Bayesian Statistical Approach for Protein Residue-Residue Contact Prediction

Susann Vorberg

2017

Dissertation zur Erlangung des Doktorgrades der Fakultät für Chemie und Pharmazie der Ludwig-Maximilians-Universität München

Bayesian Statistical Approach for Protein Residue-Residue Contact Prediction

Susann Vorberg
aus
Leipzig, Deutschland

2017

Erklärung

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

Eidesstattliche Versicherung

Diese Dissertation wurde eigenständig und ohne unerlaubte Hilfe erarbeitet.

München, 03.11.2017 ....
Ort, Datum Susann Vorberg

Dissertation eingereicht am:	03.11.2017
Erstgutachter:	Dr. Johannes Söding
Zweitgutachter:	Prof. Dr. Julien Gagneur
Tag der mündlichen Prüfung:	11.12.2017

Acknowledgements

I am very grateful to Johannes Söding, for giving me the opportunity to work in his lab, for his supervision and guidance on this fascinating project. I learned a lot from you, not only analytical and statistical skills, but also about being a scientist and what holds the scientific world together at its core. Your enthusiasm and your convincing attitude always kept me going.

I also want to thank Julien Gagneur for supervising this thesis together with the other members of my examination board: Franz Herzog, Klaus Förstermann, Karl-Peter Hopfner and Oliver Keppler.
My thanks goes also to Roland Beckmann who was part of my thesis advisory committee. Special thanks goes to Julien who had great and pragmatic ideas that helped to keep the big picture in focus.

My gratitude goes also to the Quantitative Biosciences Munich graduate school. Foremost to Ulrike Gaul and Erwin Frey for installing this great melting pot of science. With QBM's financial support I could visit inspiring conferences that helped me to grow as scientist and maybe even more as a person. Additionally, I thank the staff Mara Kieke, Julia Schlehe, Filiz Civril, Markus Hohle and Michael Mende who organized so many great lectures, workshops, and events for us and were always ready to help.

I want to thank my group, the Söding lab, for their support and distraction in- and outside the lab. You were more than mere colleagues, you became true friends and made me enjoy coming to work every single day. In particular to Anja and Mark who stayed, like me, in Munich until the very end. It will be an honor to shut down the lights in our beloved jungle office together with you. Thanks a lot, Jessica, for finding the time for proofreading, even when there is no time at all. My thanks also go to the Gagneur group with whom we shared the office space at the LMU gene center for many years. Thanks for your open office doors during my Garching visits whenever I felt I needed company.

I also want to thank my former coaches, Henrik Lindner and Torsten Kunke, who supported my decision to leave the army and send me on my way to becoming Dr. Susi. Without you I might still be soaring the skies.

I want to thank my family for raising me curious and skeptical and therefore having me equipped with fundamental scientific skills.

Daniel you are the love of my life.

Summary

Despite continuous efforts in automating experimental structure determination and systematic target selection in structural genomics projects, the gap between the number of known amino acid sequences and solved 3D structures for proteins is constantly widening. While DNA sequencing technologies are advancing at an extraordinary pace, thereby constantly increasing throughput while at the same time reducing costs, protein structure determination is still labour intensive, time-consuming and expensive. This trend illustrates the essential importance of complementary computational approaches in order to bridge the so called sequence-structure gap.

About half of the protein families lack structural annotation and therefore are not amenable to techniques that infer protein structure from homologs. These protein families can be addressed by *de novo* structure prediction approaches that in practice are often limited by the immense computational costs required to search the conformational space for the lowest-energy conformation. Improved predictions of contacts between amino acid residues have been demonstrated to sufficiently constrain the overall protein fold and thereby extend the applicability of *de novo* methods to larger proteins. Residue-residue contact prediction is based on the idea that selection pressure on protein structure and function can lead to compensatory mutations between spatially close residues. This leaves an echo of correlation signatures that can be traced down from the evolutionary record. Despite the success of contact prediction methods, there are several challenges. The most evident limitation lies in the requirement of deep alignments, which excludes the majority of protein families without associated structural information that are the focus for contact guided *de novo* structure prediction. The heuristics applied by current contact prediction methods pose another challenge, since they omit available coevolutionary information.

This work presents two different approaches for addressing the limitations of contact prediction methods. Instead of inferring evolutionary couplings by maximizing the pseudo-likelihood, I maximize the full likelihood of the statistical model for protein sequence families. This approach performed with comparable precision up to minor improvements over the pseudo-likelihood methods for protein families with few homologous sequences. A Bayesian statistical approach has been developed that provides posterior probability estimates for residue-residue contacts and eradicates the use of heuristics. The full information of coevolutionary signatures is exploited by explicitly modelling the distribution of statistical couplings that reflects the nature of residue-residue interactions. Surprisingly, the posterior probabilities do not directly translate into more precise predictions than obtained by pseudo-likelihood methods combined with prior knowledge. However, the Bayesian framework offers a statistically clean and theoretically solid treatment for the contact prediction problem. This flexible and transparent framework provides a convenient starting point for further developments, such as integrating more complex prior knowledge. The model can also easily be extended towards the derivation of probability estimates for residue-residue distances to enhance precision of predicted structures.

Table of Contents

1 Background

1.1 Biological Background

In 1972, Anfinsen and his colleges 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] or 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 reliably predict a protein's structure given only its amino acid sequence [4,5]. *De novo* protein structure prediction methods minimize physical or knowledge based energy functions to identify the lowest-energy conformation that generally corresponds to the native protein conformation. However, due to the high degree of conformational flexibility, the search space of possible conformations cannot be explored exhaustively for a typical protein. 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 $3^{200} \approx 10^{95}$ 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 successful 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 homologous protein [8], that is a protein of shared ancestry. 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 Protein Data Bank (PDB) is the main repository for macromolecular structures and currently (October 2017) holds about 135,000 atomic models of proteins [12]. The primary technique for determining protein structures is X-ray crystallography, accounting for roughly 90% of entries in the PDB. About 9%

