$$

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

$$v^*_{ia} = \log q(x_i{=}a) - \frac{1}{20}\sum_{a'=1}^{20} \log q(x_i{=}a'). \tag{7.28}$$

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. (7.26) will be close to zero, as it should be.

7.7 Analysis of Coupling Matrices

7.7.1 Correlation of Couplings with Contact Class

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

- sequence separation of residue pairs ≥ 10
- diversity ($= \frac{\sqrt{N}}{L}$) of alignment ≥ 0.3
- number of non-gapped sequences ≥ 1000
- C_β distance threshold for contact: $< 8\mathring{A}$
- C_β distance threshold for noncontact: $> 25\mathring{A}$

7.7.2 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:

- sequence separation of residue pairs ≥ 10
- percentage of gaps per column $\leq 30\%$
- evidence for a coupling w_{ijab} estimated from the alignment, $N_{ij} \cdot q_i(a) \cdot q_j(b) \geq 100$ with:
  - N_{ij}: number of sequences with no gaps at positions i or j
  - $q_i(a)$, $q_j(b)$: frequencies of amino acids a and b at positions i and j, respectively (computed as described in section 7.4)

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 enough statistical power due to insufficient 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 $N_{ij} \cdot q_i(a) \cdot q_j(b) > 80$.

7.8 Optimizing Contrastive Divergence with Stochastic Gradient Descent

This section describes hyperparameter tuning for the stochastic gradient descent optimization of CD.

The couplings w_{ijab} are initialized at 0 and single potentials v_i will not be optimized but rather kept fixed at their maximum-likleihood estimate v_i^* as described in methods section 7.6.3. The optimization is stopped when the maximum number of 5000 iterations has been reached or when the relative change over the L2-norm of parameter estimates $||\mathbf{w}||_2$ over the last five iterations falls below the threshold of $\epsilon = 1e-8$. The gradient of the full likelihood is approximated with CD which involves Gibbs sampling of protein sequences according to the current model parametrization and is described in detail in methods section 4.4. Zero centered L2-regularization is used to constrain the coupling parameters $\mathbf{w}$ using the regularization coefficient $\lambda_w = 0.2L$ which is the default setting for optimizing the pseudo-likelihood with

CCMpredPy. Performance will be evaluated by the mean precision of top ranked contact predictions over a benchmark set of 300 proteins, that is a subset of the data set described in methods section 7.1. Contact scores for couplings are computed as the APC corrected Frobenius norm as explained in section 2.4.4. Pseudo-likelihood couplings are computed with the tool *CCMpredPy* that is introduced in methods section 7.2.1 and the pseudo-likelihood contact score will serve as general reference method for tuning the hyperparameters.

7.8.1 Tuning Hyperparameters of *ADAM* Optimizer

ADAM [205] stores an exponentially decaying average of past gradients and squared gradients,

$$m_t = \beta_1 m_{t1} + (1\beta_1) g \tag{7.29}$$

$$v_t = \beta_2 v_{t1} + (1\beta_2) g^2 \ , \tag{7.30}$$

with $g = \nabla_w LL_{\text{reg}}(\mathbf{v}, \mathbf{w})$ and the rate of decay being determined by hyperparameters β_1 and β_2. Both terms m_t and v_t represent estimates of the first and second moments of the gradient, respectively. The following bias correction terms compensates for the fact that the vectors m_t and v_t are both initialized at zero and therefore are biased towards zero especially at the beginning of optimization,

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{7.31}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \ . \tag{7.32}$$

Parameters are then updated using step size α, a small noise term ϵ and the corrected moment estimates $\hat{m}_t$, $\hat{v}_t$, according to

$$x_{t+1} = x_t - \alpha \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{7.33}$$

Kingma et al. proposed the default values $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e8$ and a constant learning rate $\alpha = 1e - 3$.

For the two protein chains 1mkc_A_00 and 1c75_A_00, having 142 (Neff=96) and 28078 (Neff=16808) aligned sequences respectively, I analysed the convergence for SGD with different learning rates α (see Figure 7.5). In contrast to plain stochastic gradient descent, with *ADAM* it is possible to use larger learning rates for proteins having big alignments, because the learning rate will be adapted to the magnitude of the gradient for every parameter individually. For protein 1mkc_A_00 having a small alignment, a learning rate of 5e-3 quickly leads to convergence whereas for protein 1c75_A_00 a larger learning rate can be chosen to obtain quick convergence. As a consequence, I defined the learning rate α as a function of Neff,

$$\alpha = 2\text{e}{-3} \log(N_{\text{eff}}) \ , \tag{7.34}$$

such that it will take values ˜5e-3 for proteins with small alignments and values ˜1e-2 for proteins with large alignments.

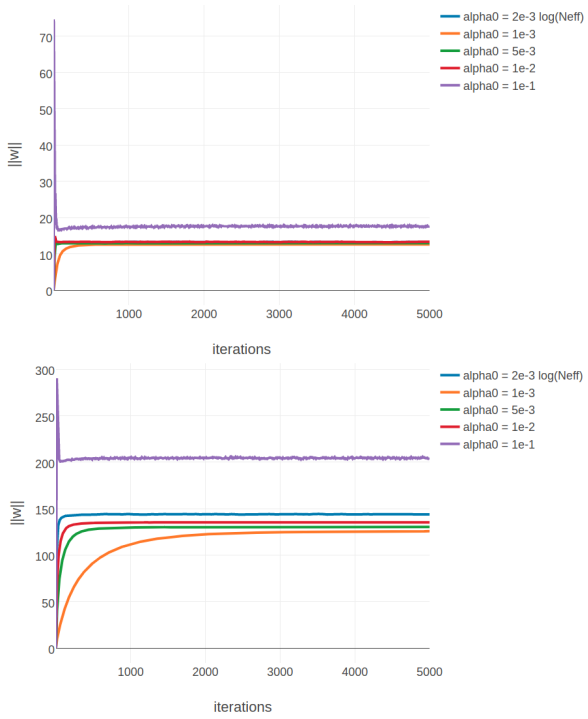

Figure 7.5: L2-norm of the coupling parameters $||\mathbf{w}||_2$ during optimization with *ADAM* and different learning rates without annealing. The learning rate α is specified in the legend. **Left** Convergence plot for protein 1mkc_A_00 having protein length L=43 and 142 sequences in the alignment (Neff=96). **Right** Convergence plot for protein 1c75_A_00 having protein length L=71 and 28078 sequences in the alignment (Neff=16808).

It is interesting to note in Figure 7.5, that the norm of the coupling parameters $||\mathbf{w}||_2$ converges towards different values depending on the choice of the learning rate α. This inidicates that it is necessary to decrease the learning rate over time. By default, *ADAM* uses a constant learning rate, because the algorithm performs a kind of step size annealing by nature. However, popular implementations of *ADAM* in the Keras [219] and Lasagne [220] packages allow the use of an annealing schedule. I therefore tested different learning rate annealing schedules for *ADAM* assuming that with decreasing learning rates the L2-norm of the coupling parameters $||\mathbf{w}||_2$ will converge towards a consistent value. Indeed, as can be seen in Figure 7.6, when using a linear or sigmoidal learning rate annealing schedule with *ADAM*, the L2-norm of the coupling parameters $||\mathbf{w}||_2$ converges roughly towards the same value that has been obtained with plain SGD shown in Figure 4.8.

7.9 Computing the Gradient with Contrastive Divergence

This section describes the implementation details for approximating the gradient of the full likelihood with CD.

The gradient of the full log likelihood with respect to the couplings $\mathbf{w}$ is computed as the difference of paiwise amino acid counts between the input alignment and a sampled alignment plus an additional regularization term as given in eq. (4.1). Pairwise amino acid counts are computed from the input alignment accounting for sequence weights (described in methods section 7.3) and including pseudo counts (described in methods section 7.4). Pairwise amino acid counts for the sampled alignment are computed in the same way using the same sequence weights that have been computed for the input alignment. A subset of sequences of size

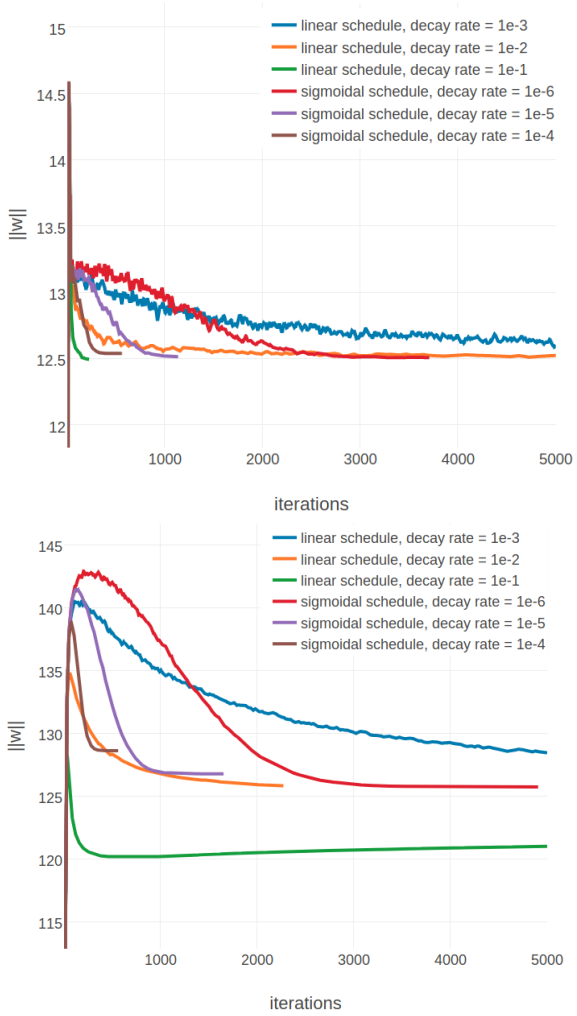

Figure 7.6: L2-norm of the coupling parameters $||\mathbf{w}||_2$ during optimization with *ADAM* and different learning rate annealing schedules. The learning rate α is specified with respect to Neff as $\alpha = 2\text{e}{-3}\log(N_{\text{eff}})$. The learning rate annealing schedule is specified in the legend. **Left** Convergence plot for protein 1mkc_A_00 having protein length L=43 and 142 sequences in the alignment (Neff=96). **Right** Convergence plot for protein 1c75_A_00 having protein length L=71 and 28078 sequences in the alignment (Neff=16808).

$S = \min(10L, N)$, with L being the length of sequences and N the number of sequences in the input alignment, is randomly selected from the input alignment and used to initialize the Markov chains for the Gibbs sampling procedure. Consequently, the input MSA is bigger than the sampled MSA whenever there are more than $10L$ sequences in the input alignment. In that case, the weighted pairwise amino acid counts of the sampled alignment need to be rescaled such that the total sample counts match the total counts from the input alignment.

During the Gibbs sampling process, every position in every sequence will be sampled K times (default $K = 1$), according to the conditional probabilties given in eq. (4.2). The sequence positions will be sampled in a random order to prevent position bias. Gap positions will not be sampled, because Dr. Stefan Seemayer showed that sampling gap positions leads to artefacts in the contat maps (not published). For PCD a copy of the input alignment is generated at the beginning of optimization that will keep the persistent Markov chains and that will be updated after the Gibbs sampling procedure. The default Gibbs sampling procedure is outlined in the following pseudo-code:

```
# Input: multiple sequence alignment X  with N sequences of length L
# Input: model parameters v and w

N = dim(X)[0]     # number of sequences in alignment
L = dim(X)[1]     # length of sequences in alignment
S = min(10L, N)   # number of sequences that will be sampled
K = 1             # number of Gibbs steps

# randomly select S sequences from the input alignment X without replacement
sequences = random.select.rows(X, size=S, replace=False)

for seq in sequences:
    # perform K steps of Gibbs sampling
    for step in range(K):
        # iterate over permuted sequence positions i in {1, ..., L}
        for i in shuffle(range(L)):
            # ignore gap positions
            if seq[i] == gap:
              continue
            # compute conditional probabilities for every
            # amino acid a in {1, ..., 20}
            for a in range(20):
              p_cond[a] = p(seq[i]=a | seq/i, v, w)
            # randomly select a new amino acid
            # a in {1, ..., 20} for position i
            # according to conditional probabilities
            seq[i] = random.integer({1, ...,20}, p_cond)

# sequences will now contain S newly sampled sequences
return sequences
```

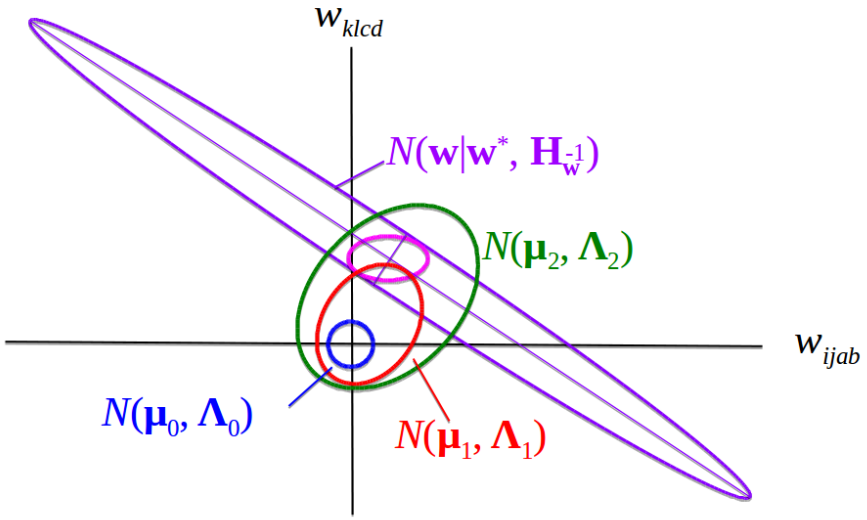

Figure 7.7: Setting the off-diagonal block matrices to zero in $\mathbf{H}$ corresponds to replacing the violett Gaussian distrubution by the pink one. The ratios between the overlaps of $\mathcal{N}\left(\mathbf{w}\,|\mathbf{w}^*, \mathbf{H}^{-1}\right)$ with the distributions $\mathcal{N}(\mathbf{w}_{ij}|\mu_k, \boldsymbol{\Lambda}_k^{-1})$ for various choices of k is only weakly affected by this replacement.

7.10 The Hessian off-diagonal Elements Carry a Negligible Signal

Assume that $\lambda_w = 0$, i.e., no regularisation is applied. Suppose in columns i and j a set of sequences in the MSA contain amino acids a and b and the same sequences contain c and d in columns k and l. Furthermore, assume that (a, b) occur nowhere else in columns i and j and the same holds for (c, d) in columns k and l. This means that the coupling between a at position i and b at position j can be perfectly compensated by the coupling between c at position k and d at position l. Adding 10^6 to w_{ijab} and substracting 10^6 from w_{klcd} leaves $p(\mathbf{X}|\mathbf{v}, \mathbf{w})$ unchanged. This means that w_{ijab} and w_{klcd} are almost perfectly negatively correlated in $\mathcal{N}(\mathbf{w}|\mathbf{w}^*, (\mathbf{H})^{-1})$. Another way to see this is to evaluate $(\mathbf{H})_{ijab,klcd}$ with eq. (7.43), which gives $(\mathbf{H})_{klcd,ijab} = N_{ij}\, p(x_i = a, x_j = b|\mathbf{v}^*, \mathbf{w}^*)\,(1 - p(x_i = a, x_j = b|\mathbf{v}^*, \mathbf{w}^*)$ for this case. Under the assumption $\lambda_w = 0$, this precision matrix element is the same as the diagonal elements $(\mathbf{H})_{ijab,ijab}$ and $(\mathbf{H})_{klcd,klcd}$ (see case 2 in eq. (7.43)).

But when a realistic regularisation constant is assumed, e.g. $\lambda_w = 0.2L \approx 20$, w_{ijab} and w_{klcd} will be pushed to near zero, because the matrix element that couples w_{ijab} with w_{klcd}, $N_{ij}\, p(x_i = a, x_j = b|\mathbf{v}^*, \mathbf{w}^*)\,(1 - p(x_i = a, x_j = b|\mathbf{v}^*, \mathbf{w}^*)$ is the number of sequences that share amino acids a and b at position (i, j) and c and d at position (k, l), and this number is usually much smaller than λ_w.

It is therefore a good approximation to set the off-diagonal block matrices $(\mathbf{H})_{klcd,ijab}$ (case 3 in eq. (7.43)) to zero. This corresponds to replacing the violett distribution in Figure 7.7 by the pink one. To see why, first note that the functions $g_k(c_{ij})$ and the component distributions $\mathcal{N}(\mathbf{w}_{ij}|\mu_k, \boldsymbol{\Lambda}_k^{-1})$ will be learned in such a way as to maximize the likelihood for predicting the correct contact state $\mathbf{c}^m$ from the respective alignments $\mathbf{X}^m$ for many MSAs of protein

families m. Therefore, these model parameters will adjust to the fact that the off-diagonal blocks in $\mathbf{H}$ are neglected. Second, note that the integral over the product of $\mathcal{N}(\mathbf{w}|\mathbf{w}^*, \mathbf{H}^{-1})$ and $\prod_{i<j} p(\mathbf{w}_{ij}|c_{ij})/\mathcal{N}(\mathbf{w}_{ij}|0, \lambda_w^{-1}\mathbf{I})$ in eq. (6.26) evaluates the overlap of these two Gaussians. Third, the components of $p(\mathbf{w}_{ij}|c_{ij})$ will be very much concentrated within a radius of less than 1 from the origin, because even residues with short C_β-C_β distance will rarely have coupling coefficients above 1. Fourth, the Gaussian components have no couplings between elements of $\mathbf{w}_{ij}$ and $\mathbf{w}_{kl}$, which is why they are axis-aligned (green in Figure 7.7). For these reasons, the relative strengths of the overlaps with different mixture components labeled by k in eq. (6.20) should be little affected by setting the off-diagonal block matrix couplings to zero.

7.11 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. (7.20) is

$$\begin{aligned}\frac{\partial^2 LL_{\mathrm{reg}}(\mathbf{v}^*, \mathbf{w})}{\partial w_{klcd}\, \partial w_{ijab}} &= -\sum_{n=1}^{N} \sum_{\mathbf{y} \in S_n} \frac{\partial \left(\frac{\exp\left(\sum_{i=1}^{L} v_i(y_i) + \sum_{1 \le i < j \le L}^{L} w_{ij}(y_i, y_j)\right)}{Z_n(\mathbf{v}, \mathbf{w})} \right)}{\partial w_{klcd}} I(y_i = a, y_j = b) \\ &\quad - \lambda_w \delta_{ijab,klcd}\,, \end{aligned} \tag{7.35}$$

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

$$\begin{aligned}\frac{\partial^2 LL_{\mathrm{reg}}(\mathbf{v}^*, \mathbf{w})}{\partial w_{klcd}\, \partial w_{ijab}} &= -\sum_{n=1}^{N} \sum_{\mathbf{y} \in S_n} \frac{\exp\left(\sum_{i=1}^{L} v_i(y_i) + \sum_{1 \le i < j \le L}^{L} w_{ij}(y_i, y_j)\right)}{Z_n(\mathbf{v}, \mathbf{w})} I(y_i = a, y_j = b) \\ &\quad \times \left[\frac{\partial}{\partial w_{klcd}} \left(\sum_{i=1}^{L} v_i(y_i) + \sum_{1 \le i < j \le L} w_{ij}(y_i, y_j) \right) - \frac{1}{Z_n(\mathbf{v}, \mathbf{w})} \frac{\partial Z_n(\mathbf{v}, \mathbf{w})}{\partial w_{klcd}} \right] \\ &\quad - \lambda_w \delta_{ijab,klcd} \end{aligned} \tag{7.36}$$

$$\begin{aligned}\frac{\partial^2 LL_{\mathrm{reg}}(\mathbf{v}^*, \mathbf{w})}{\partial w_{klcd}\, \partial w_{ijab}} &= -\sum_{n=1}^{N} \sum_{\mathbf{y} \in S_n} \frac{\exp\left(\sum_{i=1}^{L} v_i(y_i) + \sum_{1 \le i < j \le L}^{L} w_{ij}(y_i, y_j)\right)}{Z_n(\mathbf{v}, \mathbf{w})} I(y_i = a, y_j = b) \\ &\quad \times \left[I(y_k = c, y_l = d) - \frac{\partial}{\partial w_{klcd}} \log Z_n(\mathbf{v}, \mathbf{w}) \right] \\ &\quad - \lambda_w \delta_{ijab,klcd}\,. \end{aligned} \tag{7.37}$$

This expression can be simpified using

$$p(\mathbf{y}|\mathbf{v}, \mathbf{w}) = \frac{\exp\left(\sum_{i=1}^{L} v_i(y_i) + \sum_{1 \le i < j \le L} w_{ij}(y_i, y_j)\right)}{Z_n(\mathbf{v}, \mathbf{w})}, \tag{7.38}$$

yielding

$$\begin{aligned}\frac{\partial^2 LL_{\text{reg}}(\mathbf{v}^*,\mathbf{w})}{\partial w_{klcd}\,\partial w_{ijab}} &= -\sum_{n=1}^{N}\sum_{\mathbf{y}\in S_n} p(\mathbf{y}|\mathbf{v},\mathbf{w})\, I(y_i=a, y_j=b, y_k=c, y_l=d) \\ &\quad +\sum_{n=1}^{N}\sum_{\mathbf{y}\in \mathcal{S}_n} p(\mathbf{y}|\mathbf{v},\mathbf{w})\, I(y_i=a, y_j=b)\sum_{\mathbf{y}\in S_n} p(\mathbf{y}|\mathbf{v},\mathbf{w}) I(y_k=c, y_l=d) \\ &\quad -\lambda_w \delta_{ijab,klcd}\,. \end{aligned} \tag{7.39}$$

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

$$\begin{aligned}\frac{\partial^2 LL_{\text{reg}}(\mathbf{v}^*,\mathbf{w})}{\partial w_{klcd}\,\partial w_{ijab}} &= -N_{ijkl}\, p(x_i=a, x_j=b, x_k=c, x_l=d|\mathbf{v},\mathbf{w}) \\ &\quad +N_{ijkl}\, p(x_i=a, x_j=b|\mathbf{v},\mathbf{w})\, p(x_k=c, x_l=d|\mathbf{v},\mathbf{w}) - \lambda_w \delta_{ijab,klcd} \end{aligned} \tag{7.40}$$

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 $LL_{\text{reg}}(\mathbf{v}^*,\mathbf{w})$ with respect to the components of $\mathbf{w}$, are

$$\begin{aligned}\text{case 1}: (\mathbf{H})_{ijab,ijab} &= N_{ij}\, p(x_i=a, x_j=b|\mathbf{v}^*,\mathbf{w}^*)\,(1 - p(x_i=a, x_j=b|\mathbf{v}^*,\mathbf{w}^*)\,) \\ &\quad +\lambda_w \end{aligned} \tag{7.41}$$

$$\text{case 2}: (\mathbf{H})_{ijcd,ijab} = -N_{ij}\, p(x_i=a, x_j=b|\mathbf{v}^*,\mathbf{w}^*)\, p(x_i=c, x_j=d|\mathbf{v}^*,\mathbf{w}^*) \tag{7.42}$$

$$\begin{aligned}\text{case 3}: (\mathbf{H})_{klcd,ijab} &= N_{ijkl}\, p(x_i=a, x_j=b, x_k=c, x_l=d|\mathbf{v}^*,\mathbf{w}^*) \\ &\quad -N_{ijkl}\, p(x_i=a, x_j=b|\mathbf{v}^*,\mathbf{w}^*)\, p(x_k=c, x_l=d|\mathbf{v}^*,\mathbf{w}^*)\,. \end{aligned} \tag{7.43}$$

We know from eq. (7.22) that at the mode $\mathbf{w}^*$ the model probabilities match the empirical frequencies up to a small regularization term,

$$p(x_i=a, x_j=b|\mathbf{v}^*,\mathbf{w}^*) = q(x_i=a, x_j=b) - \frac{\lambda_w}{N_{ij}} w^*_{ijab}\,, \tag{7.44}$$

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

$$\begin{aligned}(\mathbf{H})_{ijab,ijab} &= N_{ij}\left(q(x_i=a, x_j=b) - \frac{\lambda_w}{N_{ij}} w^*_{ijab}\right)\left(1 - q(x_i=a, x_j=b) + \frac{\lambda_w}{N_{ij}} w^*_{ijab}\right) \\ &\quad +\lambda_w \end{aligned} \tag{7.45}$$

$$(\mathbf{H})_{ijcd,ijab} = -N_{ij}\left(q(x_i=a, x_j=b) - \frac{\lambda_w}{N_{ij}} w^*_{ijab}\right)\left(q(x_i=c, x_j=d) - \frac{\lambda_w}{N_{ij}} w^*_{ijcd}\right) \tag{7.46}$$

In order to write the previous eq. (7.46) in matrix form, the *regularised* empirical frequencies $\mathbf{q\prime}_{ij}$ will be defined as

$$(\mathbf{q\prime}_{ij})_{ab} = q'_{ijab} := q(x_i = a, x_j = b) - \lambda_w w^*_{ijab}/N_{ij} \,, \tag{7.47}$$

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

$$\mathbf{Q}_{ij} := \mathrm{diag}(\mathbf{q\prime}_{ij}) \,. \tag{7.48}$$

Now eq. (7.46) can be written in matrix form

$$\mathbf{H}_{ij} = N_{ij} \left(\mathbf{Q}_{ij} - \mathbf{q\prime}_{ij} \mathbf{q\prime}_{ij}^{\mathrm{T}} \right) + \lambda_w \mathbf{I} \,. \tag{7.49}$$

7.12 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 section 6.4 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. Equation (7.49) can be used to write $\mathbf{\Lambda}_{ij,k}$ in matrix form as

$$\mathbf{\Lambda}_{ij,k} = \mathbf{H}_{ij} - \lambda_w \mathbf{I} + \mathbf{\Lambda}_k = N_{ij} \mathbf{Q}_{ij} - N_{ij} \mathbf{q\prime}_{ij} \mathbf{q\prime}_{ij}^{\mathrm{T}} + \mathbf{\Lambda}_k \,. \tag{7.50}$$

Owing to eqs. (7.16) and (7.24), $\sum_{a,b=1}^{20} q'_{ijab} = 1$. The previous equation (7.50) facilitates the calculation of the inverse of this matrix using the *Woodbury identity* for matrices

$$(\mathbf{A} + \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{D} + \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1} \,. \tag{7.51}$$

by setting

$$\begin{aligned} \mathbf{A} &= N_{ij}\mathbf{Q}_{ij} + \mathbf{\Lambda}_k \\ \mathbf{B} &= \mathbf{q\prime}_{ij} \\ \mathbf{C} &= \mathbf{q\prime}_{ij}^{\mathrm{T}} \\ \mathbf{D} &= -N_{ij}^{-1} \end{aligned}$$

Now, the inverse of $\mathbf{\Lambda}_{ij,k}$ can be computed as

$$\begin{aligned} (\mathbf{H}_{ij} - \lambda_w \mathbf{I} + \mathbf{\Lambda}_k)^{-1} &= \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{q\prime}_{ij} \left(-N_{ij}^{-1} + \mathbf{q\prime}_{ij}^{\mathrm{T}} \mathbf{A}^{-1} \mathbf{q\prime}_{ij} \right)^{-1} \mathbf{q\prime}_{ij}^{\mathrm{T}} \mathbf{A}^{-1} \\ &= \mathbf{A}^{-1} + \frac{(\mathbf{A}^{-1}\mathbf{q\prime}_{ij})(\mathbf{A}^{-1}\mathbf{q\prime}_{ij})^{\mathrm{T}}}{N_{ij}^{-1} - \mathbf{q\prime}_{ij}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{q\prime}_{ij}} \,. \end{aligned} \tag{7.52}$$

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

$$N_{ij}^{-1} - \sum_{a,b=1}^{20} \frac{q'^2_{ijab}}{N_{ij}q'_{ijab} + (\mathbf{\Lambda}_k)_{ab,ab}} > N_{ij}^{-1} - \sum_{a,b=1}^{20} \frac{q'^2_{ijab}}{N_{ij}q'_{ijab}} = 0 \tag{7.53}$$

and therefore the inverse of $\mathbf{\Lambda}_{ij,k}$ in eq. (7.52) is well defined. The log determinant of $\mathbf{\Lambda}_{ij,k}$ is necessary to compute the ratio of Gaussians (see equation (6.38)) and can be computed using the matrix determinant lemma:

$$\det(\mathbf{A} + \mathbf{u}\mathbf{v}^{\mathrm{T}}) = (1 + \mathbf{v}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{u}) \det(\mathbf{A}) \tag{7.54}$$

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

$$\det(\mathbf{\Lambda}_{ij,k}) = \det(\mathbf{H}_{ij} - \lambda_w\mathbf{I} + \mathbf{\Lambda}_k) = (1 - N_{ij}\mathbf{q}\prime_{ij}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{q}\prime_{ij}) \det(\mathbf{A})\,. \tag{7.55}$$

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

7.13 Training the Hyperparameters in the Likelihood Function of Contact States

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 $\mathbf{c}^1, \ldots, \mathbf{c}^M$ plus a regularizer $R(\mu, \mathbf{\Lambda})$:

$$LL(\mu, \mathbf{\Lambda}, \gamma) + R(\mu, \mathbf{\Lambda}) = \sum_{n=1}^{M} \log p(\mathbf{X}^m | \mathbf{c}^m, \mu, \mathbf{\Lambda}, \gamma) + R(\mu, \mathbf{\Lambda}) \to \max\,. \tag{7.56}$$

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

$$R(\mu, \mathbf{\Lambda}) = -\frac{1}{2\sigma_\mu^2} \sum_{k=1}^{K} \sum_{ab=1}^{400} \mu_{k,ab}^2 - \frac{1}{2\sigma_{\mathrm{diag}}^2} \sum_{k=1}^{K} \sum_{ab=1}^{400} \Lambda_{k,ab,ab}^2 \tag{7.57}$$

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

The log likelihood can be optimized using L-BFGS-B [221], 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 m in c_{ij}^m, $(\mathbf{w}_{ij}^m)^*$ and $\mathbf{H}_{ij}^m$. From eq. (6.38) the log likelihood for a single protein is

$$LL(\mu, \mathbf{\Lambda}, \gamma_k) = \sum_{1 \le i < j \le L} \log \sum_{k=0}^{K} g_k(c_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} + R(\mu, \mathbf{\Lambda}) + \text{const.}\,. \tag{7.58}$$

For the optimization, I used the module `optimize.minimize` from the Python package `SciPy (v 0.19.1)` and the flag `method="L-BFGS-B"`. According to the default setting, optimization will converge when `(f^k - f^{k+1})/max{|f^k|,|f^{k+1}|,1} <= ftol` with `ftol=2.220446049250313e-09`.

7.13.1 Dataset Specifications

An equal number of residue pairs that are in physical contact $\Delta C_\beta < 8\mathring{A}$ and are not in contact $\Delta C_\beta > 8\mathring{A}$ is selected according to the following criteria:

- contact: $\Delta C_\beta < 8\mathring{A}$
- non-contact: $\Delta C_\beta > 8\mathring{A}$ or $\Delta C_\beta > 20\mathring{A}$
- diversity $(\frac{\sqrt{N}}{L}) > 0.3$
- percentage of gaps per column ≤ 0.5
- number of non-gapped sequences at position i and j, $N_{ij} > 1$
- maximum number of contacts selected per protein = 500
- maximum number of non-contacts selected per protein = 1000
- number residue pairs for contacts $(c_{ij} = 1)$ and non-contacts $(c_{ij} = 0) \in \{10000, 100000, 30000, 500000\}$

Before residue pairs are selected from a protein, they are shuffled to avoid position bias. Proteins from subsets 1-5 of the dataset described in method section 7.1 have been used for training. Training

7.13.2 Model Specifications

The mixture weights $g_k(c_{ij})$ are randomly sampled from a uniform distribution over the half-open interval [0, 1) and normalized so that $\sum_k^K g_k(c_{ij}) = 1$ for $c_{ij} = 0$ and $c_{ij} = 1$, respectively. Subsequently, the $g_k(c_{ij})$ are reparameterized as softmax functions as given in eq. (6.21) and fixing $\gamma_0(c_{ij}) = 0$ to avoid overparametrization.

The 400 dimensional μ_k vectors for $k \in \{1, \ldots, K\}$ are initialized from 400 random draws from a normal distribution with zero mean and standard deviation $\sigma = 0.05$. The zeroth component is kept fixed at zero ($\mu_{\mathbf{0}} = 0$) and will not be optimized.

The precision matrices $\mathbf{\Lambda}_k$ will be modelled as diagonal matrices, setting all off-diagonal elements to zero. The 400 diagonal elements $(\mathbf{\Lambda}_k)_{ab,ab}$ for $k \in \{1, \ldots, K\}$ are initialized from 400 random draws from a normal distribution with zero mean and standard deviation $\sigma = 0.005$. The 400 diagonal elements of the precision matrix for the zeroth component $\mathbf{\Lambda_0}$ are initialized as 400 random draws from a normal distribution with zero mean and standard deviation $\sigma = 0.0005$. Therefore, the zeroth component is sharply centered at zero. Furthermore, the diagonals of the precision matrices are reparameterized as $(\mathbf{\Lambda}_k)_{ab,ab} = \exp((\mathbf{\Lambda}_k)'_{ab,ab})$ in order to ensure that the values stay positive. Gradients for $\mathbf{\Lambda}_k$ derived in next sections have been adapted according to this reparametrization.

7.13.3 The gradient of the log likelihood with respect to μ_k

By applying the formula $df(x)/dx = f(x)\, d\log f(x)/dx$ to compute the gradient of eq. (7.58) (neglecting the regularization term) with respect to $\mu_{k,ab}$, one obtains

$$\frac{\partial}{\partial \mu_{k,ab}} LL(\mu, \mathbf{\Lambda}, \gamma_k) = \sum_{1 \leq i < j \leq L} \frac{g_k(c_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} \frac{\partial}{\partial \mu_{k,ab}} \log\left(\frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})}\right)}{\sum_{k'=0}^K g_{k'}(c_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu'_k, \mathbf{\Lambda}'^{-1}_k)}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})}}. \tag{7.59}$$

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:

$$p(k|ij) = \frac{g_k(c_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})}}{\sum_{k'=0}^{K} g_{k'}(c_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu'_k, \mathbf{\Lambda}'^{-1}_k)}{\mathcal{N}(\mathbf{0}|\mu'_{ij,k}, \mathbf{\Lambda}'^{-1}_{ij,k})}} \,. \tag{7.60}$$

By substituting the definition for responsibility, (7.59) simplifies

$$\frac{\partial}{\partial \mu_{k,ab}} LL(\mu, \mathbf{\Lambda}, \gamma_k) = \sum_{1 \le i < j \le L} p(k|ij) \frac{\partial}{\partial \mu_{k,ab}} \log \left(\frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} \right), \tag{7.61}$$

and analogously for partial derivatives with respect to $\Lambda_{k,ab,cd}$. The partial derivative inside the sum can be written

$$\frac{\partial}{\partial \mu_{k,ab}} \log \left(\frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} \right) = \frac{1}{2} \frac{\partial}{\partial \mu_{k,ab}} \left(\log |\mathbf{\Lambda}_k| - \mu_k^{\mathrm{T}} \mathbf{\Lambda}_k \mu_k - \log |\mathbf{\Lambda}_{ij,k}| + \mu_{ij,k}^{\mathrm{T}} \mathbf{\Lambda}_{ij,k} \mu_{ij,k} \right) . \tag{7.62}$$

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

$$\frac{\partial}{\partial x} \left(\mathbf{y}^{\mathrm{T}} \mathbf{A} \mathbf{y} \right) = \frac{\partial \mathbf{y}^{\mathrm{T}}}{\partial x} \mathbf{A} \mathbf{y} + \mathbf{y}^{\mathrm{T}} \mathbf{A} \frac{\partial \mathbf{y}}{\partial x} = \mathbf{y}^{\mathrm{T}} (\mathbf{A} + \mathbf{A}^{\mathrm{T}}) \frac{\partial \mathbf{y}}{\partial x} \tag{7.63}$$

the partial derivative therefore becomes

$$\begin{aligned} \frac{\partial}{\partial \mu_{k,ab}} \log \left(\frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} \right) &= \left(-\mu_k^{\mathrm{T}} \mathbf{\Lambda}_k \mathbf{e}_{ab} + \mu_{ij,k}^{\mathrm{T}} \mathbf{\Lambda}_{ij,k} \mathbf{\Lambda}_{ij,k}^{-1} \mathbf{\Lambda}_k \mathbf{e}_{ab} \right) \\ &= \mathbf{e}_{ab}^{\mathrm{T}} \mathbf{\Lambda}_k (\mu_{ij,k} - \mu_k) \,. \end{aligned} \tag{7.64}$$

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

$$\nabla_{\mu_k} LL(\mu, \mathbf{\Lambda}, \gamma_k) = \sum_{1 \le i < j \le L} p(k|ij) \, \mathbf{\Lambda}_k \left(\mu_{ij,k} - \mu_k \right) \,. \tag{7.65}$$

The correct computation of the gradient $\nabla_{\mu_k} LL(\mu, \mathbf{\Lambda}, \gamma_k)$ has been verified using finite differences.

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

Analogously to eq. (7.61) one first needs to solve

$$\frac{\partial}{\partial \Lambda_{k,ab,cd}} \log \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})}$$
$$= \frac{1}{2} \frac{\partial}{\partial \Lambda_{k,ab,cd}} \left(\log |\mathbf{\Lambda}_k| - \mu_k^{\mathrm{T}} \mathbf{\Lambda}_k \mu_k - \log |\mathbf{\Lambda}_{ij,k}| + \mu_{ij,k}^{\mathrm{T}} \mathbf{\Lambda}_{ij,k} \mu_{ij,k} \right) , \tag{7.66}$$

by applying eq. (7.63) as before as well as the formulas

$$\frac{\partial}{\partial x} \log |\mathbf{A}| = \mathrm{Tr} \left(\mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial x} \right) ,$$
$$\frac{\partial \mathbf{A}^{-1}}{\partial x} = -\mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial x} \mathbf{A}^{-1} . \tag{7.67}$$

This yields

$$\frac{\partial}{\partial \Lambda_{k,ab,cd}} \log |\mathbf{\Lambda}_k| = \mathrm{Tr} \left(\mathbf{\Lambda}_k^{-1} \frac{\partial \mathbf{\Lambda}_k}{\partial \Lambda_{k,ab,cd}} \right) = \mathrm{Tr} \left(\mathbf{\Lambda}_k^{-1} \mathbf{e}_{ab} \mathbf{e}_{cd}^{\mathrm{T}} \right) = \Lambda_{k,cd,ab}^{-1} \tag{7.68}$$
$$\frac{\partial}{\partial \Lambda_{k,ab,cd}} \log |\mathbf{\Lambda}_{ij,k}| = \mathrm{Tr} \left(\mathbf{\Lambda}_{ij,k}^{-1} \frac{\partial (\mathbf{H}_{ij} - \lambda_w \mathbf{I} + \mathbf{\Lambda}_k)}{\partial \Lambda_{k,ab,cd}} \right) = \Lambda_{ij,k,cd,ab}^{-1} \tag{7.69}$$
$$\frac{\partial (\mu_k^{\mathrm{T}} \mathbf{\Lambda}_k \mu_k)}{\partial \Lambda_{k,ab,cd}} = \mu_k^{\mathrm{T}} \mathbf{e}_{ab} \mathbf{e}_{cd}^{\mathrm{T}} \mu_k = \mathbf{e}_{ab}^{\mathrm{T}} \mu_k \mu_k^{\mathrm{T}} \mathbf{e}_{cd} = (\mu_k \mu_k^{\mathrm{T}})_{ab,cd} \tag{7.70}$$
$$\begin{aligned}
\frac{\partial (\mu_{ij,k}^{\mathrm{T}} \mathbf{\Lambda}_{ij,k} \mu_{ij,k})}{\partial \Lambda_{k,ab,cd}} =& \mu_{ij,k}^{\mathrm{T}} \frac{\partial \mathbf{\Lambda}_{ij,k}}{\partial \Lambda_{k,ab,cd}} \mu_{ij,k} + 2 \mu_{ij,k}^{\mathrm{T}} \mathbf{\Lambda}_{ij,k} \frac{\partial \mathbf{\Lambda}_{ij,k}^{-1}}{\partial \Lambda_{k,ab,cd}} (\mathbf{H}_{ij} \mathbf{w}_{ij}^* + \mathbf{\Lambda}_k \mu_k) \\
& + 2 \mu_{ij,k}^{\mathrm{T}} \frac{\partial \mathbf{\Lambda}_k}{\partial \Lambda_{k,ab,cd}} \mu_k \\
=& (\mu_{ij,k} \mu_{ij,k}^{\mathrm{T}} + 2 \mu_{ij,k} \mu_k^{\mathrm{T}})_{ab,cd} \\
& - 2 \mu_{ij,k}^{\mathrm{T}} \mathbf{\Lambda}_{ij,k} \mathbf{\Lambda}_{ij,k}^{-1} \frac{\partial \mathbf{\Lambda}_{ij,k}}{\partial \Lambda_{k,ab,cd}} \mathbf{\Lambda}_{ij,k}^{-1} (\mathbf{H}_{ij} \mathbf{w}_{ij}^* + \mathbf{\Lambda}_k \mu_k) \\
=& (\mu_{ij,k} \mu_{ij,k}^{\mathrm{T}} + 2 \mu_{ij,k} \mu_k^{\mathrm{T}})_{ab,cd} - 2 \mu_{ij,k}^{\mathrm{T}} \frac{\partial \mathbf{\Lambda}_{ij,k}}{\partial \Lambda_{k,ab,cd}} \mu_{ij,k} \\
=& (-\mu_{ij,k} \mu_{ij,k}^{\mathrm{T}} + 2 \mu_{ij,k} \mu_k^{\mathrm{T}})_{ab,cd} .
\end{aligned} \tag{7.71}$$

Inserting these results into eq. (7.66) yields

$$\frac{\partial}{\partial \Lambda_{k,ab,cd}} \log \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} = \frac{1}{2} \left(\mathbf{\Lambda}_k^{-1} - \mathbf{\Lambda}_{ij,k}^{-1} - (\mu_{ij,k} - \mu_k)(\mu_{ij,k} - \mu_k)^{\mathrm{T}} \right)_{ab,cd} . \tag{7.72}$$

Substituting this expression into the equation (7.61) analogous to the derivation of gradient for $\mu_{k,ab}$ yields the equation

$$\nabla_{\mathbf{\Lambda}_k} LL(\mu, \mathbf{\Lambda}, \gamma_k) = \frac{1}{2} \sum_{1 \leq i < j \leq L} p(k|ij) \left(\mathbf{\Lambda}_k^{-1} - \mathbf{\Lambda}_{ij,k}^{-1} - (\mu_{ij,k} - \mu_k)(\mu_{ij,k} - \mu_k)^{\mathrm{T}} \right) . \tag{7.73}$$

The correct computation of the gradient $\nabla_{\mathbf{\Lambda}_k} LL(\mu, \mathbf{\Lambda}, \gamma_k)$ has been verified using finite differences.