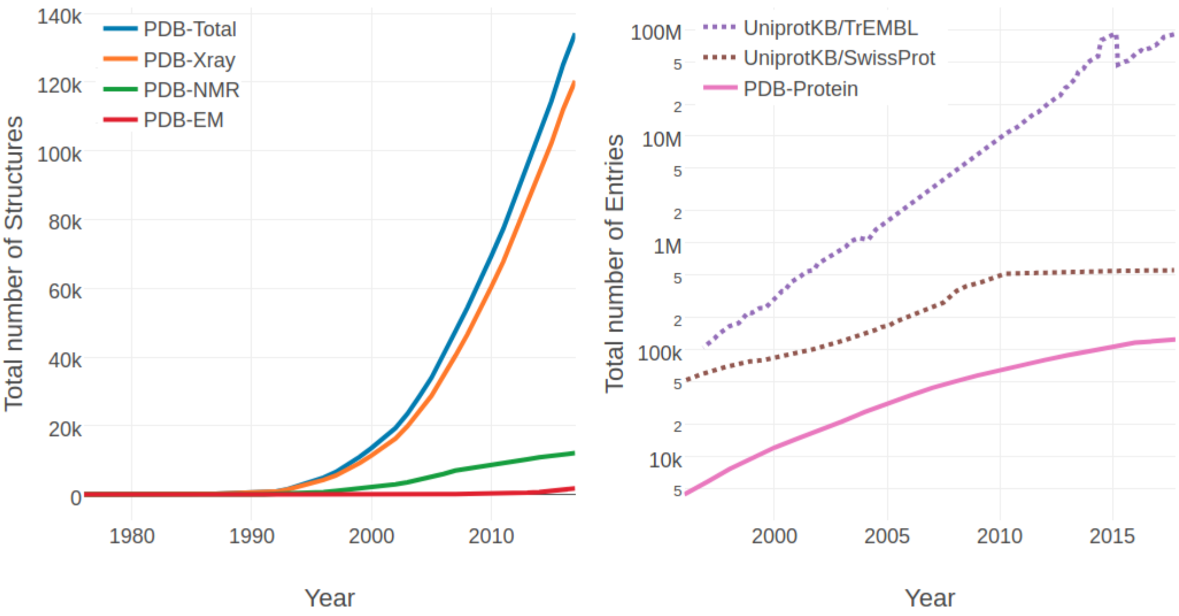


Figure 1.1: Comparing the amount of primary and tertiary protein structures over time. **Left** Yearly growth of protein structures in the PDB [12] by structure determination method. **Right** Yearly growth of database entries in the UniprotKB/TrEMBL [13], containing automatically annotated protein sequences, in the UniprotKB/SwissProt [13], containing manually curated protein sequences and in the PDB containing solved protein structures.

of protein structures have been solved using nuclear magnetic resonance (NMR) spectroscopy and less than 1% using electron microscopy (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 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 altered side chain 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 [21]. Not only has the amount of time decreased that is needed to sequence whole genomes but also costs have been drastically reduced [22]. 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[23]. At the beginning of 2017, Illumina announced the launch of their latest high-throughput sequencing technology, NovaSeq, which is capable of sequencing ∼48 human genomes in parallel at 30x coverage within ∼45 hours [24]. Advances in sequencing 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 the lab [25–27]. With these approaches the genomic coverage of the microbial world is expanding which is directly reflected in a substantial increase in novel protein families [28–30]. More than 70 000 genomes have been completely sequenced and 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 [13,31].

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 [5]. This trend illustrates the essential importance of computational approaches that can complement experimental structural biology efforts in order to bridge 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 routinely and conveniently used in life-science research and by the biological community [5,32]. *De novo* methods aiming at predicting protein structures from sequence alone are required in case no homologous 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 different types of experiments can help to reduce the degrees of freedom in the conformational search space [5]. Several sophisticated integrative approaches have been developed and proven to be powerful [33–35]. 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 [36–38].

Another complementary source of information is given by predicted protein residue-residue contacts. The invention of direct coupling analysis (DCA) in 2009 was a breakthrough in the development of computational methods to infer spatially close residue pairs from coevolutionary signals in the evolutionary record of protein families [39]. Since then, the field of contact prediction has experienced rapid progress and methods are continuously improving. Modern contact prediction approaches produce predictions that are sufficiently accurate to successfully assist the *de novo* prediction of protein structures [40]. The last years have seen an enormous 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 [41], studying alternative conformations [42] or inferring evolutionary fitness landscapes and quantifying mutational effects [43].

It has long been known that native contacts can be used to reliably reconstruct native protein 3D structure [44]. This is because a contact map retains the full 3D structural information of a protein, even though it provides only a 2D representation of the protein structure. For a protein of length L, a contact map is a binary $L \times L$ matrix, where the binary element in the matrix $C(i,j)$ for two residues i and j is given by

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

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 threshold (typically 8 $\mathring{A}$). Figure 1.2 shows an example of a residue-

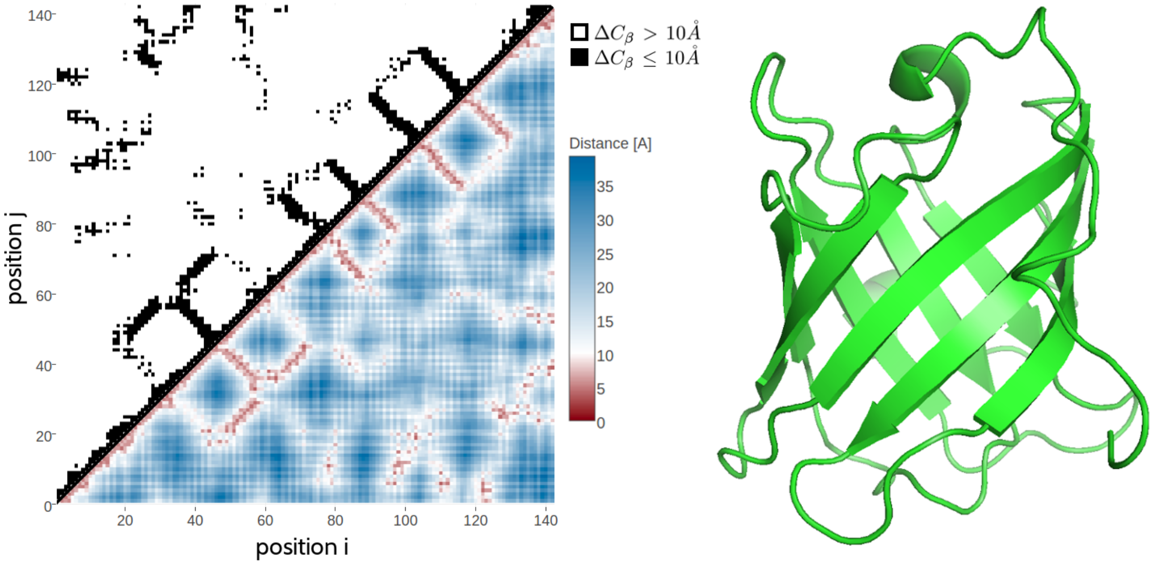


Figure 1.2: 2D and 3D representations of protein triabin, a thombin inhibitor from triatoma pallidipennis (PDB identifier 1avg chain I). **Left** The upper left matrix illustrates a contact map using an $10\mathring{A}$ C_β cutoff. A black square is drawn at position (i, j) if the C_β atoms of residues i and j are closer than $10\mathring{A}$ 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.