7.13.5 The gradient of the log likelihood with respect to γ_k

With $c_{ij} \in \{0, 1\}$ defining a residue pair in physical contact or not in contact, the mixing weights can be modelled as a softmax function according to eq. (6.21). The derivative of the mixing weights $g_k(c_{ij})$ is:

$$\frac{\partial g_{k'}(c_{ij})}{\partial \gamma_k} = \begin{cases} g_k(c_{ij})(1 - g_k(c_{ij})) & : k' = k \\ g_{k'}(c_{ij}) - g_k(c_{ij}) & : k' \neq k \end{cases} \tag{7.74}$$

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

$$\begin{aligned} \frac{\partial}{\partial \gamma_k} LL(\mu, \mathbf{\Lambda}, \gamma_k) &= \sum_{1 \leq i < j \leq L} \frac{\sum_{k'=0}^{K} \frac{\partial}{\partial \gamma_k} g_{k'}(c_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(0|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})}}{\sum_{k'=0}^{K} g_{k'}(c_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})}} \\ &= \sum_{1 \leq i < j \leq L} \frac{\sum_{k'=0}^{K} g_{k'}(c_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} \cdot \begin{cases} 1 - g_k(c_{ij}) & \text{if } k' = k \\ -g_k(c_{ij}) & \text{if } k' \neq k \end{cases}}{\sum_{k'=0}^{K} g_{k'}(c_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})}} \\ &= \sum_{1 \leq i < j \leq L} \sum_{k'=0}^{K} p(k'|ij) \begin{cases} 1 - g_k(c_{ij}) & \text{if } k' = k \\ -g_k(c_{ij}) & \text{if } k' \neq k \end{cases} \\ &= \sum_{1 \leq i < j \leq L} p(k|ij) - g_k(c_{ij}) \sum_{k'=0}^{K} p(k'|ij) \\ &= \sum_{1 \leq i < j \leq L} p(k|ij) - g_k(c_{ij}) \end{aligned} \tag{7.75}$$

7.14 Extending the Bayesian Statistical Model for the Prediction of Protein Residue-Residue Distances

It is straightforward to extend the Baysian model for contact prediction presented in section 6.1 for distances. The prior over couplings will modelled using distance dependent mixture weights $g_k(c_{ij})$. Therefore eq. (6.20) is modified such that mixture weights $g_k(c_{ij})$ are modelled as softmax over linear functions $\gamma_k(c_{ij})$ (see Figure 7.8:

$$g_k(c_{ij}) = \frac{\exp \gamma_k(c_{ij})}{\sum_{k'=0}^{K} \exp \gamma_{k'}(c_{ij})}, \tag{7.76}$$

$$\gamma_k(c_{ij}) = -\sum_{k'=0}^{k} \alpha_{k'}(c_{ij} - \rho_{k'}). \tag{7.77}$$

The functions $g_k(c_{ij})$ remain invariant when adding an offset to all $\gamma_k(c_{ij})$. This degeneracy can be removed by setting $\gamma_0(c_{ij}) = 0$ (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 $c_{ij} \to \infty$ we will obtain $g_0(c_{ij}) \to 1$ and hence $p(\mathbf{w}|\mathbf{X}) \to \mathcal{N}(0, \sigma_0^2 \mathbf{I})$.

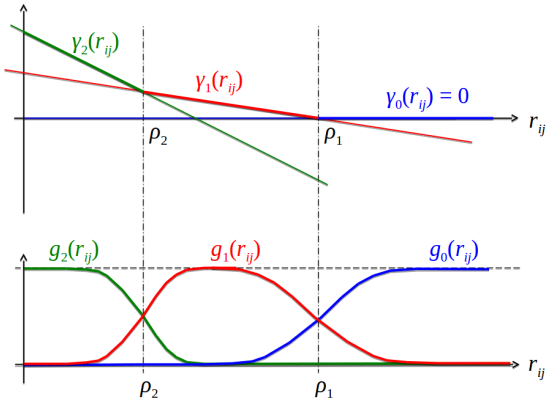

Figure 7.8: The Gaussian mixture coefficients $g_k(c_{ij})$ of $p(\mathbf{w}_{ij}|c_{ij})$ are modelled as softmax over linear functions $\gamma_k(c_{ij})$. ρ_k sets the transition point between neighbouring components $g_{k-1}(c_{ij})$ and $g_k(c_{ij})$, while α_k quantifies the abruptness of the transition between $g_{k-1}(c_{ij})$ and $g_k(c_{ij})$.

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 $\gamma_k(c_{ij}) - \gamma_{k-1}(r) = \alpha_t(c_{ij} - \rho_t)$ and hence $\gamma_{k-1}(\rho_k) == \gamma_k(\rho_k)$. A change in ρ_k or α_k only changes the behaviour of $g_{k-1}(c_{ij})$ and $g_k(c_{ij})$ in the transition region around ρ_k. Therefore, this particular definition of $\gamma_k(c_{ij})$ makes the parameters α_k and ρ_k as independent of each other as possible, rendering the optimisation of these parameters more efficient.

7.14.1 The derivative of the log likelihood with respect to ρ_k

Analogous to the derivations of μ_k in section 7.13.3 and $\mathbf{\Lambda}_k$ in section 7.13.4, the partial derivative with respect to ρ_k is

$$\frac{\partial}{\partial \rho_k} LL(\mu, \mathbf{\Lambda}, \rho, \alpha) = \sum_{1 \le i < j \le L} \sum_{k'=0}^{K} p(k'|ij) \, \frac{\partial}{\partial \rho_k} \log g_{k'}(c_{ij}) \,. \tag{7.78}$$

Using the definition of $g_k(c_{ij})$ in eq. (7.77), we find (rember that $\alpha_0 = 0$ as noted in the last section) that

$$
\begin{aligned}
\frac{\partial}{\partial \rho_k} \log g_l(c_{ij}) &= \frac{\partial}{\partial \rho_k} \log \frac{\exp\left(-\sum_{k''=1}^{k'} \alpha_{k''}(c_{ij} - \rho_{k''})\right)}{\sum_{k'=0}^{K} \exp\left(-\sum_{k''=1}^{k'} \alpha_{k''}(c_{ij} - \rho_{k''})\right)} \\
&= -\frac{\partial}{\partial \rho_k} \sum_{k''=1}^{l} \alpha_{k''}(c_{ij} - \rho_{k''}) - \frac{\partial}{\partial \rho_k} \log \sum_{k'=0}^{K} \exp\left(-\sum_{k''=1}^{k'} \alpha_{k''}(c_{ij} - \rho_{k''})\right) \\
&= \alpha_k \, I(l \geq k) - \frac{\sum_{k'=0}^{K} \frac{\partial}{\partial \rho_k} \exp(-\sum_{k''=1}^{k'} \alpha_{k''}(c_{ij} - \rho_{k''}))}{\sum_{k'=0}^{K} \exp(-\sum_{k''=1}^{k'} \alpha_{k''}(c_{ij} - \rho_{k''}))} \\
&= \alpha_k \, I(l \geq k) - \frac{\sum_{k'=0}^{K} \exp(-\sum_{k''=1}^{k'} \alpha_{k''}(c_{ij} - \rho_{k''}))\, \alpha_k \, I(k' \geq k)}{\sum_{k'=0}^{K} \exp(-\sum_{k''=1}^{k'} \alpha_{k''}(c_{ij} - \rho_{k''}))} \\
&= \alpha_k \, I(l \geq k) - \frac{\sum_{k'=0}^{K} \exp(\gamma_{k'}(c_{ij}))\, \alpha_k \, I(k' \geq k)}{\sum_{k'=0}^{K} \exp(\gamma_{k'}(c_{ij}))} \\
&= \alpha_k \, I(l \geq k) - \sum_{k'=0}^{K} g_{k'}(c_{ij})\, \alpha_k \, I(k' \geq k) \\
&= \alpha_k \left(I(l \geq k) - \sum_{k'=k}^{K} g_{k'}(c_{ij}) \right) . \qquad (7.79)
\end{aligned}
$$

Inserting this into eq. (7.78) yields

$$
\begin{aligned}
\frac{\partial}{\partial \rho_k} LL(\mu, \mathbf{\Lambda}, \rho, \alpha) &= \sum_{1 \leq i < j \leq L} \sum_{k'=0}^{K} p(k'|ij)\, \alpha_k \left(I(k' \geq k) - \sum_{k''=k}^{K} g_{k''}(c_{ij}) \right) \\
&= \alpha_k \sum_{1 \leq i < j \leq L} \left(\sum_{k'=k}^{K} p(k'|ij) - \sum_{k'=0}^{K} p(k'|ij) \sum_{k''=k}^{K} g_{k''}(c_{ij}) \right) , \qquad (7.80)
\end{aligned}
$$

and finally

$$
\frac{\partial}{\partial \rho_k} LL(\mu, \mathbf{\Lambda}, \rho, \alpha) = \alpha_k \sum_{1 \leq i < j \leq L} \sum_{k'=k}^{K} \left(p(k'|ij) - g_{k'}(c_{ij}) \right) . \qquad (7.81)
$$

This equation has an intuitive meaning: The gradient is the difference between the summed probability mass predicted to be due to components $k' \geq k$, $p(k' \geq k|ij)$, and the sum of the prior probabilities $g_k(c_{ij})$ for components $k' \geq k$, where the sum runs over all training points indexed by i, j.

7.14.2 The derivative of the log likelihood with respect to α_k

Last and similar to the previous derivation, the partial derivative with respect to α_k is

$$
\frac{\partial}{\partial \alpha_k} LL(\mu, \mathbf{\Lambda}, \rho, \alpha) = \sum_{1 \leq i < j \leq L} \sum_{k'=0}^{K} p(k'|ij) \, \frac{\partial}{\partial \alpha_k} \log g_{k'}(c_{ij}) . \qquad (7.82)
$$

Similarly as before,

$$
\begin{aligned}
\frac{\partial}{\partial \alpha_k} \log g_l(c_{ij}) &= \frac{\partial}{\partial \alpha_k} \log \frac{\exp(-\sum_{k''=1}^{l} \alpha_{k''}(c_{ij} - \rho_{k''})}{\sum_{k'=0}^{K} \exp(-\sum_{k''=1}^{k'} \alpha_{k''}(c_{ij} - \rho_{k''}))} \\
&= -\frac{\partial}{\partial \alpha_k} \sum_{k''=1}^{l} \alpha_{k''}(c_{ij} - \rho_{k''}) - \frac{\partial}{\partial \alpha_k} \log \sum_{k'=0}^{K} \exp\left(-\sum_{k''=1}^{k'} \alpha_{k''}(c_{ij} - \rho_{k''})\right) \\
&= -(c_{ij} - \rho_k)\, I(l \geq k) - \frac{\sum_{k'=0}^{K} \frac{\partial}{\partial \alpha_k} \exp(-\sum_{k''=1}^{k'} \alpha_{k''}(c_{ij} - \rho_{k''}))}{\sum_{k'=0}^{K} \exp(-\sum_{k''=1}^{k'} \alpha_{k''}(c_{ij} - \rho_{k''}))} \\
&= -(c_{ij} - \rho_k) \left(I(l \geq k) - \sum_{k''=k}^{K} g_{k''}(c_{ij}) \right) . \qquad (7.83)
\end{aligned}
$$

Inserting this into eq. (7.82) yields

$$
\begin{aligned}
\frac{\partial}{\partial \alpha_k} LL(\mu, \mathbf{\Lambda}, \rho, \alpha) &= -\sum_{1 \leq i < j \leq L} \sum_{k'=0}^{K} p(k'|ij)\,(c_{ij} - \rho_k) \left(I(k' \geq k) - \sum_{k''=k}^{K} g_{k''}(c_{ij}) \right) \\
&= -\sum_{1 \leq i < j \leq L} (c_{ij} - \rho_k) \left(\sum_{k'=k}^{K} p(k'|ij) - \sum_{k'=0}^{K} p(k'|ij) \sum_{k''=k}^{K} g_{k''}(c_{ij}) \right) , \qquad (7.84)
\end{aligned}
$$

and finally

$$
\frac{\partial}{\partial \alpha_k} LL(\mu, \mathbf{\Lambda}, \rho, \alpha) = \sum_{1 \leq i < j \leq L} (\rho_k - c_{ij}) \sum_{k'=k}^{K} (p(k'|ij) - g_{k'}(c_{ij})) . \qquad (7.85)
$$

7.15 Features used to train Random Forest Model

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 7.3.

7.15.1 Global Features

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

Figure 7.9: Observed number of contacts per residue has a non-linear relationship with protein length. Distribution is shown for several thresholds of sequence separation |j-i|.

Table 7.1: Features characterizing the total alignment

Feature	Description	No. Features per residue pair (i, j)
L	log of protein length	1
N	number of sequences	1
Neff	number of effective sequences computed as the sum over sequence weights (see section 7.3)	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
Psipred	secondary structure prediction by PSIPRED (v4.0)[217] given as average three state propensities	3
NetsurfP	secomdary structure prediction by Netsurfp (v1.0)[216] given as average three state propensities	3
contact prior protein length	simple contact predictor based on expected number of contacts per protein with respect to protein length (see description below)	1

There are in total 32 global alignment features per reside pair.

The last feature listed in table 7.1 ("contact prior protein length") stands for a simple contact predictor based on expected number of contacts per protein with respect to protein length. 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 7.9.

In log space, the average number of contats per residue can be fitted with a linear regression (see Figure 7.10) and yields the following functions:

Figure 7.10: Distribution of average number of contacts per residue against protein length and corresponding linear regression fits. Protein length is on logarithmic scale. Distribution and linear regression fits are shown for different sequence separation thresholds |j-i|.

- $f(L) = 1.556 + 0.596 \log(L)$ 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.

7.15.2 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 7.2: Caption hereSingle Position Sequence Features

Feature	Description	No. Features per residue pair (i, j)
shannon entropy (excluding gaps)	$-\sum_{a=1}^{20} p_a \log p_a$	2
shannon entropy (including gaps)	$-\sum_{a=1}^{21} p_a \log p_a$	2

Feature	Description	No. Features per residue pair (i, j)
kullback leibler divergence	between weighted observed and background amino acid frequencies [222]	2
jennson shannon divergence	between weighted observed and background amino acid frequencies [222]	2
PSSM	log odds ratio of weighted observed and background amino acid frequencies [222]	40
secondary structure prediction	three state propensities PSIPRED (v4.0) [217]	6
secondary structure prediction	three state propensities Netsurfp (v1.0) [216]	6
solvent accessibility prediction	RSA and RSA Z-score Netsurfp (v1.0) [216]	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 Atchley Factor	Atchley Factors 1-5 [223]	10
Average polarity (Grantham)	Polarity according to Grantham [224]. Data taken from AAindex Database [225].	2
Average polarity (Zimmermann)	Polarity according to Zimmermann et al. [226]. Data taken from AAindex Database [225].	2
Average isoelectricity	Isoelectric point according to Zimmermann et al. [226]. Data taken from AAindex Database [225].	2
Average hydrophobicity (Wimley&White)	Hydrophobicity scale according to Wimley & White [227]. Data taken from UCSF Chimera [227].	2
Average hydrophobicity (Kyte&Dolittle)	Hydrophobicity index according to Kyte & Doolittle [228]. Data taken from AAindex Database [225].	2
Average hydrophobicity (Cornette)	Hydrophobicity according to Cornette [229].	2
Average bulkiness	Bulkiness according to Zimmerman et al. [226]. Data taken from AAindex Database [225].	2
Average volume	Average volumes of residues according to Pontius et al. [230]. Data taken from AAindex Database [225].	2

There are 48 single sequnece features per residue and consequently 96 single sequence features per residue pair.

Additionally, all single features will be computed within a window of size 5. The window fea-

ture 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 per residue pair.

7.15.3 Pairwise Features

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

Table 7.3: 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 table 7.2	13
pairwise potential (buried)	Average quasi-chemical energy of interactions in an average buried environment according to Miyazawa&Jernigan [215]. Data taken from AAindex Database [225].	1
pairwise potential (water)	Average quasi-chemical energy of transfer of amino acids from water to the protein environment according to Miyazawa&Jernigan [215]. Data taken from AAindex Database [225].	1
pairwise potential (Li&Fang)	Average general contact potential by Li&Fang [69]	1
pairwise potential (Zhu&Braun)	Average statistical potential from residue pairs in beta-sheets by Zhu&Braun [231]	1
joint shannon entropy (excluding gaps)	$-\sum_{a=1}^{20}\sum_{b=1}^{20} p(a,b)\log p(a,b)$	1
joint shannon entropy (including gaps)	$-\sum_{a=1}^{21}\sum_{b=1}^{21} p(a,b)\log p(a,b)$	1
normalized MI	normalized mutual information of amino acid counts at two positions	1
MI (+pseudo-counts)	mutual information of amino acid counts at two positions, including uniform pseudo-counts	1
MI (+pseudo-counts + APC)	mutual information of amino acid counts at two positions; including pseudo-counts and average product correction	1
OMES coeevolution score	according to Fodor&Aldrich [214] with and without APC	2

Feature	Description	No. Features per residue pair (i, j)

There are in total 26 pairwise sequence features.

7.16 Training Random Forest Contact Prior

Proteins constitute highly imbalanced datasets with respect to the number of residue pairs that form and do not form physical contacts. As can be seen in Figure 7.11, depending on the enforced sequence separation threshold and protein length the percentage of contacts per protein varies between 25% and 0%. Most studies applying machine learning algorithms for predicting residue-residue contacts rebalanced the data set by undersampling of the majority class. Table 7.4 lists choices for the proportion of contacts to non-contacts used to train some machine learning contact predictors. 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.

Table 7.4: Important machine learning contact prediction approaches and their choices for rebalancing the data set.

Study	Machine Learning Algorithm	Proportion of Contacts : Non-contacts
Wu et al. (2008) [68]	SVM	1:4
Li et al. (2011) [69]	Random Forest	1:1, 1:2
Wang et al. (2011) [70]	Random Forest	1:4
DiLena et al. (2012) [78]	deep neural network	1:≈4 (sampling 20% of non-contacts)
Wang et al. (2013) [71]	Random Forest	1:≈ 4 (sampling 20% of non-contacts)

The total training set is comprised of 50,000 residue pairs $< 8\mathring{A}$ ("contacts") and 250,000 residue pairs $> 8\mathring{A}$ ("non-contacts"). I filtered residue pairs using a sequence separation of 12 positions and selected at maximum 100 contacts and 500 non-contacts per protein. The data is collected in equal parts from data subsets 1-5 (see methods section 7), so that the training set consists of five subsets that are non-redundand at the fold level. Each of the five models for cross-validation will be trained on 40,000 contacts and 200,000 non-contacts originating from four of the five subsets. As the training set has been undersampled for non-contacts, it is not representative of real world proteins and the models need to be validated on a more realistic validation set. Therefore, each of the five trained models is not validated on the hold-out set but on separate validation sets containing 40 proteins at a time. The proteins of the validation sets are randomly selected from the respective fifth data subset and consequently are non-redundant at the fold level with training data. Performance is assessed by means of the standard contact prediction benchmark (mean precision against top ranked contacts).

I used the module RandomForestClassifier in the Python package `sklearn (v. 0.19)` [232] and trained the models on features extracted from MSAs which are listed in methods section 7.15.

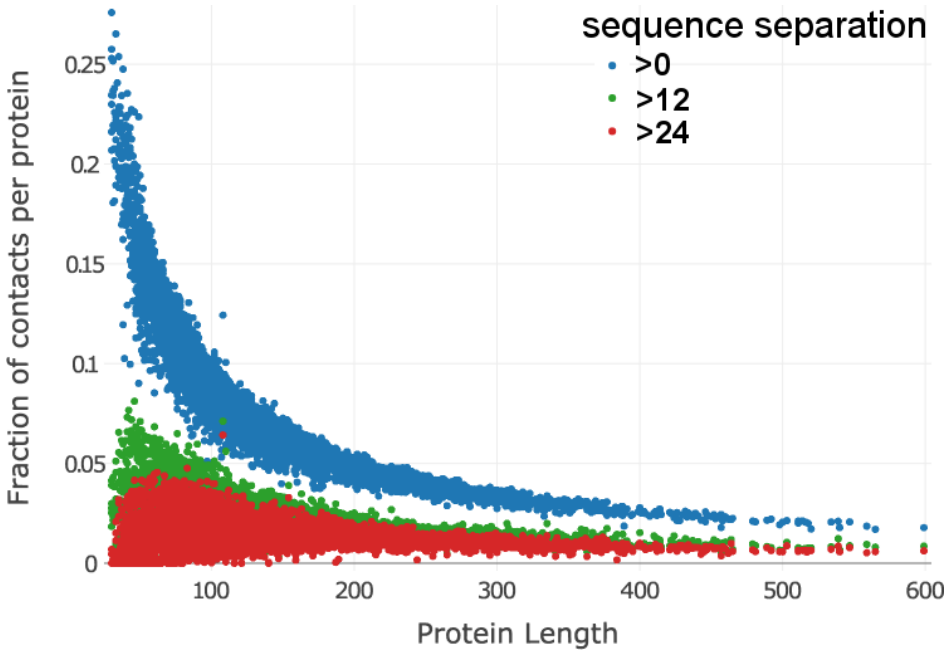

Figure 7.11: Fraction of contacts among all possible contacts in a protein against protein length L. The distribution has a non-linear relationship. At a sequence separation >8 positions the fraction of contacts for intermediate size proteins with length >100 is approximately 2%. Data set contains 6368 proteins and is explained in methods section 7.1.

7.16.1 Feature Selection

A random forest model is trained on the total set of features. Given the distribution of *Gini importance* values of features from the model, subsets of features are defined by features having *Gini importance* values larger than the $\{10, 30, 50, 70, 90\}$-percentile of the distribution. Performance of the models trained on these subsets of features is evaluated on the same validation set.

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

B

Amino Acid Alphabet

Table B.1: Amino acid abbreviations and physico-chemical properties according to Livingstone et al., 1993 [233]

One letter Code	Three letter Code	Amino Acid	Physico-chemical properties
A	Ala	**A**lanine	tiny, hydrophobic
C	Cys	**C**ysteine	small, hydrophobic, polar (C_{S-H})
D	Asp	Aspartic Aci**D**	small, negatively charged, polar
E	Glu	Glutamic Acid	negatively charged, polar
F	Phe	Phenylalanine	aromatic, hydrophobic
G	Gly	**G**lycine	tiny, hydrophobic
H	His	**H**istidine	hydrophobic, aromatic, polar, (positively charged)
I	Ile	**I**soleucine	aliphatic, hydrophobic
K	Lys	Lysine	positively charged, polar
L	Leu	**L**eucine	aliphatic, hydrophobic
M	Met	**M**ethionine	hydrophobic
N	Asn	AsparagiN**e**	small, polar
P	Pro	**P**roline	small
Q	Gln	Glutamine	tiny, hydrophobic
R	Arg	A**R**ginine	positively charged, polar
S	Ser	**S**erine	tiny, polar
T	Thr	**T**hreonine	hydrophobic, polar
V	Val	**V**aline	small, aliphatic
W	Trp	**T**ryptophan	aromatic, hydrophobic, polar
Y	Tyr	T**Y**rosine	aromatic, hydrophobic, polar

C

Dataset Properties

The following figures display various statistics about the dataset used throughout this thesis. See section 7.1 for information on how this dataset has been generated.