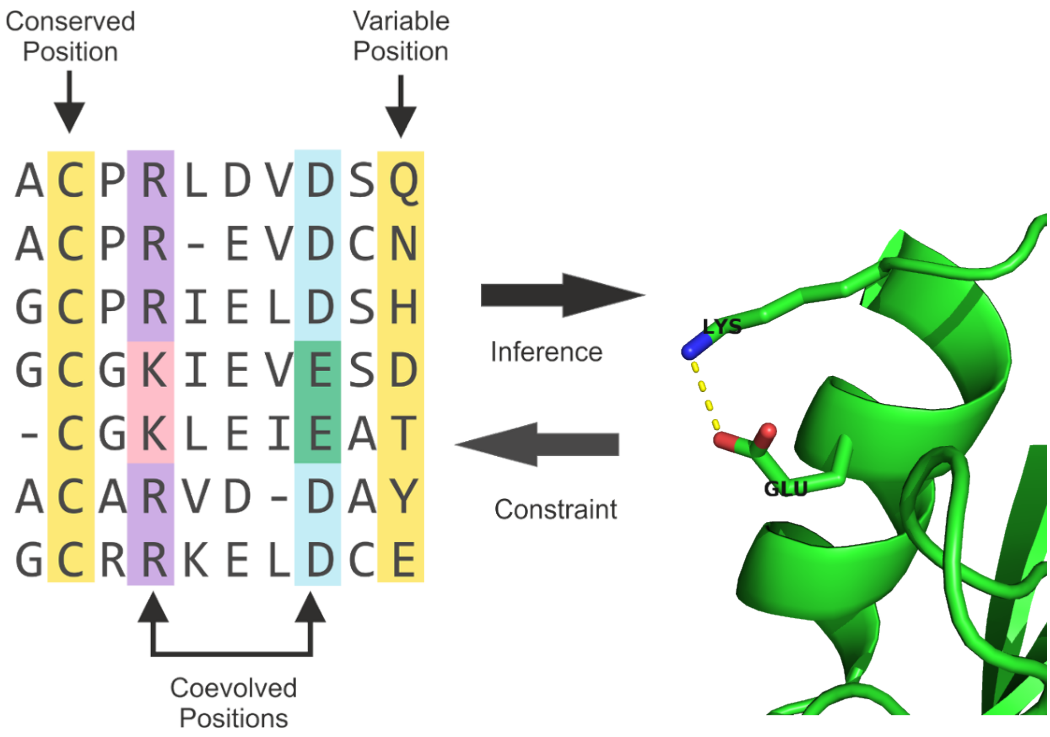
residue contact map generated from a small protein domain. While it has been shown that only a small subset of native contacts is sufficient to allow accurate modelling of the protein structure, the quality of predicted residue-residue contacts crucially controls the quality of the final structural model [45,46]. Currently published DCA methods are very successful at predicting contacts for large protein families. However they all apply the same heuristics on top of the underlying statistical model thereby ignoring valuable information. It is a reasonable assumption that by making full use of the available information, the predictive performance of the models should improve and as a consequence extend the applicability of DCA methods to smaller protein families. The aim of this thesis is therefore to improve the models for residue-residue contact prediction by developing a flexible and transparent Bayesian framework that dresses these issues.

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.

1.2 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 [47]. 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.



A C P R L D V D S Q
A C P R - E V D C N
G C P R I E L D S H
G C G K I E V E S D
- C G K L E I E A T
A C A R V D - D A Y
G C R R K E L D C E

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

Highly constrained residues and interactions will be strongly conserved [48]. Another possibility to maintain structural integrity is the mutual compensation of unbeneficial mutations. For example, the unfavorable 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[49]. The MSA of a protein family comprises homologous sequences that have descended from a common ancestor and are aligned relative to each other. According to the hypothesis, compensatory mutations show up as correlations between the amino acid types of pairs of MSA columns and can be used to infer spatial proximity of residue pairs (see Figure 1.3).

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

1.2.1 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 [47,49–52].

Many methods have been developed that are rooted in information theory and use MI measures to describe the dependencies between sites in the alignment [53–55]. Phylogenetic and

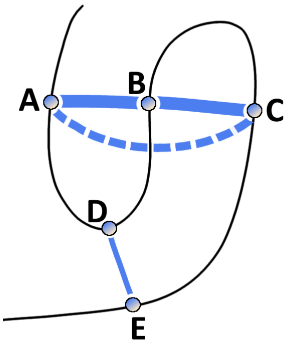


Figure 1.4: Effects of chained covariation obscure signals from true physical interactions. Consider residues A through E with physical interactions between the residue pairs A-B, B-C and D-E. The thickness of blue lines between residues reflects the strength of statistical dependencies between the corresponding 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.

entropic biases have been identified as strong sources of noise that confound the true co-evolution signal [55–57]. Different variants of MI based approaches address these effects and improve on the signal-to-noise ratio [56,58,59]. The most prominent correction for background noises is the so called average product correction (APC) that is still used by many modern methods and is discussed in section 1.3.6 [60]. 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 [61,62].

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 [39,63,64]. The concept of transitive effects is illustrated in Figure 1.4. 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, even though 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 [64].

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 [39,64]. Even though local statistical methods cannot compete with modern predictors, *OMES* and MI based scores often serve as a baseline in performance benchmarks for contact prediction [65,66].

1.2.2 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 [39,63].

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 [63]. They consider a

Potts model that can be derived under a maximum entropy assumption and use the model specific coupling parameters to infer interactions. At that time the wider implications of this advancement went unnoted, but meanwhile the Pott's Model has become the most prominent statistical model for contact prediction. Section 1.3 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 [64,67]. Their fast Bayesian network model incorporates additional prior information and phylogenetic correction via APC but cannot compete with the pseudo-likelihood approaches presented in section 1.3.5.

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* [68], *SVM-SEQ* [69]), random forests (*ProC_S3* [70], *TMhhcp* [71], *PhyCMap* [72]), neural networks (*NETCSS* [73], *SAM* [74], [75], *SPINE-2D* [76], *NNCon* [77]) deep neural networks (*DNCon* [78], *CMAPpro* [79]) and ensembles of genetic algorithm classifiers (*GaC* [80]).

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 [68,81]. 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* [82]. In a follow-up version *PSICOV* has been replaced with *gaussianDCA* and the sequence-based method *PhyCMap* [83]. *EPC-MAP* was published in 2014 integrating *GREMLIN* as a coevolution feature with physico-chemical information from predicted ab initio protein structures [84]. In 2015, *MetaPSICOV* was released combining predictions from *PSICOV*, *mfDCA* and *CCMpred* with other sequence derived features [85]. *RaptorX* uses *CCMpred* as coevolution feature and other standard contact prediction features within an ultra-deep neural network [86]. 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 [87,88].

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 [79]. 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 recognized [89]. 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 [79,85,86,90].

Even though 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 [91]. As coevolution scores comprise the most informative features among the set of input features, it is clear that meta-predictors will benefit from further improvements of pure coevolution methods [86,87].