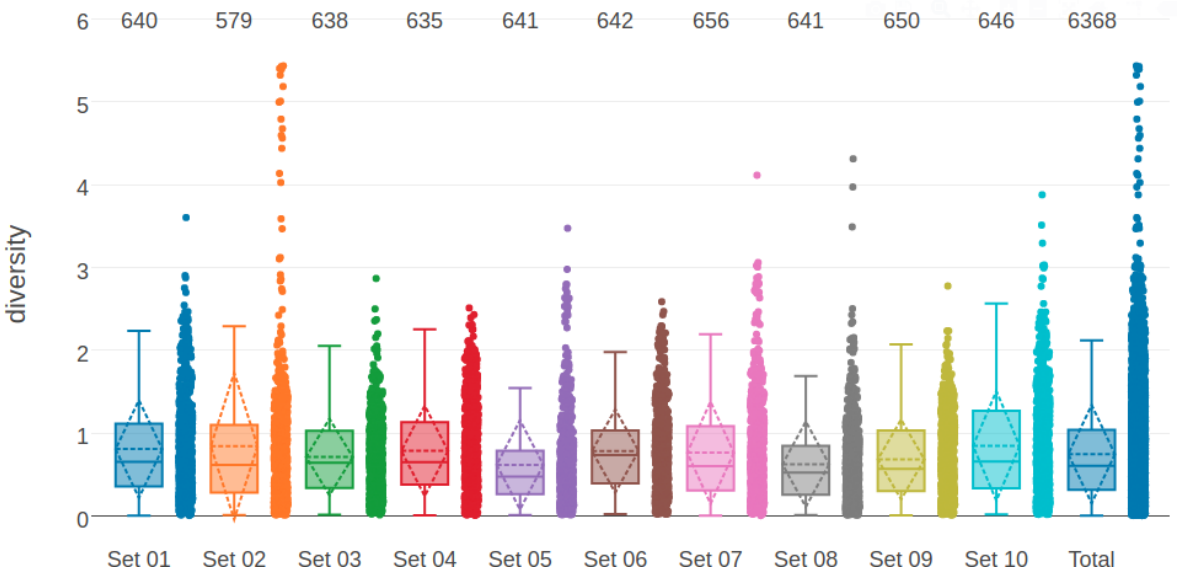

Figure C.1: Distribution of alignment diversity ($= \sqrt{(\frac{N}{L})}$) in the dataset and its ten subsets.

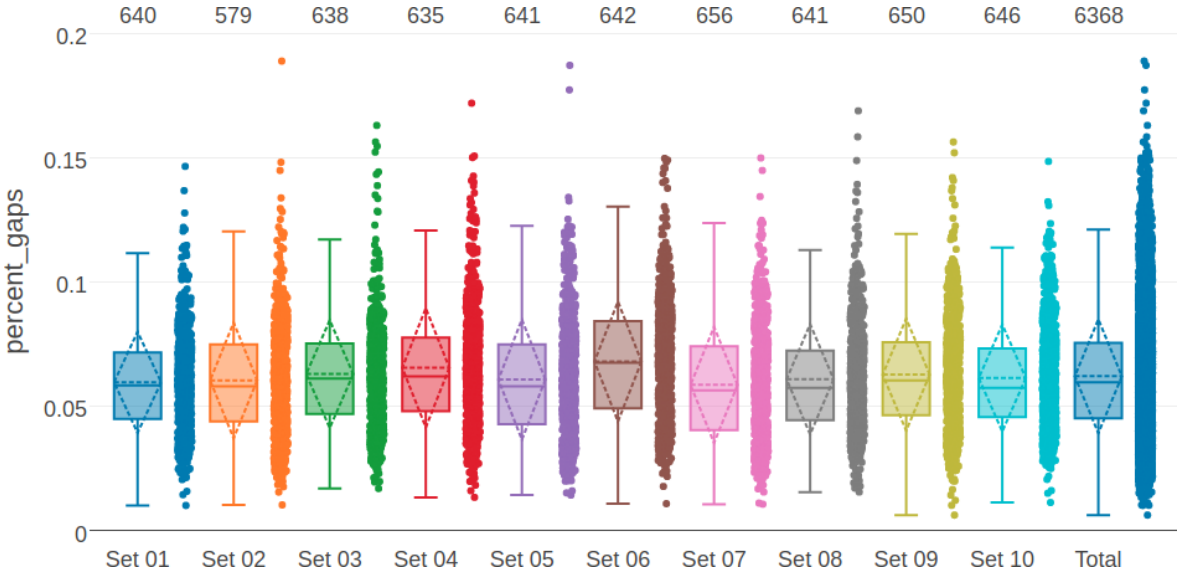

Figure C.2: Distribution of gap percentage of alignments in the dataset and its ten subsets.

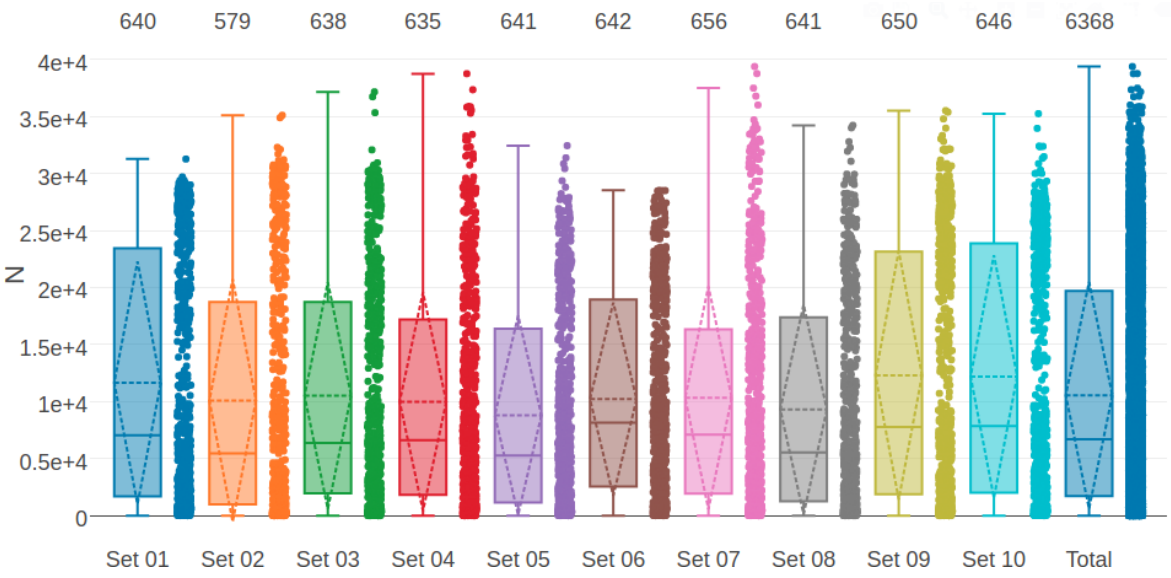

Figure C.3: Distribution of alignment size (number of sequences N) in the dataset and its ten subsets.

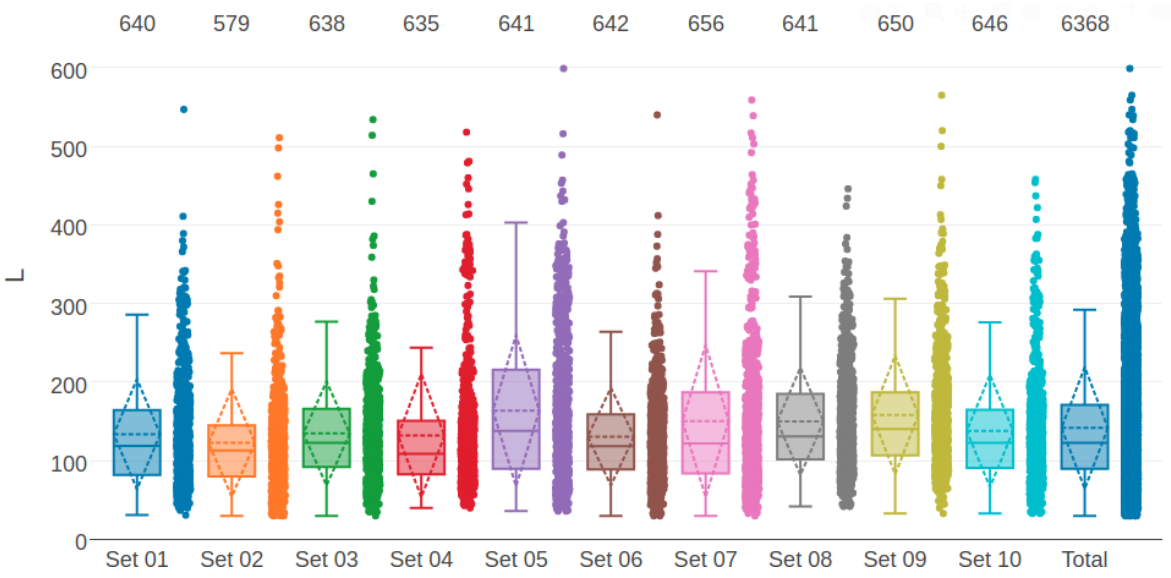

Figure C.4: Distribution of protein length L in the dataset and its ten subsets.

D

Standard Deviation of Couplings for Noncontacts

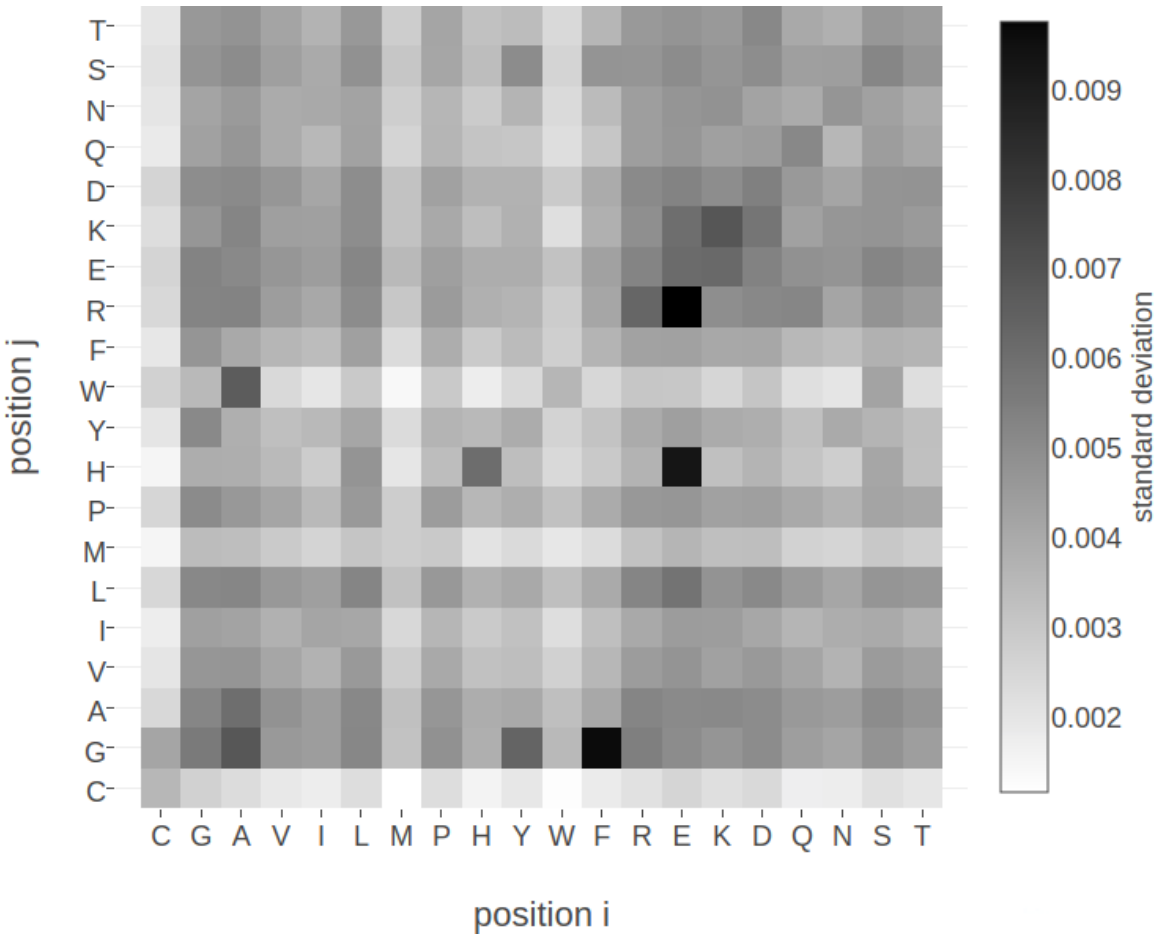

Figure D.1: Standard deviation of squared coupling values ${w_{ijab}}^2$ for residue pairs not in physical contact ($\Delta C_\beta > 25\mathring{A}$). Dataset contains 100.000 residue pairs per class (for details see methods section 7.7.1). Amino acids are abbreviated with one-letter code and they are broadly grouped with respect to physico-chemical properties listed in Appendix B

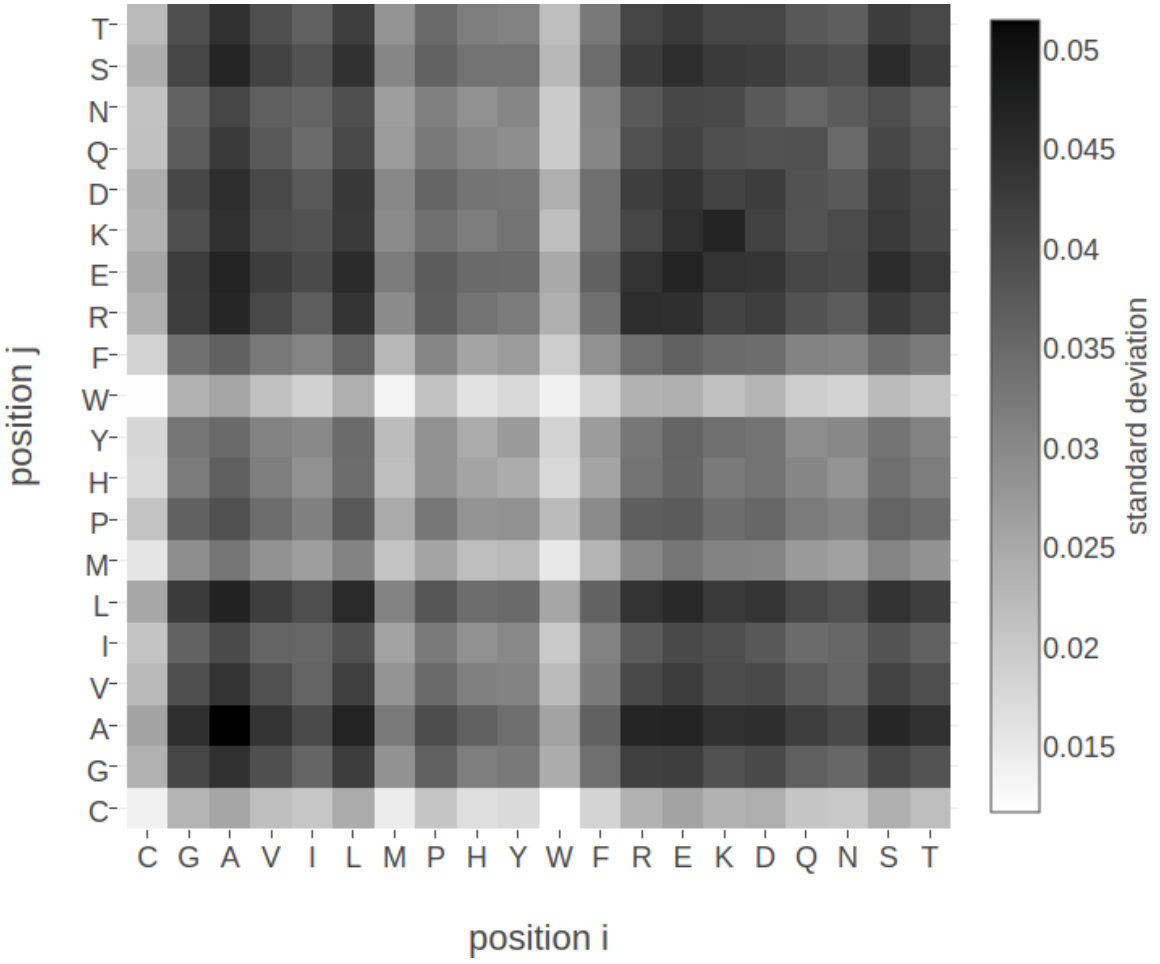

Figure D.2: Standard deviation of coupling values w_{ijab} for residue pairs not in physical contact ($\Delta C_\beta > 25\mathring{A}$). Dataset contains 100.000 residue pairs per class (for details see section 7.7.1). Amino acids are abbreviated with one-letter code and they are broadly grouped with respect to physico-chemical properties listed in Appendix B.

E

Amino Acid Interaction Preferences Reflected in Coupling Matrices

E.1 Pi-Cation interactions

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

E.2 Disulfide Bonds

Figure E.2 shows two cysteine residues forming a covalent disulfide bond in protein 1alu. The corresponding coupling matrix in figure **??** reflects the strong interaction preference of cysteines.

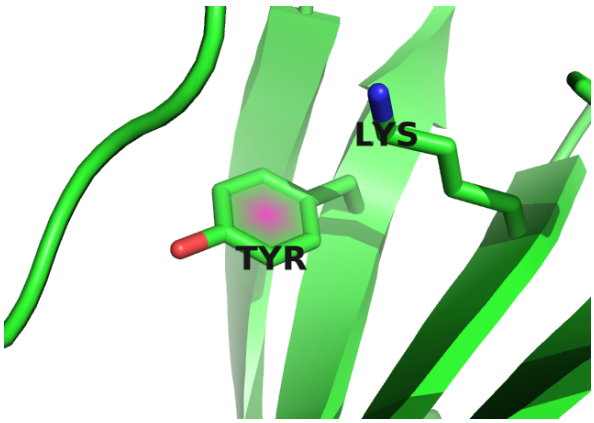

Figure E.1: Tyrosine (residue 37) and Lysine (residue 48) forming a cation-π interaction in protein 2ayd.

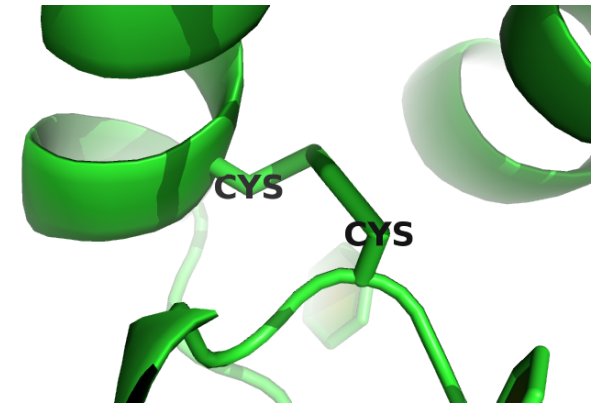

Figure E.2: Two cystein residues (residues 54 and 64) forming a covalent disulfide bond in protein 1alu.

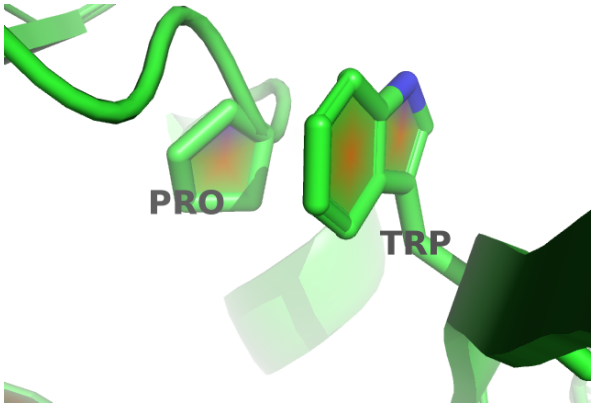

Figure E.3: Proline and tryptophan (residues 17 and 34) stacked on top of each otherengaging in a CH/π interaction in protein chain 1aol_A_00.

E.3 Aromatic-Proline Interactions

Figure E.3 shows a proline and a tryptophan residue forming such a CH/π interaction in protein 1aol. The corresponding coupling matrix in figure **??** reflects this interaction with strong positive coupling between proline and tryptophan.

E.4 Network-like structure of aromatic residues

E.5 Aromatic Sidechains at small Cb-C_β distances

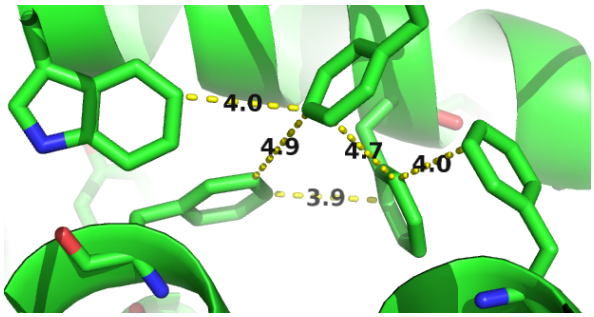

Figure E.4: Network-like structure of aromatic residues in the protein core. 80% of aromatic residues are involved in such networks that are important for protein stability [187].

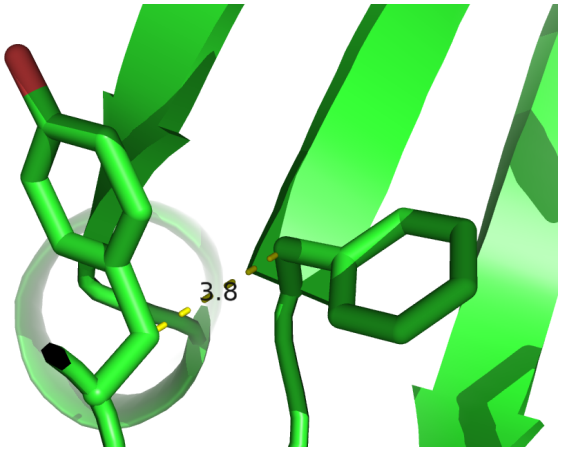

Figure E.5: The planar ring system of aromatic sidechains at short C_β-C_β distances (e.g. $\Delta C_\beta < 5\mathring{A}$) often points away from each other to avoid steric hindrance.

F

Optimizing Full Likelihood with Gradient Descent

F.1 Visualisation of learning rate schedules

F.2 Benchmarking learning rate schedules

F.2.1 Linear learning rate schedule

F.2.2 Sigmoidal learning rate schedule

F.2.3 Square root learning rate schedule

F.2.4 Exponential learning rate schedule

F.3 Number of iterations until convergence for different learning rate schedules

F.3.1 Linear learning rate schedule

(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 **linear** learning rate schedule $\alpha = \alpha_0/(1 + \gamma \cdot t)$ with t being the iteration number and the decay rate γ as specified in the legend. Initial learning rate α_0 defined with respect to Neff and maximum number of iterations is set to 5000.

F.3.2 Sigmoidal learning rate schedule

F.3.3 Square root learning rate schedule

(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 $\alpha = \alpha_0/\sqrt{1 + \gamma t}$ with t being the iteration number and the decay rate γ as specified in the legend. Initial learning rate α_0 defined with respect to Neff and maximum number of iterations is set to 5000.

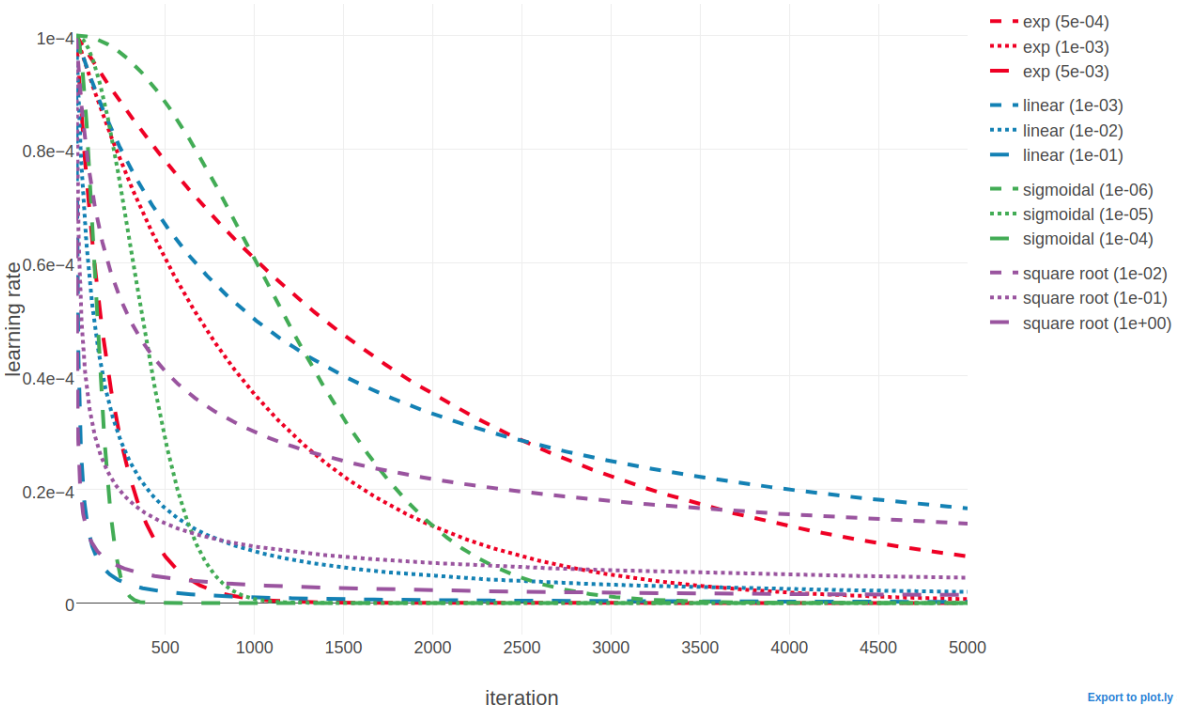

Figure F.1: Value of learning rate against the number of iterations for different learning rate schedules. Red legend group represents the **exponential** learning rate schedule $\alpha_{t+1} = \alpha_0 \cdot \exp(-\gamma t)$. Blue legend group represents the **linear** learning rate schedule $\alpha = \alpha_0/(1+\gamma \cdot t)$. Green legend group represents the **sigmoidal** learning rate schedule $\alpha_{t+1} = \alpha_t/(1+\gamma \cdot t)$ with γ. Purple legend group 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.

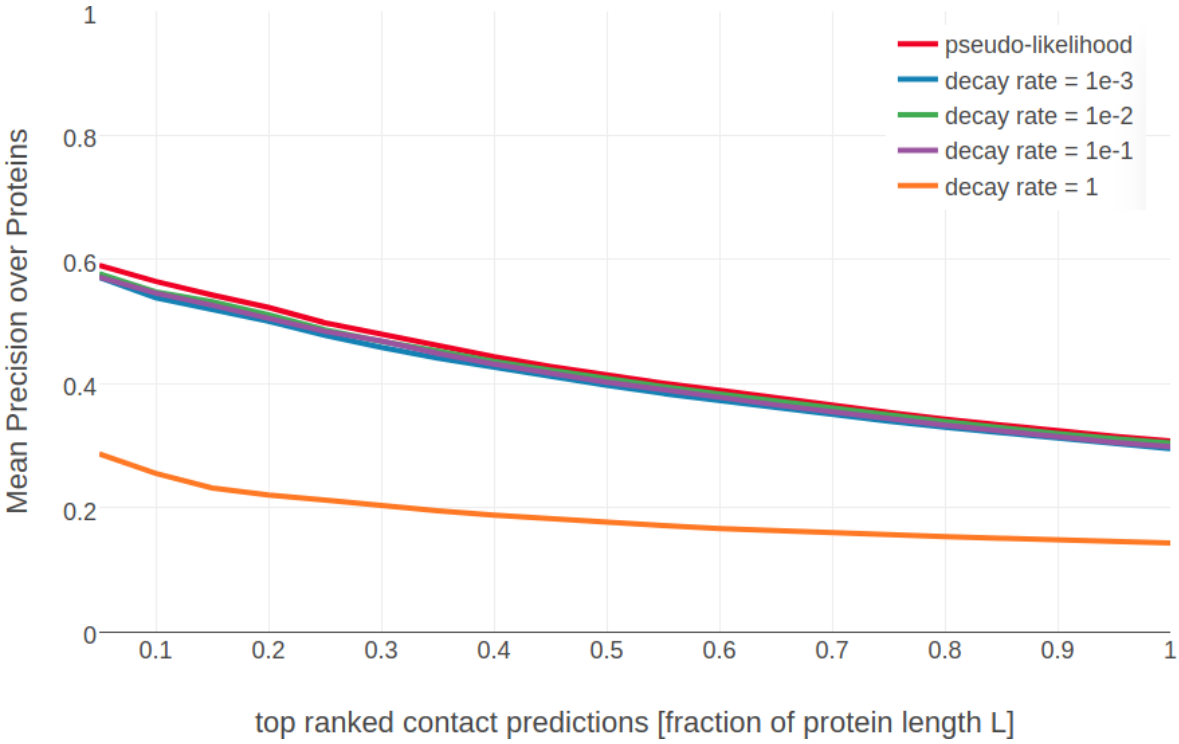

Figure F.2: Mean precision for top ranked contact predictions over 288 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. pseudo-likelihood: Contact scores computed from pseudo-likelihood. The other methods derive contact scores from couplings computed from CD using stochastic gradient descent with an initial learning rate defined with respect to Neff and a *linear* learning rate annealing schedule $\alpha = \frac{\alpha_0}{1+\gamma t}$ with decay rate γ as specified in the legend.

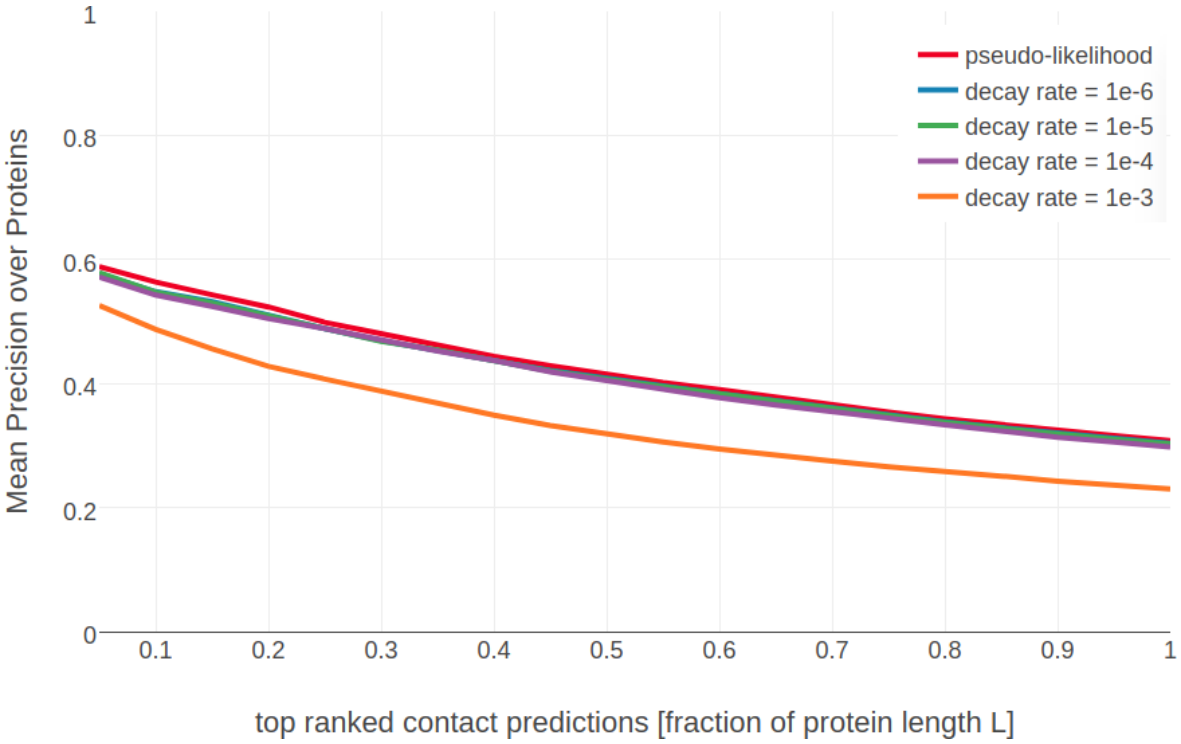

Figure F.3: Mean precision for top ranked contact predictions over 288 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. pseudo-likelihood: Contact scores computed from pseudo-likelihood. The other methods derive contact scores from couplings computed from CD using stochastic gradient descent with an initial learning rate defined with respect to Neff and a *sigmoidal* learning rate annealing schedule $\alpha_{t+1} = \frac{\alpha_t}{1+\gamma t}$ with t being the iteration number and decay rate γ as specified in the legend.

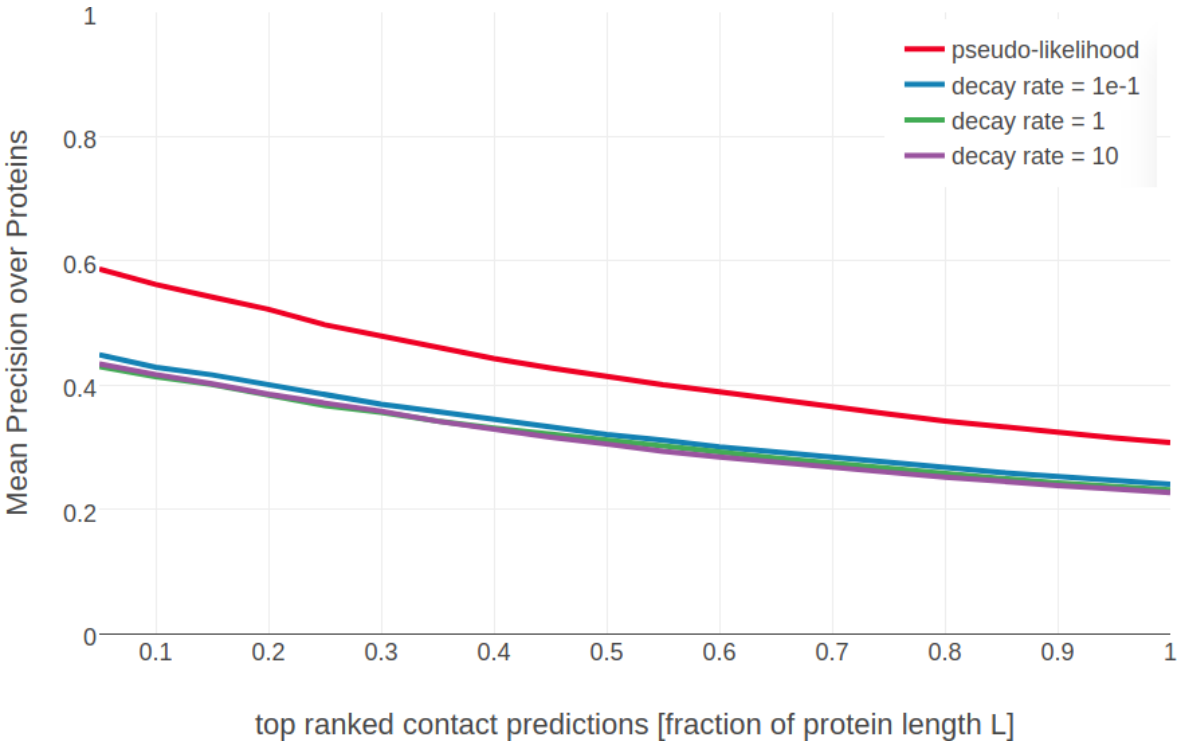

Figure F.4: Mean precision for top ranked contact predictions over 288 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. pseudo-likelihood: Contact scores computed from pseudo-likelihood. The other methods derive contact scores from couplings computed from CD using stochastic gradient descent with an initial learning rate defined with respect to Neff and a *square root* learning rate annealing schedule $\alpha = \frac{\alpha_0}{\sqrt{1+\gamma t}}$ with t being the iteration number and decay rate γ as specified in the legend.

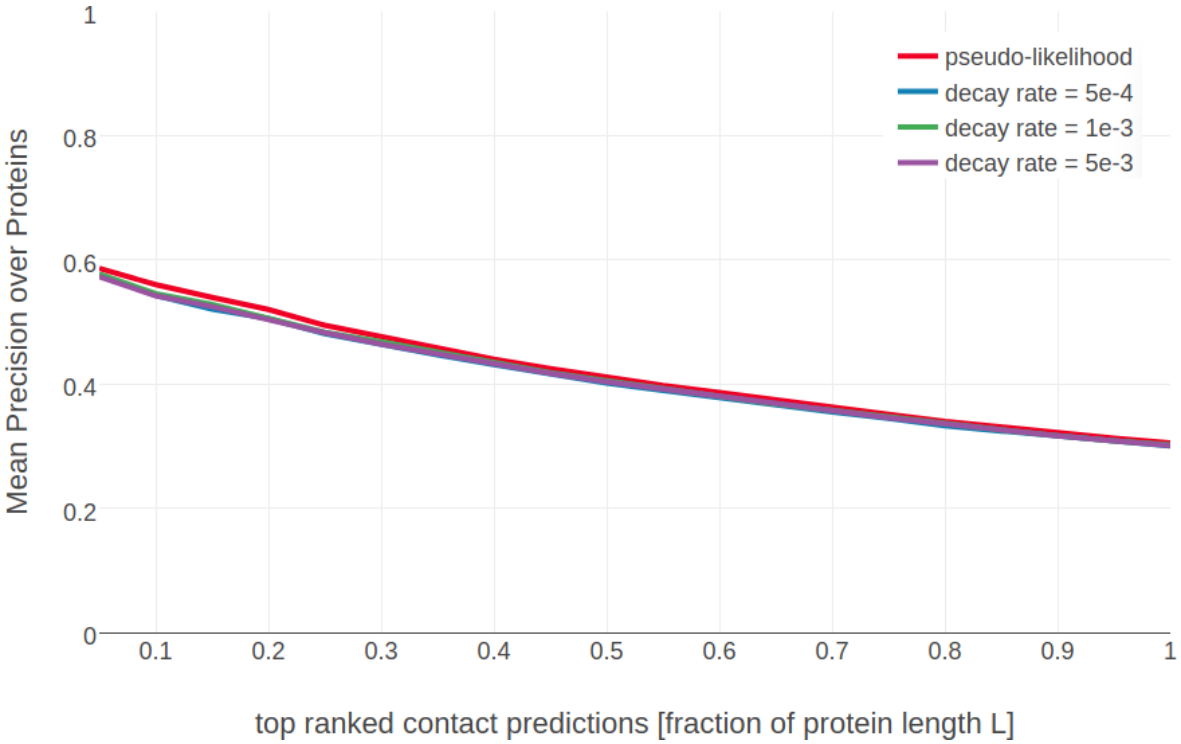

Figure F.5: Mean precision for top ranked contact predictions over 288 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. pseudo-likelihood: Contact scores computed from pseudo-likelihood. The other methods derive contact scores from couplings computed from CD using stochastic gradient descent with an initial learning rate defined with respect to Neff and a *exponential* learning rate annealing schedule $\alpha = \alpha_0 \cdot \exp(-\gamma t)$ with t being the iteration number and decay rate γ as specified in the legend.

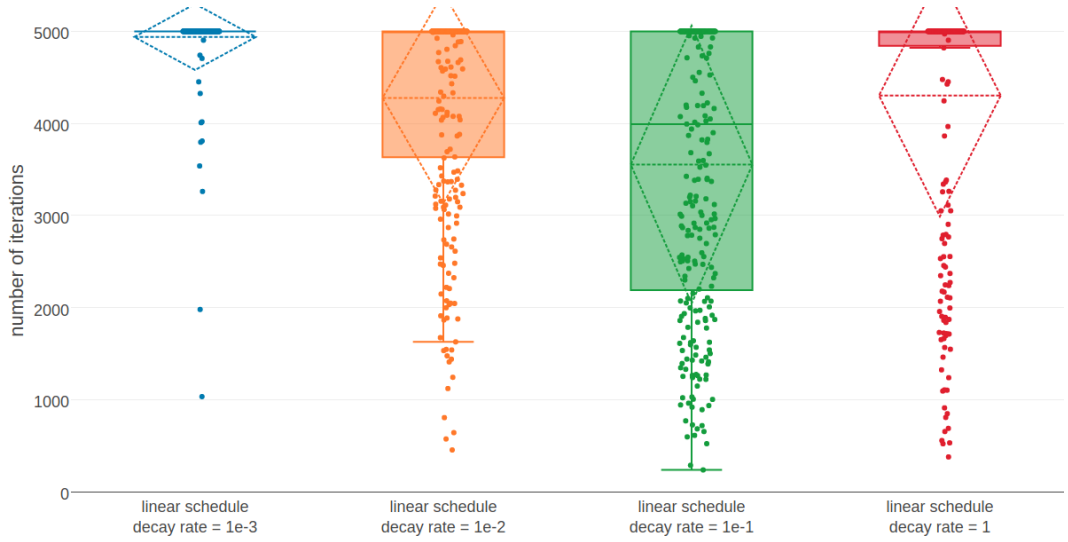

Figure F.6: (ref:caption-full-likelihood-opt-numit-lin-learning-rate-schedule)

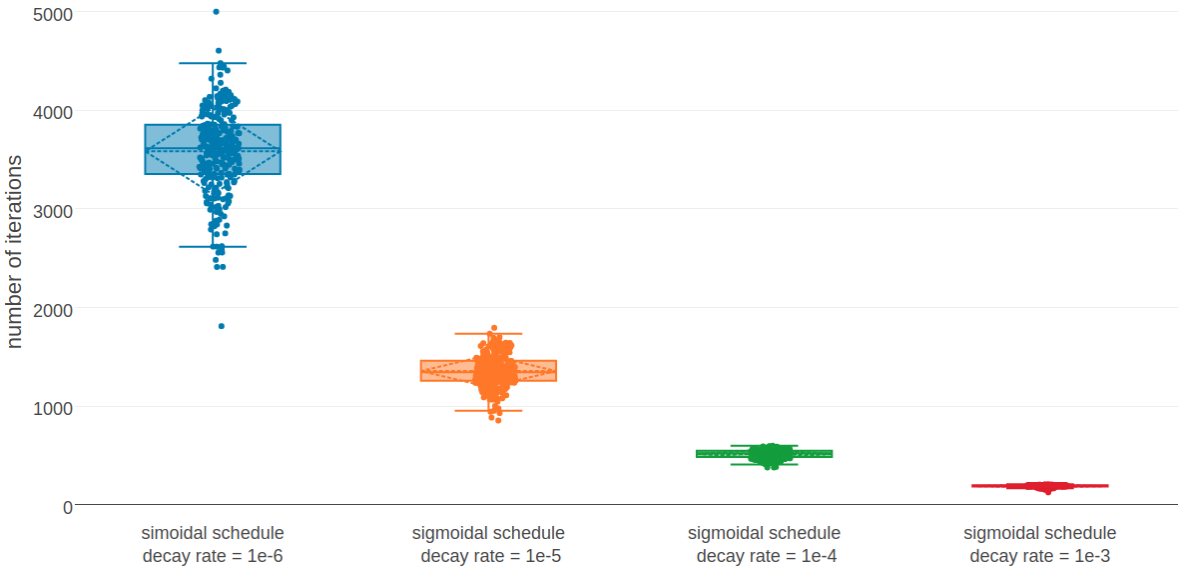

Figure F.7: Distribution of the number of iterations until convergence for gradient descent optimizations of the full likelihood using different decay rates with a **sigmoidal** learning rate schedule $\alpha_{t+1} = \alpha_t/(1 + \gamma \cdot t)$ with t being the iteration number and the decay rate γ as specified in the legend. Initial learning rate α_0 defined with respect to Neff and maximum number of iterations is set to 5000.