1.3 Modelling Protein Families with Potts Model

Global statistical models enable the distinction of direct statistical dependencies between residues from indirect dependencies mediated through other residues. This is achieved by inferring contacts from a joint probability distribution over all residues instead of treating residues independently. 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 [92,93], 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

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

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

$$\begin{aligned} f(x_i\!=\!a) &\equiv p(x_i\!=\!a) \\ &= \sum_{y_1,\ldots,y_L=1}^{20} p(\mathbf{y}) I(y_i\!=\!a) \\ f(x_i\!=\!a, x_j\!=\!b) &\equiv p(x_i\!=\!a, x_j\!=\!b) \\ &= \sum_{y_1,\ldots,y_L=1}^{20} p(\mathbf{y}) I(y_i\!=\!a, y_j\!=\!b) \; . \end{aligned} \tag{1.3}$$

Solving for the distribution $p(\mathbf{x})$ that maximizes the Shannon entropy $S = -\sum_{\mathbf{x}} p(\mathbf{x}) \log p(\mathbf{x})$ while satisfying the constraints given by the empirical amino acid frequencies in eq. (1.3) by introducing Lagrange multipliers $\mathbf{w}_{ij}$ and v_i, results in the formulation of the *Potts model*,

$$p(\mathbf{x}|\mathbf{v},\mathbf{w}) = \frac{1}{Z(\mathbf{v},\mathbf{w})} \exp\left(\sum_{i=1}^{L} v_i(x_i) \sum_{1 \leq i < j \leq L} w_{ij}(x_i, x_j) \right) . \tag{1.4}$$

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 probability adds up to one by summing over all possible assignments to $\mathbf{x}$,

$$Z(\mathbf{v},\mathbf{w}) = \sum_{y_1,\ldots,y_L=1}^{20} \exp\left(\sum_{i=1}^{L} v_i(y_i) \sum_{1 \leq i < j \leq L} w_{ij}(y_i, y_j) \right) . \tag{1.5}$$

1.3.1 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 negligible 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 [94,95].

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

1.3.2 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 parameterization yield identical probability distributions.

For example, adding a constant to all elements in v_i for any fixed position i or similarly adding a constant to v_{ia} for any fixed position i and amino acid a and subtracting the same constant from the qL coefficients w_{ijab} with $b \in \{1, \ldots, q\}$ and $j \in \{1, \ldots, L\}$ leaves the probabilities for all sequences under the model unchanged, since such a change will be compensated by a change of $Z(\mathbf{v},\mathbf{w})$ in eq. (1.5).

The over-parameterization is referred to as *gauge invariance* in statistical physics literature and can be eliminated by removing parameters [39,96]. 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. [39] imposed by the restraints,

$$\sum_{a=1}^{q} v_{ia} = \sum_{a=1}^{q} w_{ijab} = \sum_{a=1}^{q} w_{ijba} = 0 \tag{1.6}$$

for all i, j, b or the *lattice-gas gauge* used by Morcos et al [96] and Marks et al [40] imposed by restraints

$$\mathbf{w}_{ij}(q,a) = \mathbf{w}_{ij}(a,q) = v_i(q) = 0 \tag{1.7}$$

for all i, j, a [97].

Alternatively, the indeterminacy can be fixed by including a regularization prior (see next section). The regularizer selects for a unique solution among all parameterization of the optimal distribution and therefore eliminates the need to choose a gauge [98–100].

1.3.3 Inferring Parameters of 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 $\mathbf{X}$:

$$\begin{aligned}
\mathrm{LL}(\mathbf{v},\mathbf{w}|\mathbf{X}) &= \sum_{n=1}^{N} \log p(\mathbf{x}_n|\mathbf{v},\mathbf{w}) \\
&= \sum_{n=1}^{N} \left[\sum_{i=1}^{L} v_i(x_{ni}) + \sum_{1\leq i<j\leq L} w_{ij}(x_{xn}, x_{nj}) - \log Z(\mathbf{v},\mathbf{w}) \right]
\end{aligned} \tag{1.8}$$

The number of parameters in a Potts model is typically larger than the number of observations, i.e. the number of sequences in the MSA. Considering a protein of length $L = 100$, there are approximately 2×10^6 parameters in the model whereas the largest protein families comprise only around 10^5 sequences (see Figure 1.12). An under determined problem like this renders the use of regularizers necessary in order to prevent overfitting.

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

$$\begin{aligned}
R(\mathbf{v},\mathbf{w}) &= \log\left[\mathcal{N}(\mathbf{v}|\mathbf{0},\lambda_v^{-1}I)\mathcal{N}(\mathbf{w}|\mathbf{0},\lambda_w^{-1}I)\right] \\
&= -\frac{\lambda_v}{2}||\mathbf{v}||_2^2 - \frac{\lambda_w}{2}||\mathbf{w}||_2^2 + \text{const.}\ ,
\end{aligned} \tag{1.9}$$

where the strength of regularization is tuned via the regularization coefficients λ_v and λ_w [101–103].

However, optimizing the log-likelihood requires computing the partition function Z given in eq. (1.5) that sums q^L terms. Computing this sum is intractable for realistic protein domains with more than 100 residues. Consequently, evaluating the likelihood function at each iteration of an optimization procedure is infeasible due to the exponential complexity of the partition function in protein length L.

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

1.3.4 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 [63]. 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 [104]. As the calculations involved were very time-consuming and at that time required super computing 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 [39]. Even though 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 [105]. 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 [103].

Also in 2011, Morcos et al. introduced a naive mean-field inversion approximation to the partition function, named *mfDCA* [96]. 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 intra-domain contacts for 131 protein families and facilitated the prediction of protein structures from accurately predicted contacts in [40].

The initial work by Balakrishnan and colleagues went almost unnoted as it was not primarily targeted to the problem of contact prediction. Ekeberg and colleagues independently developed the pseudo-likelihood method *plmDCA* in 2013 and showed its superior precision over *mfDCA* [99].

A related approach to mean-field approximation is sparse inverse covariance estimation, named *PSICOV*, developed by Jones et al. (2012) [66]. PSICOV uses an L1-regularization, known as graphical Lasso, to invert the correlation matrix and learn a sparse graphical model [106]. 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 [107].

Another related approach to *mfDCA* and *PSICOV* is *gaussianDCA*, proposed in 2014 by Baldassi et al. [108]. Similar to the other both approaches, they model the data as multivariate Gaussian but within a simple Bayesian formalism by using a suitable prior and estimating parameters over the posterior distribution.

So far, pseudo-likelihood has proven to be the most successful approximation of the likelihood with respect to contact prediction performance. Currently, there exist several implementations of pseudo-likelihood maximization that vary in slight details, perform similarly and thus are equally popular in the community, such as CCMpred [101], plmDCA[102] and GREMLIN [103].