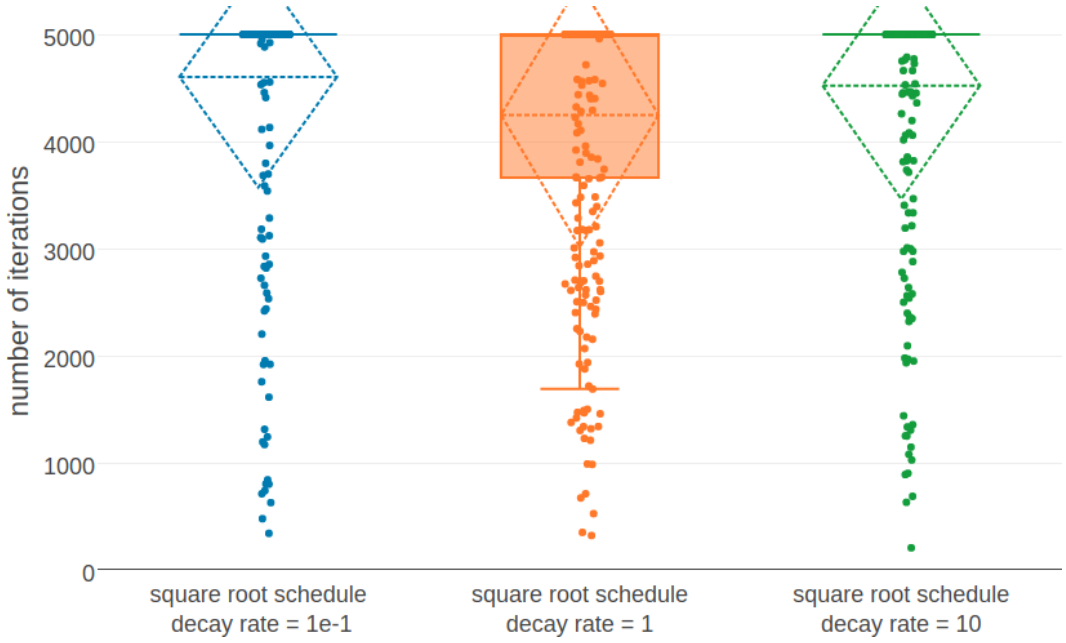

Figure F.8: (ref:caption-full-likelihood-opt-numit-sqrt-learning-rate-schedule)

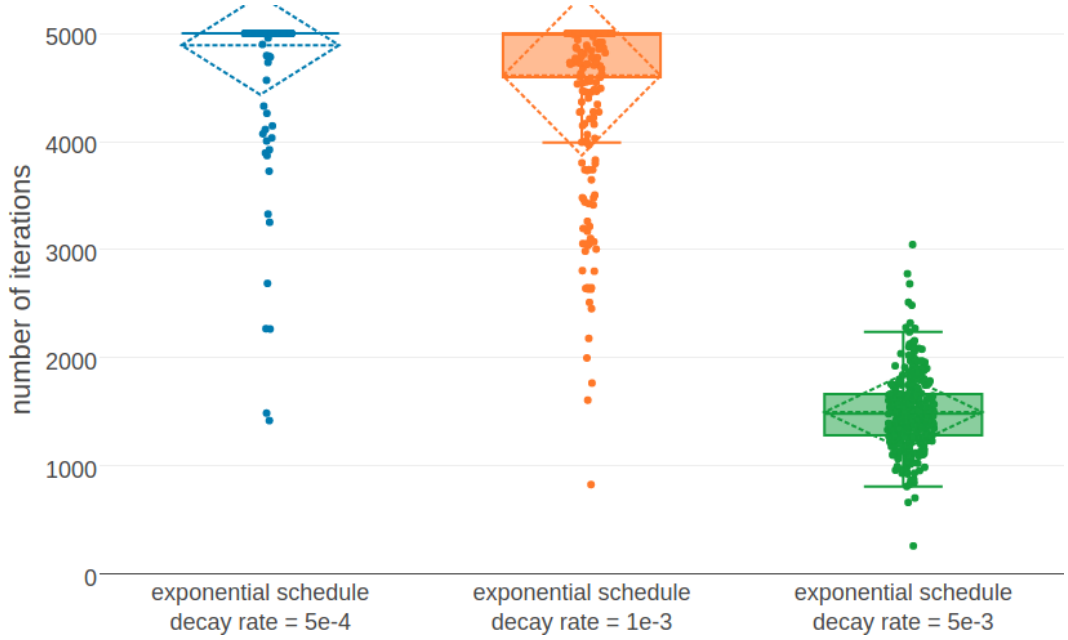

Figure F.9: (ref:caption-full-likelihood-opt-numit-exp-learning-rate-schedule)

F.3.4 Exponential learning rate schedule

(ref:caption-full-likelihood-opt-numit-exp-learning-rate-schedule) Distribution of the number of iterations until convergence for gradient descent optimizations of the full likelihood using different decay rates with an **exponential** learning rate schedule $\alpha = \alpha_0 \cdot \exp(-\gamma t)$ with t being the iteration number and the decay rate γ as specified in the legend. Initial learning rate α_0 defined with respect to Neff and maximum number of iterations is set to 5000.

F.4 Modifying Number of Iterations over which Relative Change of Coupling Norm is Evaluated

F.5 Number of Gibbs steps with respect to Neff

F.6 Fix single potentials at maximum-likelihood estimate v*

F.7 Monitoring Optimization for Different Sample Sizes

F.8 Monitoring Norm of Gradients for Different Number of Gibbs Steps

F.9 Statistics for Comparing Couplings computed with Pseudo-likelihood and Contrastive Divergence

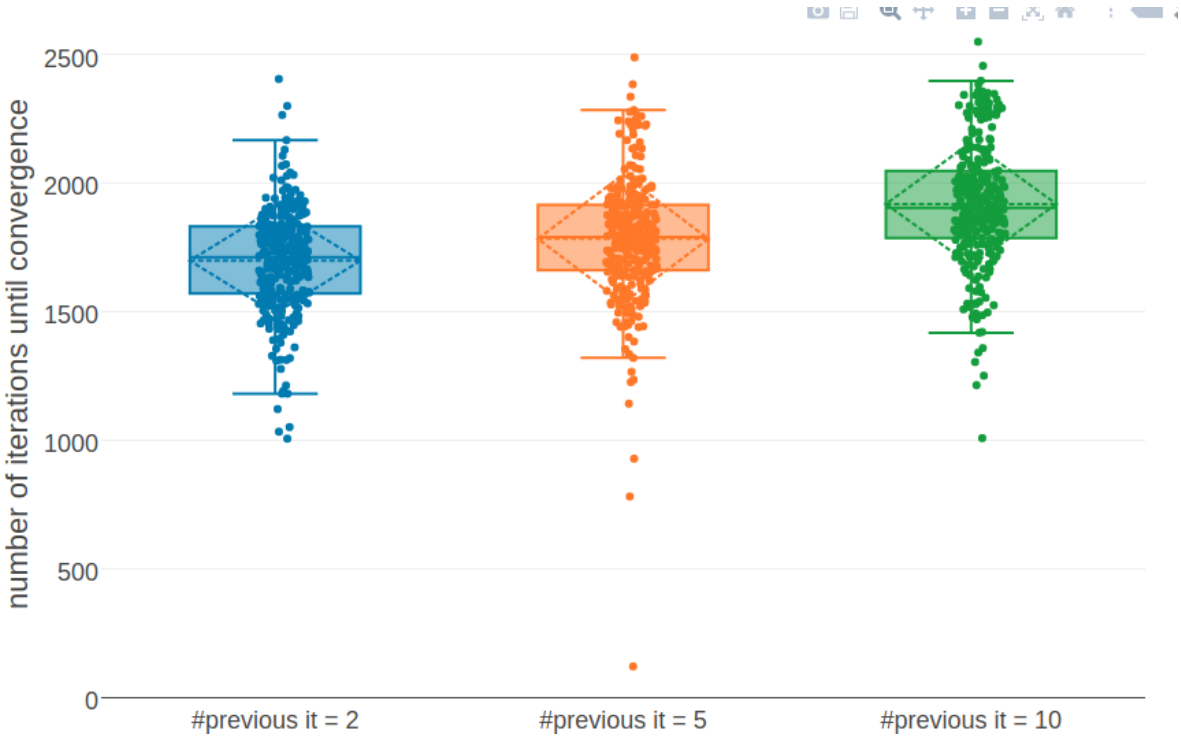

Figure F.10: Distribution of the number of iterations until convergence for gradient descent optimizations of the full likelihood using different values for the number of previous iterations over which the relative change of the L2 norm of coupling parameters $||\mathbf{w}||_2$ will be measured. Convergence criteria is given in eq. (4.3). The number of previous iterations is given on the x-axis in the plot. The optimal hyperparameters settings for SGD as described in section 4.2.2 have been used.

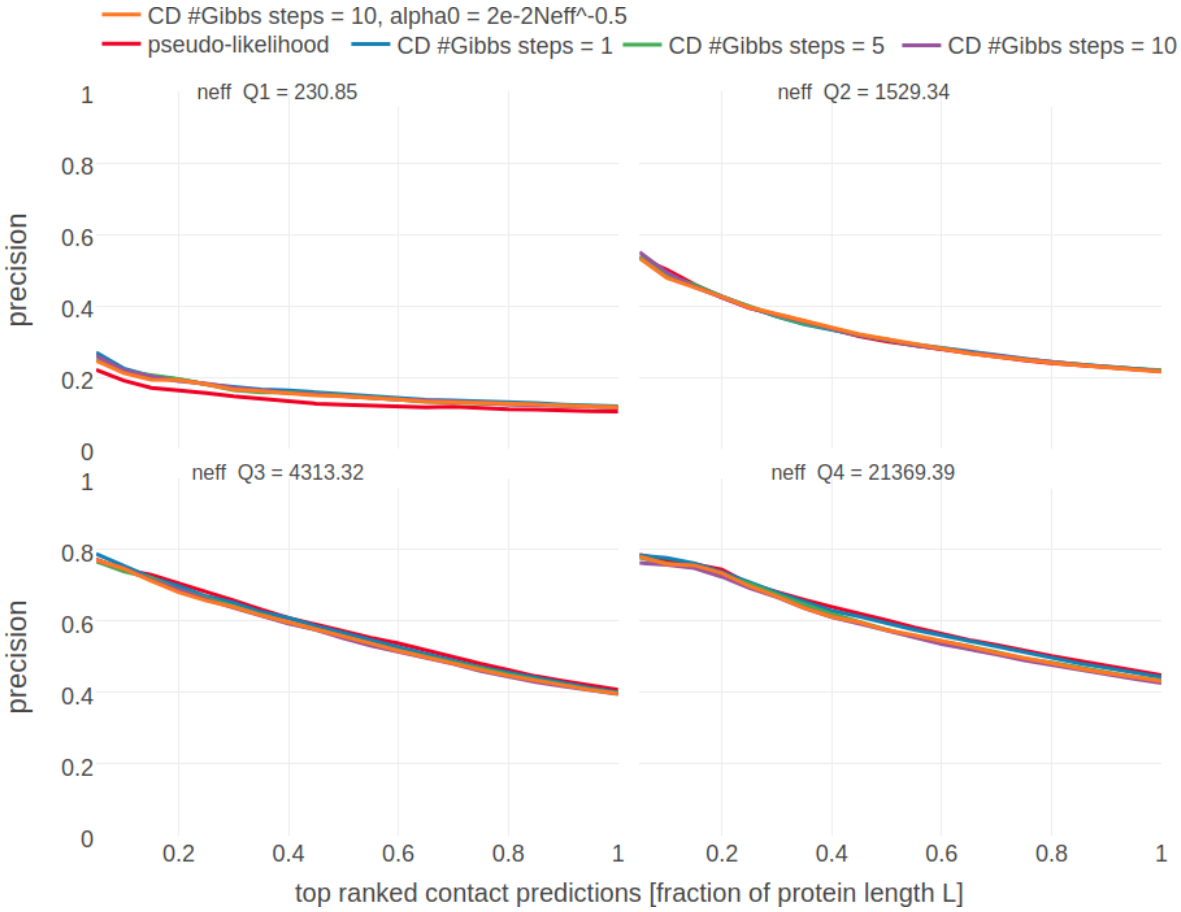

Figure F.11: Mean precision for top ranked contact predictions over 300 proteins splitted into four equally sized subsets with respect to Neff. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. Subsets are defined according to quantiles of Neff values. 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. **pseudo-likelihood**: contact scores computed from pseudo-likelihood. **CD #Gibbs steps = X**: contact scores computed from CD optimized with SGD and evolving each Markov chain using the number of Gibbs steps specified in the legend.

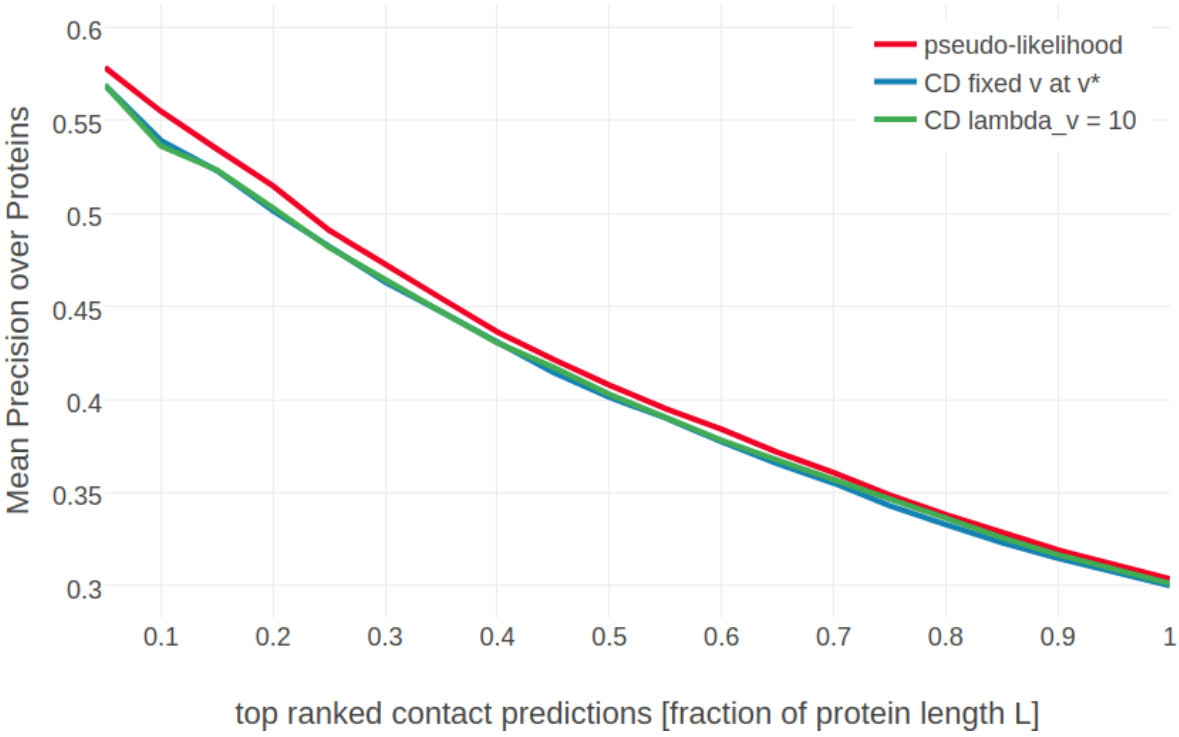

Figure F.12: Mean precision for top ranked contact predictions over 300 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. SGD settings for CD optimization are as follows: sigmoidal learning rate schedule with decay rate $\gamma = 5\mathrm{e}-6$ and initial learning rate $\alpha_0 = 5\mathrm{e}-2/N_{\mathrm{eff}}$. **pseudo-likelihood**: contact scores computed from pseudo-likelihood. **CD fixed v at v**: contact scores computed from CD with SGD and single potentials $\mathbf{v}$ are not optimized but fixed at $\mathbf{v}^*$ as given in eq. (7.28). **CD lambda_v = 10**: contact scores computed from CD with SGD and single potentials $\mathbf{v}$ are subject to optimization using L2-reglarization with $\lambda_v = 10$.

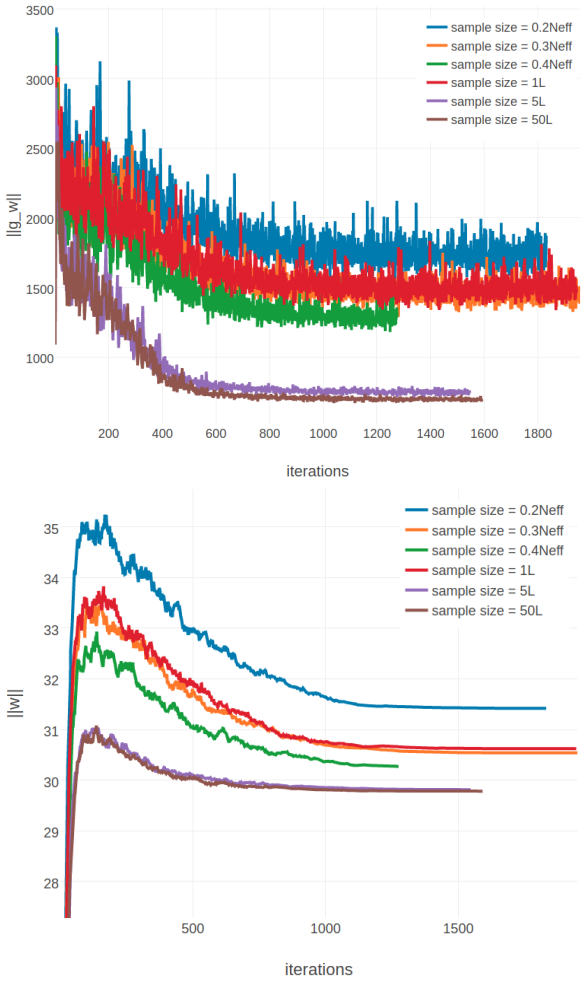

Figure F.13: Monitoring parameter norm and gradient norm for protein 1aho_A_00 during SGD using different sample sizes. Protein 1aho_A_00_00 has length L=64 and 378 sequences in the alignment (Neff=229) **Left** L2-norm of the gradients for coupling parameters, $||\nabla_{\mathbf{w}} LL(\mathbf{v}^*, \mathbf{w})||_2$ (without contribution of regularizer). The number of sequences, that is used for Gibbs sampling to approximate the gradient, is given in the legend. **Right** L2-norm of the coupling parameters $||\mathbf{w}||_2$. The number of sequences, that is used for Gibbs sampling to approximate the gradient, is given in the legend.

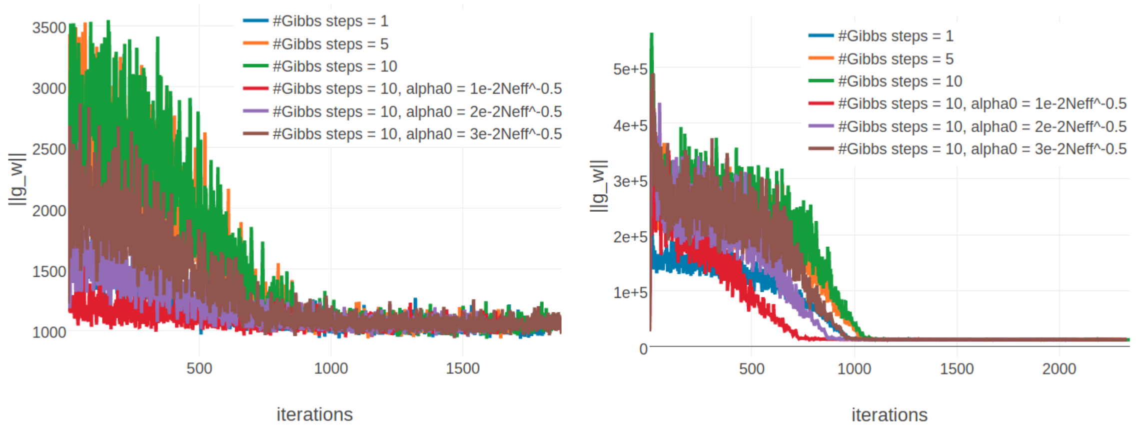

Figure F.14: Monitoring gradient norm, $||\nabla_{\mathbf{w}} LL(\mathbf{v}^*, \mathbf{w})||_2$, for protein 1aho_A_00 and 1c75_A_00 during SGD optimization using different number of Gibbs steps and initial learning rates, α_0. Number of Gibbs steps is given in the legend, as well as particular choices for the initial learning rate, when not using the default $\alpha_0 = \frac{5e-2}{\sqrt{N_{\text{eff}}}}$. **Left** Protein 1aho_A_00 has length L=64 and 378 sequences in the alignment (Neff=229) **Right** Protein 1c75_A_00 has length L=71 and 28078 sequences in the alignment (Neff=16808).

L	Neff	kendall[illegible]ov_pvalue
321	20.0	0.7257[illegible]2017e-275
213	205.360	[illegible]73833e-48
155	1266.64	[illegible]2757e-45
83	1987.35	[illegible]52393
156	2750.04	[illegible]70778e-12
98	21369.3	[illegible]11975
182	19856.6	[illegible]16747e-32
91	536.750	[illegible]0558e-05
108	74.8700	[illegible]02517e-31
150	702.650	[illegible]58639e-16
80	164.800	[illegible]7486e-08
148	6326.65	[illegible]13e-08
40	3451.28	[illegible]62982 305

G

Training of the Random Forest Contact Prior

G.1 Training Random Forest Model with pseudo-likelihood Feature

G.2 Evaluating window size with 5-fold Cross-validation

G.3 Evaluating non-contact threshold with 5-fold Cross-validation

G.4 Evaluating ratio of non-contacts and contacts in the training set with 5-fold Cross-validation

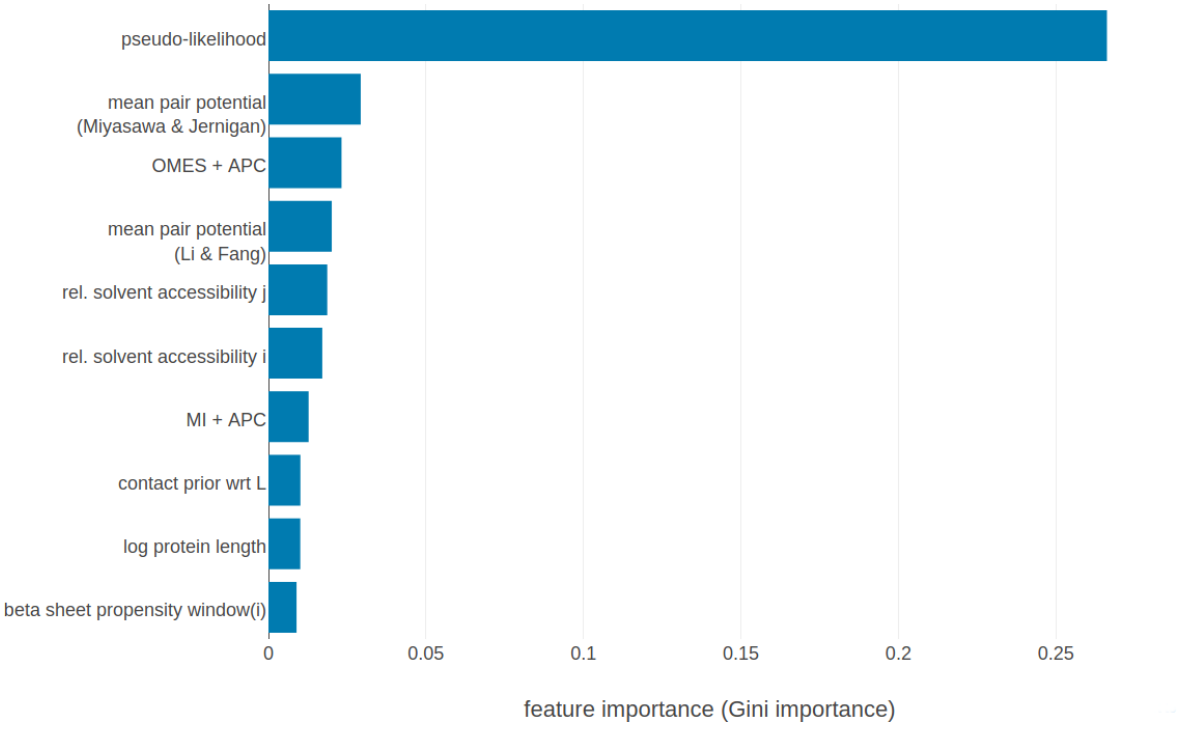

Figure G.1: Top ten features ranked according to *Gini importance*. **pseudo-likelihood**: APC corrected Frobenius norm of couplings computed with pseudo-likelihood. **mean pair potential (Miyasawa & Jernigan)**: average quasi-chemical energy of transfer of amino acids from water to the protein environment [215]. **OMES+APC**: APC corrected OMES score according to Fodor&Aldrich [214]. **mean pair potential (Li&Fang)**: average general contact potential by Li & Fang [69]. **rel. solvent accessibilty i(j)**: RSA score computed with Netsurfp (v1.0) [216] for position i(j). **MI+APC**: APC corrected mutual information between amino acid counts (using pseudo-counts). **contact prior wrt L**: simple contact prior based on expected number of contacts wrt protein length (see methods section **??**). **log protein length**: logarithm of protein length. **beta sheet propensity window(i)**: beta-sheet propensity according to Psipred [217] computed within a window of five positions around i. Features are described in detail in methods section 7.15.