1.3.5 Maximum Likelihood Inference for Pseudo-Likelihood

The pseudo-likelihood is a rather old estimation principle that was suggested by Besag already in 1975 [109]. 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:

$$\begin{aligned} p(\mathbf{x}|\mathbf{v},\mathbf{w}) &\approx \prod_{i=1}^{L} p(x_i|\mathbf{x}_{\backslash xi},\mathbf{v},\mathbf{w}) \\ &= \prod_{i=1}^{L} \frac{1}{Z_i(\mathbf{v},\mathbf{w})} \exp\left(v_i(x_i) \sum_{1 \leq i < j \leq L} w_{ij}(x_i, x_j) \right) \end{aligned} \tag{1.10}$$

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

$$Z_i(\mathbf{v},\mathbf{w}) = \sum_{a=1}^{q} \exp\left(v_i(a) \sum_{1 \leq i < j \leq L} w_{ij}(a, x_j) \right) \tag{1.11}$$

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,

$$\begin{aligned} \mathrm{pLL}(\mathbf{v},\mathbf{w}|\mathbf{X}) &= \sum_{n=1}^{N}\sum_{i=1}^{L} \log p(x_i|\mathbf{x}_{\backslash xi},\mathbf{v},\mathbf{w}) \\ &= \sum_{n=1}^{N}\sum_{i=1}^{L} \left[v_i(x_{ni}) + \sum_{j=i+1}^{L} w_{ij}(x_{ni}, x_{nj}) - \log Z_{ni}(\mathbf{v},\mathbf{w}) \right] , \end{aligned} \tag{1.12}$$

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

$$\hat{\mathbf{v}}, \hat{\mathbf{w}} = \underset{\mathbf{v},\mathbf{w}}{\operatorname{argmax}} \ \mathrm{pLL}(\mathbf{v},\mathbf{w}|\mathbf{X}) + R(\mathbf{v},\mathbf{w}) \ . \tag{1.13}$$

Even though 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 [95,98–100]. 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 [98,109,110]. That is, as the number of sequences in the alignment increases, pseudo-likelihood estimates converge towards the true full likelihood parameters.

1.3.6 Computing Contact Maps

Model inference as described in the last section yields MAP estimates of the couplings $\hat{\mathbf{w}}_{ij}$. In order to obtain a scalar measure for the coupling strength between two residues i and j, all available methods presented in section 1.3.4 heuristically map the 21×21 dimensional coupling matrix $\mathbf{w}_{ij}$ to a single scalar quantity.

mpDCA [39] and *mfDCA* [40,96] 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* [99,102], *CCMpred* [101], *GREMLIN* [103]) compute the *Frobenius norm* of the coupling matrix $\mathbf{w}_{ij}$ to obtain a scalar contact score C_{ij},

$$C_{ij} = ||\mathbf{w}_{ij}||_2 = \sqrt{\sum_{a,b=1}^{q} w_{ijab}^2} \ . \tag{1.14}$$

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 [99,108,111]. *PSICOV* [66] 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* [39]. Therefore, the coupling matrices should be transformed to *zero-sum gauge* before computing the Frobenius norm

$$\mathbf{w}'_{ij} = \mathbf{w}_{ij} - \mathbf{w}_{ij}(\cdot, b) - \mathbf{w}_{ij}(a, \cdot) + \mathbf{w}_{ij}(\cdot, \cdot) \ , \tag{1.15}$$

where $\cdot$ denotes average over the respective indices [99,101,102,108].

Another commonly applied heuristic, known as average product correction (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 [60]. 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},

$$C_{ij}^{APC} = C_{ij} - \frac{\overline{C_i} \; \overline{C_j}}{\overline{C_{ij}}} \ . \tag{1.16}$$

Visually, APC creates a *smoothing* effect on the contact maps that is illustrated in Figure 1.5 and it has been found to substantially boost contact prediction performance [60,103]. It was first adopted by *PSICOV* [66] 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 [112]. 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.

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

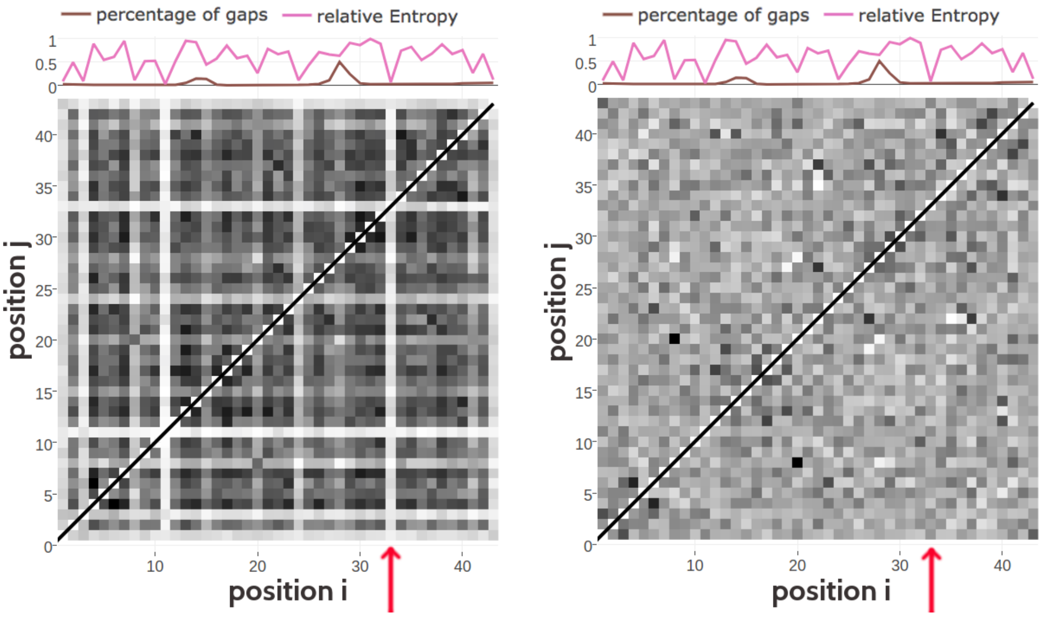


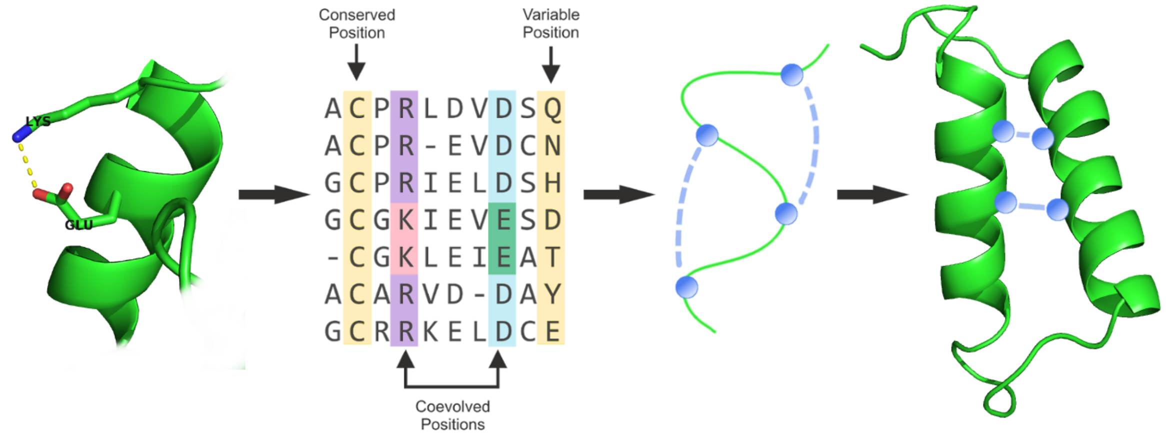
Figure 1.5: Contact maps computed from pseudo-likelihood couplings. Subplot on top of the contact maps illustrates the normalized Shannon entropy (pink line) and percentage of gaps for every position in the alignment (brown line). **Left**: Contact map computed with Frobenius norm as in eq. (1.14). Overall coupling values are dominated by entropic effects, i.e. the amount of variation for a MSA position, leading to striped brightness patterns. Positions with high column entropy have higher overall coupling values than positions with low column entropy, for example position 33 (marked with red arrow). **Right**: previous contact map but corrected for background noise with the APC, given in eq. (1.16).

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

1.4 Applications for Contact Prediction

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 [44]. 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 [36,113–117]. Predicted contacts, however, have long been regarded as being of little use for structure prediction because of their high false-positive rates [118,119]. 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 *mfDCA* are



```
ACPRLDVDSQ
ACPR-EVDCN
GCPRIELDSH
GCGKIEVESD
-CGKLEIEAT
ACARVD-DAY
GCRRKELDCE
```

Figure 1.6: Generalized structure prediction pipeline integrating predicted contacts in form of distance constraints that guide conformational sampling.

sufficiently accurate to successfully deduce the native fold of the protein [40]. 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 [120–128].

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 [129], RASREC [130], RBOAleph [131], GDFuzz3D [132], PconsFold [133], C2S_Pipeline [134], FRAGFOLD + PSICOV [135], FILM3 [136], EVFold [40]. Figure 1.6 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 [44,45,134,137–139]. 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 [137]. 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 [45]. 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 overall fold and packing of tertiary structure whereas short-range contacts define local secondary structure [140]. It is a consistent finding that even though long-range contacts are of higher relevance than short-range contacts for structure reconstruction, their information alone is not sufficient [135,137,141]. 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 [45,46]. Marks and colleagues observed that isolated false positives have a much stronger detrimental effect on structure prediction than false positives close to true contacts [40]. 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 [142]. Frequently, folding protocols employ a filtering step to eliminate unsatisfied or conflicting constraints possibly originating from false-positive contacts [143,144]. Generally it is assumed that higher precision of predicted long-range contacts results in improved structural models, albeit there is no strong correlation as model quality depends on many other factors such as the secondary structure composition of the protein,

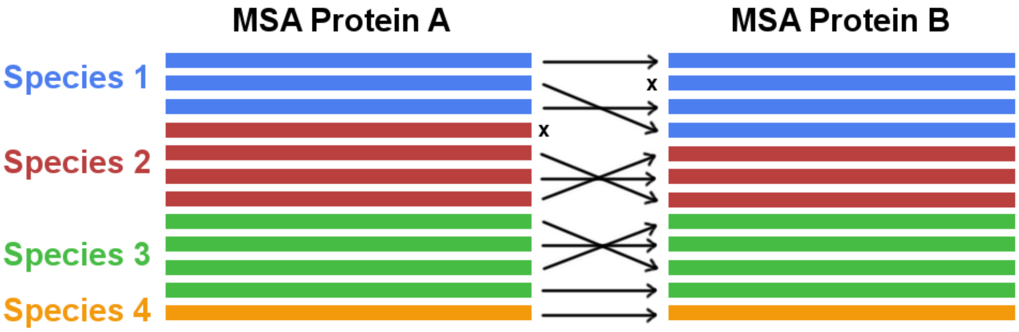


Figure 1.7: Concatenating two multiple sequence alignments. In case multiple paralogs exist for a gene in one species the correct interaction partner needs to be identified and matched (marked with arrows). Sequences that cannot be paired with a unique interaction partner need to be discarded (marked with x).

the domain size, the usage of additional sources of structural information, the type of distance constraint function and the particular structure reconstruction protocol [40,135,140,142,145].

Coevolution has not only been studied for residues pairs within a protein but also for residue pairs across protein–protein interfaces [121,122,127,146,147]. Even though 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 1.7. 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 [147]. New strategies have been developed based on the idea that an alignment with correctly matched paralogs will maximize the coevolution score [148,149].

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 unnecessary 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 inter-protein 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 [150,151] (see Figure 1.8).

Predicted contacts have also been applied in the analysis of potential alternative conformations of proteins [42,152–156]. Coevolutionary analysis detects all direct 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 (see Figure 1.8). Sfriso and colleagues developed an automated pipeline that introduces filtered predicted contacts as ensemble restraints into a molecular dynamics simulations and is able to detect alternative relevant conformational states [152].

Even though the coevolutionary methods have been developed for proteins, they have been

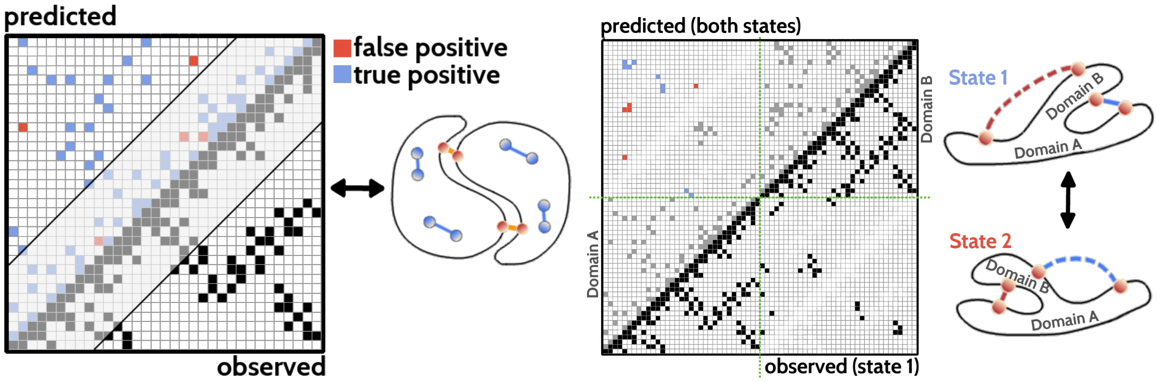


Figure 1.8: Predicted residue-residue contacts facilitate the study of diverse problems in the field of structural biology. **Left** Contacts can be used to define the interface of homo-oligomers. Given a monomeric protein structure, false positive long-range predictions represent potential contacts in the oligomeric protein interface. **Right** Predicted contacts can capture multiple states of a protein. Given the structure of a protein in a distinct conformation, false positive contacts might be satisfied in an alternative conformation of that protein.

successfully applied to analyse nucleotide coevolution and to predict RNA tertiary structures with the help of predicted nucleotide-nucleotide contacts [157–159]. 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 [159]. Despite the diminished accuracy, predicted nucleotide contacts have been demonstrated to improve RNA structure prediction over conventional methods [158].

Predicted residue-residue contacts have been used to tackle various other problems in the area of structural biology. Sadowski used predicted contacts to parse domain boundaries based on the simple idea that contacts are more abundant within domains than between domains [41]. Contact maps display patterns that reflect secondary structure elements, which can be parsed to detect alpha helices and beta-sheets [89,160]. 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 [119,161]. 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 [162,163].

The mathematical framework of the coevolution models used to predict residue-residue proteins has been found to be useful in other fields of biology beyond structure prediction. Skwark and colleagues applied the popular coevolution statistical models to genomes and developed a statistical method called *genomeDCA* [164,165]. 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) resistance.