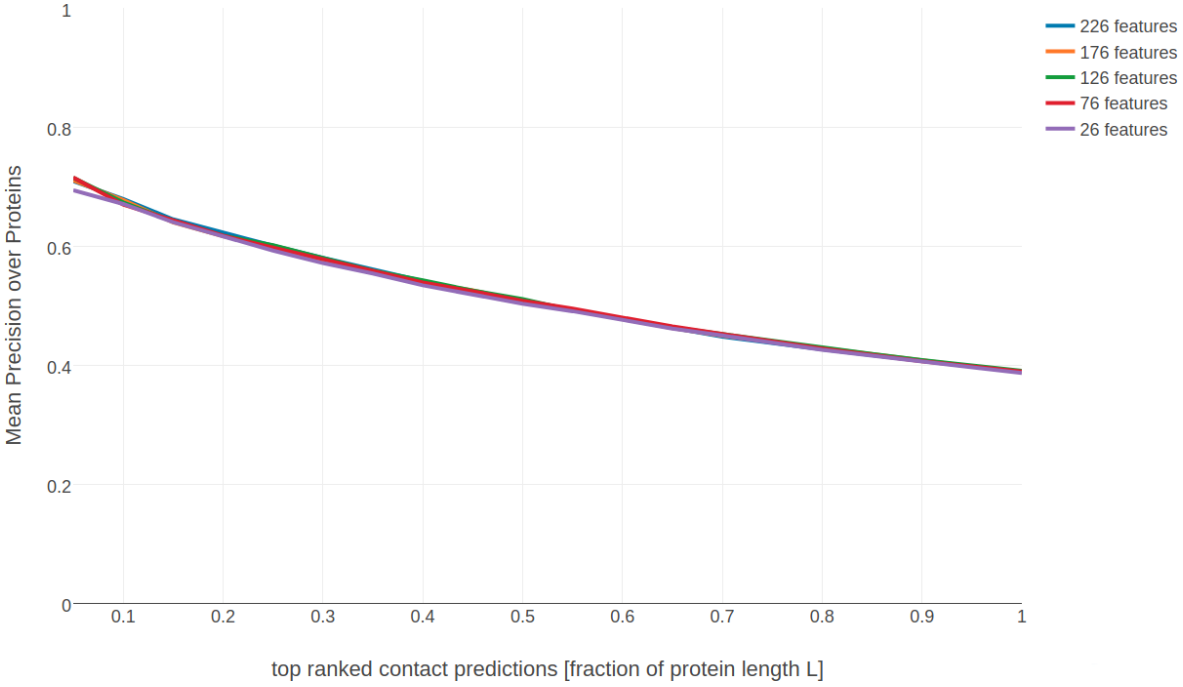

Figure G.2: Mean precision for top ranked contacts over 200 proteins for variaous random forest models trained on subsets of features. Subsets of features have been selected as described in section 7.16.1.

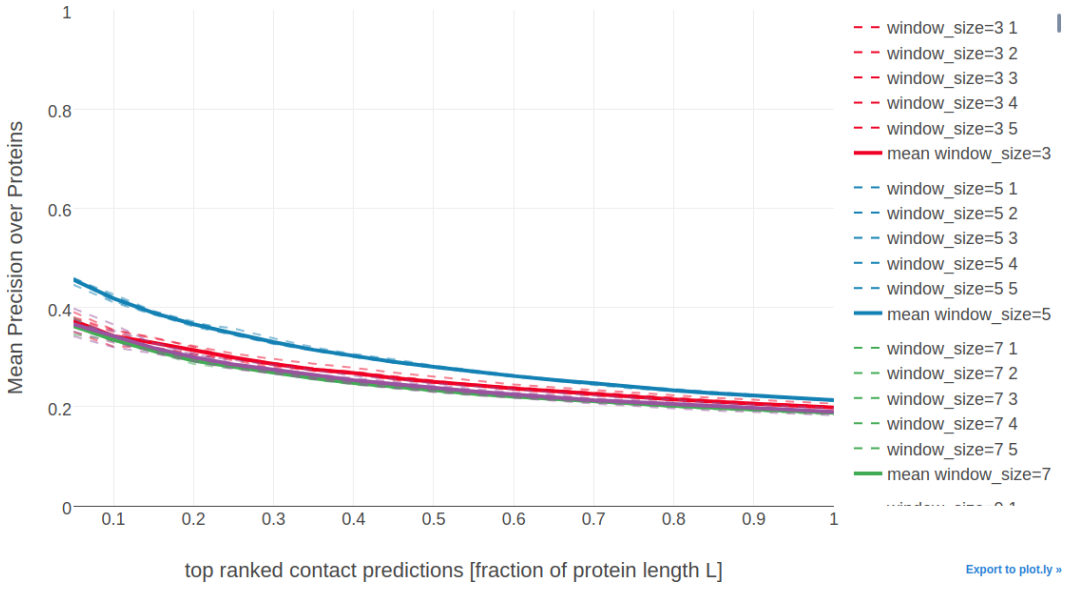

Figure G.3: Mean precision over validation set of 200 proteins for top ranked contact predictions for different choices of window size for single position features. Dashed lines represent the models trained on four subsets of the training data according to the 5-fold cross-validation scheme. Solid lines represent the mean over the five cross-validation models.

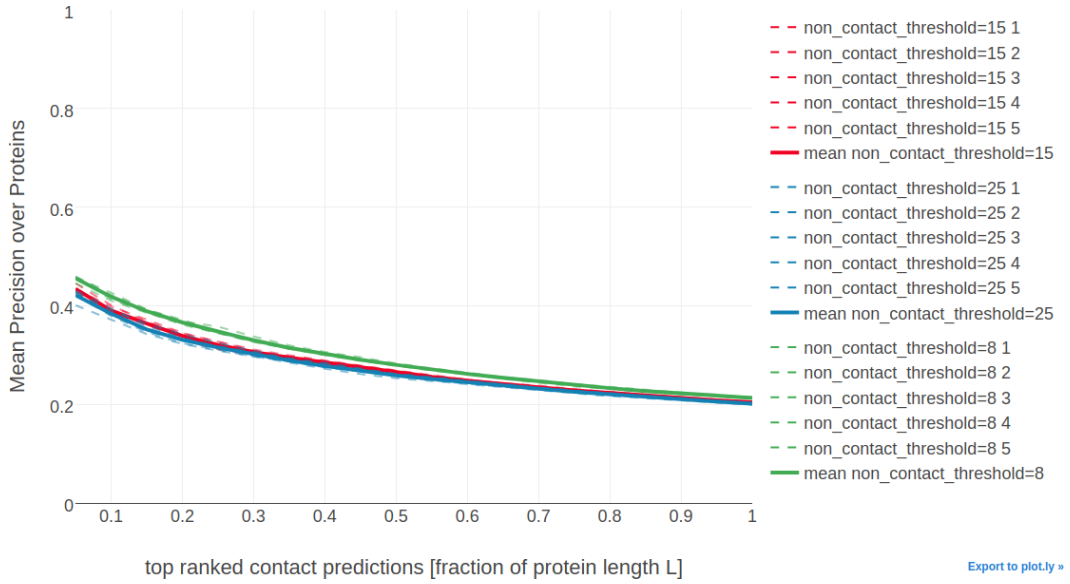

Figure G.4: Mean precision over validation set of 200 proteins for top ranked contact predictions for different choices of the non-contact threshold to define non-contacts. Dashed lines represent the models trained on four subsets of the training data according to the 5-fold cross-validation scheme. Solid lines represent the mean over the five cross-validation models.

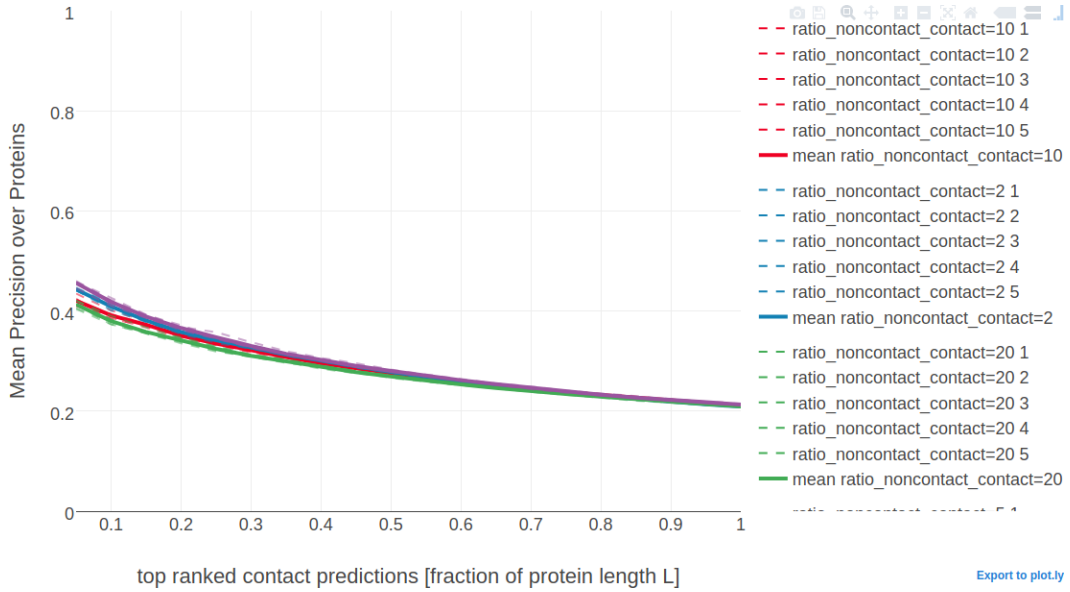

Figure G.5: Mean precision over validation set of 200 proteins for top ranked contact predictions for different choices of dataset composition with respect to the ratio of contacts and non-contacts. Dashed lines represent the models trained on four subsets of the training data according to the 5-fold 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/S00222836999

3. Fraser, P.E. (2014). Prions and prion-like proteins. J. Biol. Chem. *289*, 19839–40. Available at: http://www.ncbi.nlm.nih.gov/pubmed/24860092 http://www.pubmedcentral.nih.gov/articlerender.

4. 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/31/1/10.1
r8gIyHaHX5ANqUoy6w0PUuete2b{~}5ZU{~}{~}D6KOB1vq5A8MgCnrq3pDHUn0OSgz0QFmtU2RYf907-
.

5. Schwede, T. (2013). Protein modeling: what happened to the "protein structure gap"? Structure *21*, 1531–40. Available at: http://www.ncbi.nlm.nih.gov/pubmed/24010712 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3816506.

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

7. 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.

8. 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.

9. 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?artid=PMC1166865.

10. 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.

11. 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.

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

13. The UniProt Consortium (2017). UniProt: the universal protein knowledgebase.

Nucleic Acids Res. *45*, D158–D169. Available at: https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkw1099.

14. Carpenter, E.P., Beis, K., Cameron, A.D., and Iwata, S. (2008). Overcoming the challenges of membrane protein crystallography. Curr. Opin. Struct. Biol. *18*, 581–6. Available at: http://www.ncbi.nlm.nih.gov/pubmed/18674618 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2580798.

15. Moraes, I., Evans, G., Sanchez-Weatherby, J., Newstead, S., and Stewart, P.D.S. (2014). Membrane protein structure determination - the next generation. Biochim. Biophys. Acta *1838*, 78–87. Available at: http://www.ncbi.nlm.nih.gov/pubmed/23860256 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3898769.

16. Jacobson, M.P., Friesner, R.A., Xiang, Z., and Honig, B. (2002). On the Role of the Crystal Environment in Determining Protein Side-chain Conformations. J. Mol. Biol. *320*, 597–608. Available at: http://www.sciencedirect.com/science/article/pii/S0022283602004709?via{\%}3Dihub.

17. Bieri, M., Kwan, A.H., Mobli, M., King, G.F., Mackay, J.P., and Gooley, P.R. (2011). Macromolecular NMR spectroscopy for the non-spectroscopist: beyond macromolecular solution structure determination. FEBS J. *278*, 704–715. Available at: http://www.ncbi.nlm.nih.gov/pubmed/21214861 http://doi.wiley.com/10.1111/j.1742-4658.2011.08005.x.

18. Billeter, M., Wagner, G., and Wthrich, K. (2008). Solution NMR structure determination of proteins revisited. J. Biomol. NMR *42*, 155–8. Available at: http://www.ncbi.nlm.nih.gov/pubmed/18827972 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=

19. 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.

20. Fernandez-Leiro, R., and Scheres, S.H.W. (2016). Unravelling biological macromolecules with cryo-electron microscopy. Nature *537*, 339–46. Available at: http://www.ncbi.nlm.nih.gov/pubmed/27629 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC5074357.

21. Reuter, J.A., Spacek, D.V., and Snyder, M.P. (2015). High-throughput sequencing technologies. Mol. Cell *58*, 586–97. Available at: http://www.ncbi.nlm.nih.gov/pubmed/26000844 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4494749.

22. Goodwin, S., McPherson, J.D., and McCombie, W.R. (2016). Coming of age: ten years of next-generation sequencing technologies. Nat. Rev. Genet. *17*, 333–351. Available at: http://www.nature.com/doifinder/10.1038/nrg.2016.49.

23. NovaSeq System Specifications | The next era of sequencing starts now Available at: https://www.illumina.com/systems/sequencing-platforms/novaseq/specifications.html [Accessed October 18, 2017].

24. Tringe, S.G., and Rubin, E.M. (2005). Metagenomics: DNA sequencing of environmental samples. Nat. Rev. Genet. *6*, 805–814. Available at: http://www.nature.com/doifinder/10.1038/nrg1709.

25. Hugenholtz, P., and Tyson, G.W. (2008). Microbiology: Metagenomics. Nature *455*, 481–483. Available at: http://www.nature.com/doifinder/10.1038/455481a.

26. Wooley, J.C., Godzik, A., and Friedberg, I. (2010). A primer on metagenomics. PLoS Comput. Biol. *6*, e1000667. Available at: http://www.ncbi.nlm.nih.gov/pubmed/20195499 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2829047.

27. Rinke, C., Schwientek, P., Sczyrba, A., Ivanova, N.N., Anderson, I.J., Cheng, J.-F., Darling, A., Malfatti, S., Swan, B.K., and Gies, E.A. *et al.* (2013). Insights into the phylogeny

and coding potential of microbial dark matter. Nature *499*, 431–437. Available at: http://www.nature.com/doifinder/10.1038/nature12352.

28. Mukherjee, S., Seshadri, R., Varghese, N.J., Eloe-Fadrosh, E.A., Meier-Kolthoff, J.P., Gker, M., Coates, R.C., Hadjithomas, M., Pavlopoulos, G.A., and Paez-Espino, D. *et al.* (2017). 1,003 reference genomes of bacterial and archaeal isolates expand coverage of the tree of life. Nat. Biotechnol. *35*, 676–683. Available at: http://www.nature.com/doifinder/10.1038/nbt.3886.

29. Forster, S.C. (2017). Illuminating microbial diversity. Nat. Rev. Microbiol. *15*, 578–578. Available at: http://www.nature.com/doifinder/10.1038/nrmicro.2017.106.

30. Clabbers, M.T.B., Genderen, E. van, Wan, W., Wiegers, E.L., Gruene, T., and Abrahams, J.P. (2017). Protein structure determination by electron diffraction using a single three-dimensional nanocrystal. Acta Crystallogr. Sect. D, Struct. Biol. *73*, 738–748. Available at: http://www.ncbi.nlm.nih.gov/pubmed/28876237 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC5586247.

31. Dukka, B.K. (2016). Recent advances in sequence-based protein structure prediction. Brief. Bioinform. *31*, 1–12. Available at: https://academic.oup.com/bib/article-lookup/doi/10.1093/bib/bbw070.

32. Li, W., Zhang, Y., and Skolnick, J. (2004). Application of sparse NMR restraints to large-scale protein structure prediction. Biophys. J. *87*, 1241–8. Available at: http://www.ncbi.nlm.nih.gov/pubmed/15298926 http://www.pubmedcentral.nih.gov/articlerender.fcgi?arti

33. Walzthoeni, T., Leitner, A., Stengel, F., and Aebersold, R. (2013). Mass spectrometry supported determination of protein complex structure. Curr. Opin. Struct. Biol. *23*, 252–260. Available at: http://www.ncbi.nlm.nih.gov/pubmed/23522702 http://linkinghub.elsevier.com/retrieve/pii/S0959440X13000377.

34. Rappsilber, J. (2011). The beginning of a beautiful friendship: cross-linking/mass spectrometry and modelling of proteins and multi-protein complexes. J. Struct. Biol. *173*, 530–40. Available at: http://www.ncbi.nlm.nih.gov/pubmed/21029779 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3043253.

35. Ornes, S. (2016). Let the structural symphony begin. Nature *536*, 361–363. Available at: http://www.nature.com/doifinder/10.1038/536361a.

36. Ward, A.B., Sali, A., and Wilson, I.A. (2013). Biochemistry. Integrative structural biology. Science *339*, 913–5. Available at: http://www.ncbi.nlm.nih.gov/pubmed/23430643 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3633482.

37. Tang, Y., Huang, Y.J., Hopf, T.A., Sander, C., Marks, D.S., and Montelione, G.T. (2015). Protein structure determination by combining sparse NMR data with evolutionary couplings. Nat. Methods *advance on*. Available at: http://dx.doi.org/10.1038/nmeth.3455.

38. 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.

39. Vendruscolo, M., Kussell, E., and Domany, E. (1997). Recovery of protein structure from contact maps. Fold. Des. *2*, 295–306. Available at: http://dx.doi.org/10.1016/S1359-0278(97)00041-2.

40. Kim, D.E., Dimaio, F., Yu-Ruei Wang, R., Song, Y., and Baker, D. (2014). One contact for every twelve residues allows robust and accurate topology-level protein structure

modeling. Proteins *82 Suppl 2*, 208–18. Available at: http://www.ncbi.nlm.nih.gov/pubmed/23900763.

41. 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.

42. Sadowski, M.I. (2013). Prediction of protein domain boundaries from inverse covariances. Proteins *81*, 253–60. Available at: http://www.ncbi.nlm.nih.gov/pubmed/22987736 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3563215.

43. Parisi, G., Zea, D.J., Monzon, A.M., and Marino-Buslje, C. (2015). Conformational diversity and the emergence of sequence signatures during evolution. Curr. Opin. Struct. Biol. *32C*, 58–65. Available at: http://www.sciencedirect.com/science/article/pii/S0959440X15000147.

44. Hopf, T.A., Ingraham, J.B., Poelwijk, F.J., Schrfe, C.P.I., Springer, M., Sander, C., and Marks, D.S. (2017). Mutation effects predicted from sequence co-variation. Nat. Biotechnol. *35*, 128–135. Available at: http://www.ncbi.nlm.nih.gov/pubmed/28092658 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC5383098 http://www.nature.com/doifinder/

45. 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.

46. 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.

47. 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{\&}rendertype=abst

48. 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.

49. 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.

50. 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.

51. 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?artid=2143025{\&}to

52. 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=6e1c9cbdef66bd0beefe

53. 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.

54. 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=2a1f0a044a8fd2213955e4f2c17fa84682dd25

55. 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.

56. 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.abstract?ijkey=d19c5a2492bfa46d

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

58. 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.

59. 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.

60. 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.

61. 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.

62. 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.

63. 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.

64. 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.

65. 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.

66. 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.

67. Cheng, J., and Baldi, P. (2007). Improved residue contact prediction using support vector machines and a large feature set. BMC Bioinformatics *8*, 113. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1852326{\&}tool=pmcentrez{\&}rendertype=

68. 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=pmcentrez{\&}rendertype=

69. 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.

70. 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=pmcentrez{\&}rendertype=ab

71. 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=pmcentrez{\&}rendertype=ab

72. 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.

73. 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.

74. 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.

75. 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=pmcentrez{\&}rendertype=ab

76. 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{\&}tool=pmcentrez{\&}re

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

78. 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/28/19/2
14.

79. 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.

80. 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.

81. 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.

82. 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.

83. Schneider, M., and Brock, O. (2014). Combining Physicochemical and Evolu-

tionary Information for Protein Contact Prediction. PLoS One *9*, e108438. Available at: http://www.plosone.org/article/info{\%}3Adoi{\%}2F10.1371{\%}2Fjournal.pone.0108438{\#}abstract0.

84. 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.

85. 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/articlerender.fcgi?artid=PMC5249242.

86. 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.

87. 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.

88. Andreani, J., and Sding, J. (2015). Bbcontacts: Prediction of β-strand pairing from direct coupling patterns. Bioinformatics *31*, 1729–37. Available at: http://bioinformatics.oxfordjournals.org/content/31/11/1729.

89. Skwark, M.J., Raimondi, D., Michel, M., and Elofsson, A. (2014). Improved contact predictions using the recognition of protein like contact patterns. PLoS Comput. Biol. *10*, e1003889. Available at: http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1003889.

90. 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.

91. 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.

92. 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.

93. 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.

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

95. 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.

96. 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.

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

98. 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.