The statistical 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 overall compliance of a sequence with the protein family under study and thereby provide quantitative predictions of mutational effects

[43,166,167]. Computational screening of mutational effects can support and complement the costly and time-consuming directed evolution or mutational screening experiments [43]. With a similar idea in mind, the coevolution models have been applied to sequences of human immune repertoires [168,169]. 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 [170]. The interpretation of the model parameters as energies has helped to analyse the sequence capacity of protein folds, that is how many sequences can fold into a specific structure [171].

Fox and colleagues turn the idea of DCA upside down. They developed a benchmark for testing the accuracy of large MSAs by evaluating the agreement between the predicted and the native contacts [172]. Based on the assumption that better alignments provide more accurate contact predictions, the alignment quality is inferred from the precision of predicted contacts.

1.5 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 1.4. 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 overall protein fold as already discussed. From these considerations emerged various standard benchmarks that have been established by the CASP community over many years [91,173,174]. 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 test set 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,

$$\text{precision} = \frac{TP}{TP + FP} \; , \tag{1.17}$$

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 [175]. Another informative metric is mean error defined as:

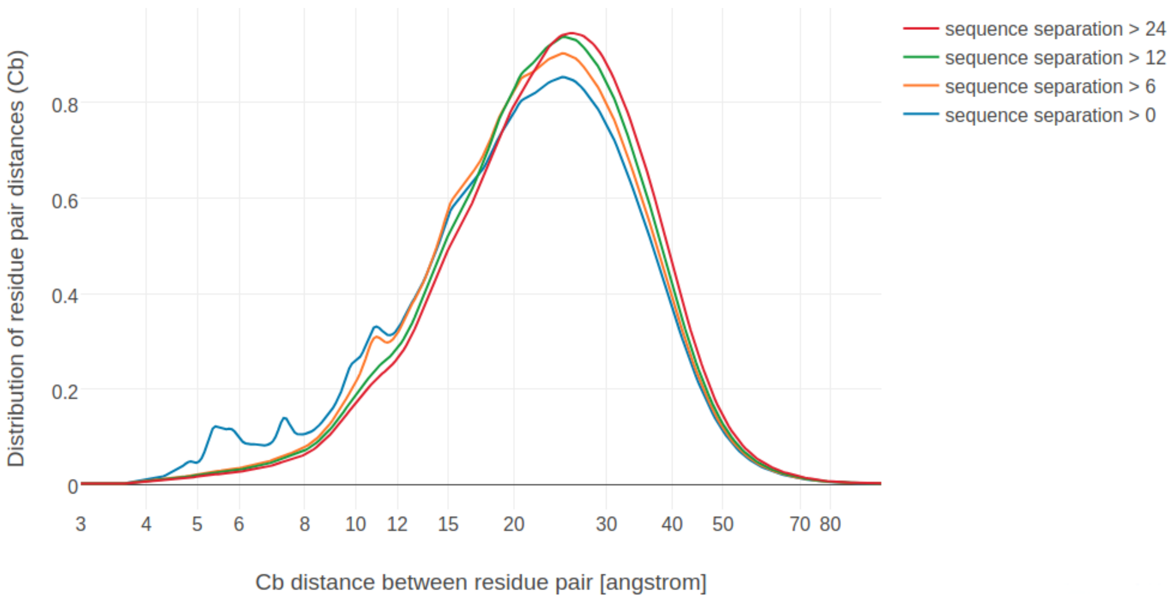


Figure 1.9: Distribution of residue pair C_β distances over 6741 proteins in the data set (see Methods 2.6.1) at different minimal sequence separation thresholds.

$$\text{mean error} = \frac{\text{error}}{TP + FP} \begin{cases} error = \Delta C_\beta - T & \text{if } \Delta C_\beta > T \\ error = 0, & \text{otherwise} \end{cases} \tag{1.18}$$

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 [91].

Precision of residue-residue contact predictions correlates with the number of diverse sequences in the multiple sequence alignment [66,91]. In the latest CASP12 competition, the best methods achieved average precisions of over 80% for the top L/2 ranked predictions for targets with several thousand diverse sequence homologs [176]. For coevolution methods the precision drops gradually as less sequence homologs are available. In contrast, methods trained on general sequence features are generally more robust. Their performance is less dependent on the number of available sequence homologs and therefore they can outperform pure coevolution methods in low data ranges [72,177].

1.5.1 Sequence Separation

Local residue pairs separated by only some positions in sequence (e.g $|i-j| < 6$) are usually filtered out for evaluating contact prediction methods. They are trivial to predict as they typically correspond to contacts within secondary structure elements and reflect the local geometrical constraints. Figure 1.9 shows the distribution of C_β distances for various minimal sequence separation thresholds. Without filtering local residue pairs (sequence separation 1), there are several additional peaks in the distribution around $5.5\mathring{A}$, $7.4\mathring{A}$ and $10.6\mathring{A}$ that can be attributed to local interactions in e.g. helices (see Figure 1.10).

Commonly, sequence separation bins are applied to distinguish short ($6 < |i-j| \leq 12$), medium ($12 < |i-j| \leq 24$) and long range ($|i-j| > 24$) contacts [91,174]. Especially long

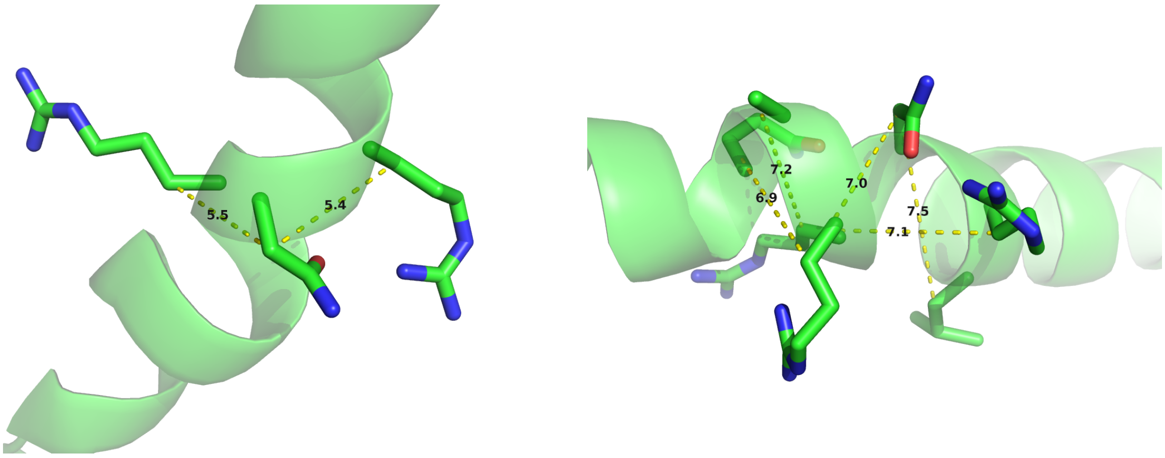


Figure 1.10: C_β distances between neighboring residues in α-helices. **Left**: Direct neighbors in α-helices have C_β distances around $5.4\mathring{A}$ due to the geometrical constraints from α-helical architecture. **Right**: Residues separated by two positions ($|i-j|=2$) are less geometrically restricted to C_β distances between $7\mathring{A}$ and $7.5\mathring{A}$.

range contacts are of importance for structure prediction as they are the most informative and able to constrain the overall fold of a protein [173].