99. 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=pmcen

100. 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.

101. 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.

102. 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.f

103. Lapedes, A., Giraud, B., and Jarzynski, C. (2012). Using Sequence Alignments to Predict Protein Structure and Stability With High Accuracy. Available at: http://arxiv.org/abs/1207.2484.

104. 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.

105. 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.

106. 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.

107. 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.

108. 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.

109. 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{_}maximum{_}likelihood{_}and{_}pseudo-likelihood{_}estimators{_}for{_}Gibbs{_}Distributions.

110. 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.

111. Zhang, H., Gao, Y., Deng, M., Wang, C., Zhu, J., Li, S.C., Zheng, W.-M., and Bu, D. (2016). Improving residue-residue contact prediction via low-rank and sparse decomposition of residue correlation matrix. Biochem. Biophys. Res. Commun. Available at: http://www.sciencedirect.com/science/article/pii/S0006291X16301838.

112. Yu, X., Wu, X., Bermejo, G.A., Brooks, B.R., and Taraska, J.W. (2013). Ac-

curate high-throughput structure mapping and prediction with transition metal ion FRET. Structure *21*, 9–19. Available at: http://www.ncbi.nlm.nih.gov/pubmed/23273426 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3700372.

113. Kalinin, S., Peulen, T., Sindbert, S., Rothwell, P.J., Berger, S., Restle, T., Goody, R.S., Gohlke, H., and Seidel, C.A.M. (2012). A toolkit and benchmark study for FRET-restrained high-precision structural modeling. Nat. Methods *9*, 1218–1225. Available at: http://www.ncbi.nlm.nih.gov/pubmed/23142871 http://www.nature.com/doifinder/10.1038/nmeth.2222.

114. Bowers, P.M., Strauss, C.E., and Baker, D. (2000). De novo protein structure determination using sparse NMR data. J. Biomol. NMR *18*, 311–8. Available at: http://www.ncbi.nlm.nih.gov/pubmed/11200525.

115. Kolinski, A., and Skolnick, J. (1998). Assembly of protein structure from sparse experimental data: An efficient Monte Carlo model. Proteins Struct. Funct. Genet. *32*, 475–494. Available at: http://doi.wiley.com/10.1002/{\%}28SICI{\%}291097-0134{\%}2819980901{\%}2932{\%}3A4{\%}3C475{\%}3A{\%}3AAID-PROT6{\%}3E3.0.CO{\%}3B2-F.

116. Aszdi, A., Taylor, W.R., and Gradwell, M.J. (1995). Global Fold Determination from a Small Number of Distance Restraints. J. Mol. Biol. *251*, 308–326. Available at: http://www.sciencedirect.com/science/article/pii/S0022283685704366?via{\%}3Dihub.

117. Wu, S., Szilagyi, A., and Zhang, Y. (2011). Improving protein structure prediction using multiple sequence-based contact predictions. Structure *19*, 1182–91. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3154634{\&}tool=pmcentrez{\&}rendertype=

118. Tress, M.L., and Valencia, A. (2010). Predicted residue-residue contacts can help the scoring of 3D models. Proteins Struct. Funct. Bioinforma. *78*, NA–NA. Available at: http://www.ncbi.nlm.nih.gov/pubmed/20408174 http://doi.wiley.com/10.1002/prot.22714.

119. 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=3641781{\&}tool=pmc

120. Ovchinnikov, S., Kamisetty, H., and Baker, D. (2014). Robust and accurate prediction of residue-residue interactions across protein interfaces using evolutionary information. Elife *3*, e02030. Available at: http://elifesciences.org/content/3/e02030.abstract.

121. Hopf, T.A., Schrfe, C.P.I., Rodrigues, J.P.G.L.M., Green, A.G., Sander, C., Bonvin, A.M.J.J., and Marks, D.S. (2014). Sequence co-evolution gives 3D contacts and structures of protein complexes. Available at: http://arxiv.org/abs/1405.0929.

122. Hayat, S., Sander, C., Marks, D.S., and Elofsson, A. (2015). All-atom 3D structure prediction of transmembrane β-barrel proteins from sequences. Proc. Natl. Acad. Sci. U. S. A. *112*, 5413–8. Available at: http://www.pnas.org/content/112/17/5413.short.

123. Hopf, T.A., Morinaga, S., Ihara, S., Touhara, K., Marks, D.S., and Benton, R. (2015). Amino acid coevolution reveals three-dimensional structure and functional domains of insect odorant receptors. Nat. Commun. *6*, 6077. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4364406{\&}tool=pmcentrez{\&}rendertype=

124. Raval, A., Piana, S., Eastwood, M.P., and Shaw, D.E. (2015). Assessment of the utility of contact-based restraints in accelerating the prediction of protein structure using molecular dynamics simulations. Protein Sci. Available at: http://www.ncbi.nlm.nih.gov/pubmed/26266489.

125. 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.

126. Ovchinnikov, S., Kinch, L., Park, H., Liao, Y., Pei, J., Kim, D.E., Kamisetty, H., Grishin, N.V., and Baker, D. (2015). Large scale determination of previously unsolved protein structures using evolutionary information. Elife *4*, e09248. Available at: http://elifesciences.org/content/early/2015/09/03/eLife.09248.abstract.

127. Ovchinnikov, S., Park, H., Varghese, N., Huang, P.-S., Pavlopoulos, G.A., Kim, D.E., Kamisetty, H., Kyrpides, N.C., and Baker, D. (2017). Protein structure determination using metagenome sequence data. Science *355*, 294–298. Available at: http://www.ncbi.nlm.nih.gov/pubmed/28104891 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=

128. Bhattacharya, D., Cao, R., and Cheng, J. (2016). UniCon3D: de novo protein structure prediction using united-residue conformational search via stepwise, probabilistic sampling. Bioinformatics, btw316. Available at: http://www.ncbi.nlm.nih.gov/pubmed/27259540.

129. Braun, T., Koehler Leman, J., and Lange, O.F. (2015). Combining Evolutionary Information and an Iterative Sampling Strategy for Accurate Protein Structure Prediction. PLoS Comput. Biol. *11*, e1004661. Available at: http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004661.

130. Mabrouk, M., Putz, I., Werner, T., Schneider, M., Neeb, M., Bartels, P., and Brock, O. (2015). RBO Aleph: leveraging novel information sources for protein structure prediction. Nucleic Acids Res. *43*, W343–8. Available at: http://nar.oxfordjournals.org/content/43/W1/W343.

131. Pietal, M.J., Bujnicki, J.M., and Kozlowski, L.P. (2015). GDFuzz3D: a method for protein 3D structure reconstruction from contact maps, based on a non-Euclidean distance function. Bioinformatics, btv390. Available at: http://bioinformatics.oxfordjournals.org/content/early/2015/07/23/bioinformatics.btv390.

132. Michel, M., Hayat, S., Skwark, M.J., Sander, C., Marks, D.S., and Elofsson, A. (2014). PconsFold: improved contact predictions improve protein models. Bioinformatics *30*, i482–i488. Available at: http://bioinformatics.oxfordjournals.org/content/30/17/i482.long.

133. Konopka, B.M., Ciombor, M., Kurczynska, M., and Kotulska, M. (2014). Automated Procedure for Contact-Map-Based Protein Structure Reconstruction. J. Membr. Biol. Available at: http://www.ncbi.nlm.nih.gov/pubmed/24682239.

134. Kosciolek, T., and Jones, D.T. (2014). De novo structure prediction of globular proteins aided by sequence variation-derived contacts. PLoS One *9*, e92197. Available at: http://dx.plos.org/10.1371/journal.pone.0092197.

135. Nugent, T., and Jones, D.T. (2012). Accurate de novo structure prediction of large transmembrane protein domains using fragment-assembly and correlated mutation analysis. Proc. Natl. Acad. Sci. U. S. A. *109*, E1540–7. Available at: http://www.pnas.org/content/early/2012/05/21/1120036109.

136. Sathyapriya, R., Duarte, J.M., Stehr, H., Filippis, I., and Lappe, M. (2009). Defining an essence of structure determining residue contacts in proteins. PLoS Comput. Biol. *5*, e1000584. Available at: http://dx.plos.org/10.1371/journal.pcbi.1000584.

137. Chen, Y., Ding, F., and Dokholyan, N.V. (2007). Fidelity of the Protein Structure Reconstruction from Inter-Residue Proximity Constraints. J. Phys. Chem. B *111*, 7432–7438. Available at: http://www.ncbi.nlm.nih.gov/pubmed/17542631

http://pubs.acs.org/doi/abs/10.1021/jp068963t.

138. Vassura, M., Margara, L., Di Lena, P., Medri, F., Fariselli, P., and Casadio, R. (2007). Reconstruction of 3D structures from protein contact maps. IEEE/ACM Trans. Comput. Biol. Bioinform. *5*, 357–67. Available at: http://www.ncbi.nlm.nih.gov/pubmed/18670040.

139. Adhikari, B., Bhattacharya, D., Cao, R., and Cheng, J. (2017). Assessing Predicted Contacts for Building Protein Three-Dimensional Models. Methods Mol. Biol. *1484*, 115–126. Available at: http://link.springer.com/10.1007/978-1-4939-6406-2{_}9.

140. Di Lena, P., Vassura, M., Margara, L., Fariselli, P., and Casadio, R. (2009). On the Reconstruction of Three-dimensional Protein Structures from Contact Maps. Algorithms *2*, 76–92. Available at: http://www.mdpi.com/1999-4893/2/1/76.

141. Zhang, Y., Kolinski, A., and Skolnick, J. (2003). TOUCHSTONE II: a new approach to ab initio protein structure prediction. Biophys. J. *85*, 1145–64. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1303233{\&}tool=pmcentrez{\&}rendertype=

142. Wang, S., Li, W., Zhang, R., Liu, S., and Xu, J. (2016). CoinFold: a web server for protein contact prediction and contact-assisted protein folding. Nucleic Acids Res., gkw307. Available at: http://nar.oxfordjournals.org/content/early/2016/04/25/nar.gkw307.long.

143. Adhikari, B., Bhattacharya, D., Cao, R., and Cheng, J. (2015). CONFOLD: Residue-residue contact-guided ab initio protein folding. Proteins *83*, 1436–49. Available at: http://www.ncbi.nlm.nih.gov/pubmed/25974172.

144. Oliveira, S.H.P. de, Shi, J., and Deane, C.M. (2016). Comparing co-evolution methods and their application to template-free protein structure prediction. Bioinformatics, btw618. Available at: http://www.ncbi.nlm.nih.gov/pubmed/27682254.

145. Rodriguez-Rivas, J., Marsili, S., Juan, D., and Valencia, A. (2016). Conservation of coevolving protein interfaces bridges prokaryote-eukaryote homologies in the twilight zone. Proc. Natl. Acad. Sci. U. S. A. *113*, 15018–15023. Available at: http://www.ncbi.nlm.nih.gov/pubmed/27965389 http://www.pubmedcentral.nih.gov/articlerender.fcgi?arti

146. Feinauer, C., Szurmant, H., Weigt, M., and Pagnani, A. (2016). Inter-Protein Sequence Co-Evolution Predicts Known Physical Interactions in Bacterial Ribosomes and the Trp Operon. PLoS One *11*, e0149166. Available at: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0149166.

147. Gueudr, T., Baldassi, C., Zamparo, M., Weigt, M., and Pagnani, A. (2016). Simultaneous identification of specifically interacting paralogs and inter-protein contacts by Direct-Coupling Analysis. 19. Available at: http://arxiv.org/abs/1605.03745.

148. Bitbol, A.-F., Dwyer, R.S., Colwell, L.J., and Wingreen, N.S. (2016). Inferring interaction partners from protein sequences. Proc. Natl. Acad. Sci. *113*, 12180–12185. Available at: http://www.ncbi.nlm.nih.gov/pubmed/27663738 http://www.pubmedcentral.nih.gov/articlerender.fcgi?arti http://www.pnas.org/lookup/doi/10.1073/pnas.1606762113.

149. Uguzzoni, G., John Lovis, S., Oteri, F., Schug, A., Szurmant, H., and Weigt, M. (2017). Large-scale identification of coevolution signals across homo-oligomeric protein interfaces by direct coupling analysis. Proc. Natl. Acad. Sci. *114*, E2662–E2671. Available at: http://www.ncbi.nlm.nih.gov/pubmed/28289198 http://www.pubmedcentral.nih.gov/articlerender.fcgi?arti http://www.pnas.org/lookup/doi/10.1073/pnas.1615068114.

150. 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.

151. 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.

152. 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.

153. 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.

154. Morcos, F., Jana, B., Hwa, T., and Onuchic, J.N. (2013). Coevolutionary signals across protein lineages help capture multiple protein conformations. Proc. Natl. Acad. Sci. U. S. A. *110*, 20533–8. Available at: http://www.pnas.org/content/110/51/20533.short.

155. Jeon, J., Nam, H.-J., Choi, Y.S., Yang, J.-S., Hwang, J., and Kim, S. (2011). Molecular evolution of protein conformational changes revealed by a network of evolutionarily coupled residues. Mol. Biol. Evol. *28*, 2675–85. Available at: http://mbe.oxfordjournals.org/content/28/9/2675.full.

156. Woniak, P., Kotulska, M., and Vriend, G. (2017). Correlated mutations distinguish misfolded and properly folded proteins. Bioinformatics *33*, 1497–1504.

157. Cao, R., Adhikari, B., Bhattacharya, D., Sun, M., Hou, J., and Cheng, J. (2016). QAcon: single model quality assessment using protein structural and contact information with machine learning techniques. Bioinformatics *14*, btw694. Available at: https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btw694.

158. Terashi, G., Nakamura, Y., Shimoyama, H., and Takeda-Shitaka, M. (2014). Quality Assessment Methods for 3D Protein Structure Models Based on a Residue–Residue Distance Matrix Prediction. Chem. Pharm. Bull. *62*, 744–753. Available at: https://www.jstage.jst.go.jp/article/cpb/62/8/62{_}c13-00973/{_}article.

159. Nawy, T. (2016). Structural biology: RNA structure from sequence. Nat. Methods *13*, 465–465. Available at: http://dx.doi.org/10.1038/nmeth.3892.

160. Weinreb, C., Gross, T., Sander, C., and Marks, D.S. (2015). 3D RNA from evolutionary couplings. Available at: http://arxiv.org/abs/1510.01420.

161. De Leonardis, E., Lutz, B., Ratz, S., Cocco, S., Monasson, R., Schug, A., and Weigt, M. (2015). Direct-Coupling Analysis of nucleotide coevolution facilitates RNA secondary and tertiary structure prediction. Nucleic Acids Res., gkv932. Available at: http://nar.oxfordjournals.org/content/early/2015/09/29/nar.gkv932.full.

162. Wu, N.C., Du, Y., Le, S., Young, A.P., Zhang, T.-H., Wang, Y., Zhou, J., Yoshizawa, J.M., Dong, L., and Li, X. *et al.* (2016). Coupling high-throughput genetics with phylogenetic information reveals an epistatic interaction on the influenza A virus M segment. BMC Genomics *17*, 46. Available at: http://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-015-2358-7.

163. Figliuzzi, M., Jacquier, H., Schug, A., Tenaillon, O., and Weigt, M. (2015). Coevolutionary landscape inference and the context-dependence of mutations in beta-lactamase TEM-

1. Mol. Biol. Evol., msv211. Available at: http://mbe.oxfordjournals.org/content/early/2015/10/06/molbev.msv211.abstract.

164. Asti, L., Uguzzoni, G., Marcatili, P., and Pagnani, A. (2016). Maximum-Entropy Models of Sequenced Immune Repertoires Predict Antigen-Antibody Affinity. PLoS Comput. Biol. *12*, e1004870. Available at: http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004870.

165. Elhanati, Y., Murugan, A., Callan, C.G., Mora, T., and Walczak, A.M. (2014). Quantifying selection in immune receptor repertoires. Proc. Natl. Acad. Sci. U. S. A. *111*, 9875–80. Available at: http://www.ncbi.nlm.nih.gov/pubmed/24941953 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4103359.

166. Franceus, J., Verhaeghe, T., and Desmet, T. (2016). Correlated positions in protein evolution and engineering. J. Ind. Microbiol. Biotechnol., 1–9. Available at: http://link.springer.com/10.1007/s10295-016-1811-1.

167. Skwark, M.J., Croucher, N.J., Puranen, S., Chewapreecha, C., Pesonen, M., Xu, Y.Y., Turner, P., Harris, S.R., Beres, S.B., and Musser, J.M. *et al.* (2017). Interacting networks of resistance, virulence and core machinery genes identified by genome-wide epistasis analysis. PLOS Genet. *13*, e1006508. Available at: http://dx.plos.org/10.1371/journal.pgen.1006508.

168. Fox, G., Sievers, F., and Higgins, D.G. (2016). Using de novo protein structure predictions to measure the quality of very large multiple sequence alignments. Bioinformatics *32*, 814–20. Available at: http://www.ncbi.nlm.nih.gov/pubmed/26568625.

169. 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?

170. 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/articlerender.fcgi?artid=3823628{\&}tool=pmce

171. 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.

172. 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.

173. 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.

174. 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.

175. 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.

176. 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.

177. 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.

178. 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.

179. 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.

180. 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.

181. 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.

182. 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.

183. 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.

184. 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.

185. 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.

186. 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.

187. 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.

188. 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.

189. Andrieu, C., Freitas, N. de, Doucet, A., and Jordan, M.I. (2003). An Introduction to MCMC for Machine Learning. Mach. Learn. *50*, 5–43. Available at: http://link.

springer.com/10.1023/A:1020281327116.

190. Fischer, A., and Igel, C. (2012). An Introduction to Restricted Boltzmann Machines. Lect. Notes Comput. Sci. Prog. Pattern Recognition, Image Anal. Comput. Vision, Appl. *7441*, 14–36. Available at: http://link.springer.com/chapter/10.1007/978-3-642-33275-3{_}2.

191. 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/inde

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

193. 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.

194. Bottou, L. (2010). Large-Scale Machine Learning with Stochastic Gradient Descent. 177–186. Available at: https://link.springer.com/chapter/10.1007/978-3-7908-2604-3{_}16.

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

196. Zeiler, M.D. (2012). ADADELTA: An Adaptive Learning Rate Method. 6. Available at: http://arxiv.org/abs/1212.5701.

197. Bengio, Y. (2012). Practical Recommendations for Gradient-Based Training of Deep Architectures. In Neural networks: Tricks of the trade (Springer Berlin Heidelberg), pp. 437–478. Available at: https://arxiv.org/pdf/1206.5533v2.pdf.

198. Mahsereci, M., Balles, L., Lassner, C., and Hennig, P. (2017). Early Stopping without a Validation Set. arXiv. Available at: http://arxiv.org/abs/1703.09580.

199. 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/cdmiguel.pdf.

200. Tieleman, T. (2008). Training Restricted Boltzmann Machines using Approximations to the Likelihood Gradient. Proc. 25th Int. Conf. Mach. Learn. *307*, 7.

201. Fischer, A., and Igel, C. (2010). Empirical Analysis of the Divergence of Gibbs Sampling Based Learning Algorithms for Restricted Boltzmann Machines. In Artif. neural networks – icann 2010 (Springer, Berlin, Heidelberg), pp. 208–217. Available at: http://link.springer.com/10.1007/978-3-642-15825-4{_}26.

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

203. 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/.

204. Asuncion, A.U., Liu, Q., Ihler, A.T., and Smyth, P. (2010). Learning with Blocks: Composite Likelihood and Contrastive Divergence. Proc. Mach. Learn. Res. *9*, 33–40. Available at: http://www.ics.uci.edu/{~}asuncion/pubs/AISTATS{_}10.pdf.

205. Kingma, D., and Ba, J. (2014). Adam: A Method for Stochastic Optimization. Available at: http://arxiv.org/abs/1412.6980.

206. 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.

207. 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/.

208. 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/.

209. Breiman, L. (2001). Random Forests. Mach. Learn. *45*, 5–32. Available at: http://link.springer.com/10.1023/A:1010933404324.

210. 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.

211. Louppe, G. (2014). Understanding Random Forests: From Theory to Practice. Available at: http://arxiv.org/abs/1407.7502.

212. 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.

213. 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.

214. 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.

215. 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.

216. 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.

217. 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/S0022283699930

218. 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.

219. Chollet, F. and others (2015). Keras. Available at: https://github.com/fchollet/keras.

220. Dieleman, S., Schlter, J., Raffel, C., Olson, E., Snderby, S.K., Nouri, D., Maturana, D., Thoma, M., Battenberg, E., and Kelly, J. *et al.* (2015). Lasagne: First release. Available at:

https://zenodo.org/record/27878.

221. Byrd, R.H., Lu, P., Nocedal, J., and Zhu, C. (1995). A Limited Memory Algorithm for Bound Constrained Optimization. SIAM J. Sci. Comput. *16*, 1190–1208. Available at: http://epubs.siam.org/doi/10.1137/0916069.

222. 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/articlerender.fcgi?artid=PMC52614.

223. 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.

224. Grantham, R. (1974). Amino acid difference formula to help explain protein evolution. Science *185*, 862–4. Available at: http://www.ncbi.nlm.nih.gov/pubmed/4843792.

225. 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/articlerender.fcgi?artid=PMC2238890.

226. Zimmerman, J.M., Eliezer, N., and Simha, R. (1968). The characterization of amino acid sequences in proteins by statistical methods. J. Theor. Biol. *21*, 170–201. Available at: http://www.ncbi.nlm.nih.gov/pubmed/5700434.

227. 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.

228. Kyte, J., and Doolittle, R.F. (1982). A simple method for displaying the hydropathic character of a protein. J. Mol. Biol. *157*, 105–132. Available at: http://www.sciencedirect.com/science/article/pii/0022283682905150.

229. 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.

230. Pontius, J., Richelle, J., and Wodak, S.J. (1996). Deviations from Standard Atomic Volumes as a Quality Measure for Protein Crystal Structures. J. Mol. Biol. *264*, 121–136. Available at: http://www.ncbi.nlm.nih.gov/pubmed/8950272 http://linkinghub.elsevier.com/retrieve/pii/S0022283696906282.

231. Zhu, H., and Braun, W. (1999). Sequence specificity, statistical potentials, and three-dimensional structure prediction with self-corrected 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{\&}rendertype=

232. 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.

233. Livingstone, C.D., and Barton, G.J. (1993). Protein sequence alignments: a strategy for the hierarchical analysis of residue conservation. Bioinformatics *9*, 745–756. Available at: http://bioinformatics.oxfordjournals.org/content/9/6/745.