1.5.2 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 side chains. More importantly, interactions between side chains depend on their physico-chemical properties, on their orientation and different environments within proteins [178]. 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 [46]. 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 [179].

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 homologous structure [179]. 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 precision 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 [40,45,81]. For example, given

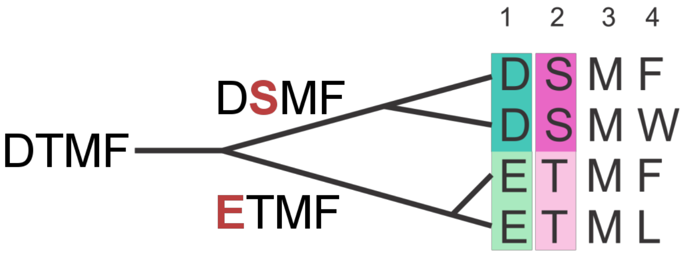


Figure 1.11: The phylogenetic dependence of closely related sequences can produce covariation signals. Here, two independent mutation events (highlighted in red) in two branches of the tree result in a perfect covariation signal for two positions.

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 overall 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 euclidean distance between true and predicted contacts, but are not commonly evaluated yet [40,145].

1.6 Challenges for Coevolution Methods

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.

1.6.1 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 over-represented [96,97,180]. 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 [59,63,64]. Figure 1.11 illustrates a simplified example, where dependence of sequences due to phylogeny leads to a covariation signal.

To reduce the effects of over-represented 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 [100]. It has been found that reweighting improves contact prediction performance [66,96,181] significantly but results are robust against the choice of the identity threshold in a range between 0.7 and 0.9 [96].

1.6.2 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 [64]. 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 [57]. Several heuristics have been proposed to reduce entropy effects, such as Row-Column-Weighting (RCW) [59] or Average Product Correction (APC) [60] (see section 1.3.6).

1.6.3 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 [103,180]. 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 [179].

Interesting targets for contact prediction are protein families without any associated structural information. As can be seen in Figure 1.12, 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 1.1) the number of protein families with enough sequences for accurate contact predictions will increase steadily [103,182]. Nevertheless, because of the already mentioned sequencing biases, better and more sensitive statistical models are indispensable to extend the applicability domain of coevolutionary methods.

1.6.4 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 [184]. 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 [179]. Therefore, MSAs need to be generated with great care and covariation methods need to be tailored to these specific types of proteins [185,186].

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 [175,187]. However, deep alignments can capture coevolutionary signals from different subfamilies [150]. 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 [170]. Therefore, a trade-off between specificity and diversity of the alignment is required to reach optimal results [120].

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 behaviors which can result in gap-induced artefacts [111].

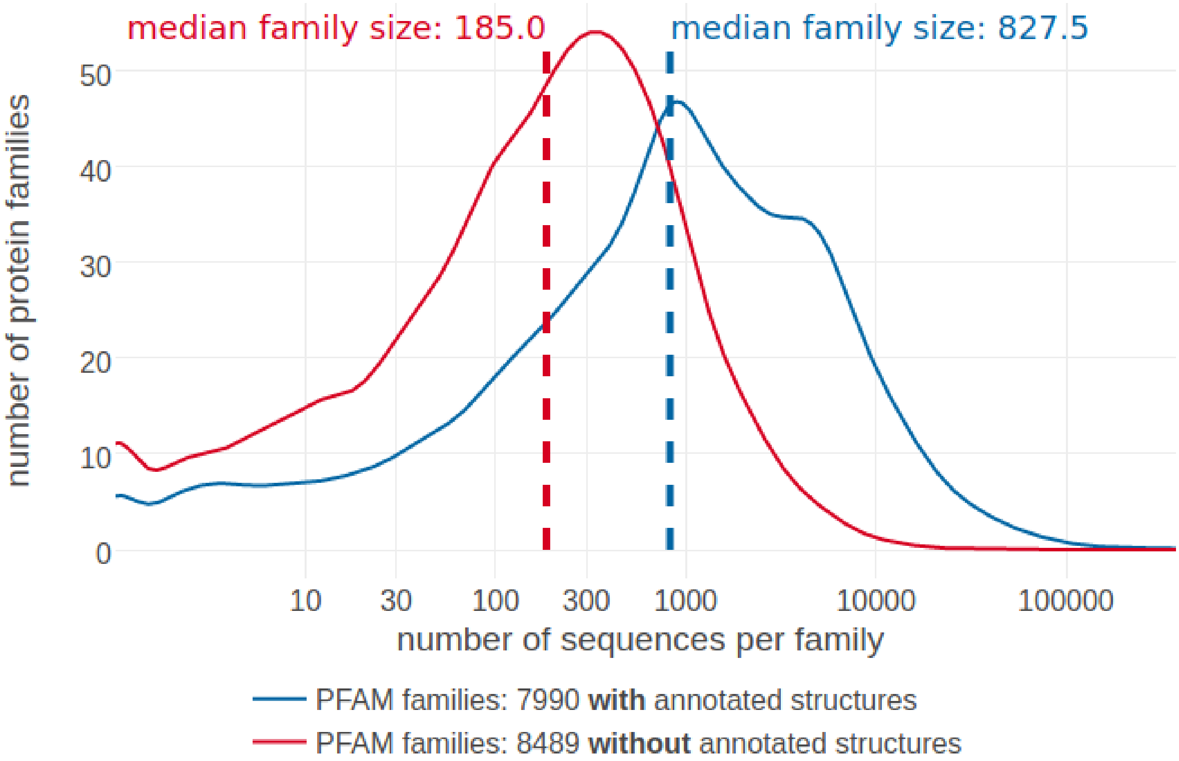


Figure 1.12: Distribution of PFAM family sizes. Less than half of the families in PFAM (7990 compared to 8489 families) do not have an annotated structure. The median family size in number of sequences for families with and without annotated structures is 185 and 828, respectively. Data taken from PFAM 31.0 (March 2017, 16712 entries) [183].

1.6.5 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 1.13) [180]. 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 1.13 b). Currently it is hard to reliably distinguish intra- and inter-molecular contacts [150]. Anishchenko et al. found that approximately one third of strong co-evolutionary signals between residue pairs at long distances (minimal heavy atom distance $>15\AA$) can be attributed to interactions across homo-oligomeric interfaces [179]. 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 [120,126,150,153,154,188] or used co-evolutionary signals to predict homo-dimeric complexes [151].

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 1.13 c) [181,189]. Confirming this hypothesis, a study showed that the cumulative strength of couplings for a particular residue can be used to predict functional sites [120,180].

Another important aspect is conformational flexibility (Figure 1.13 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

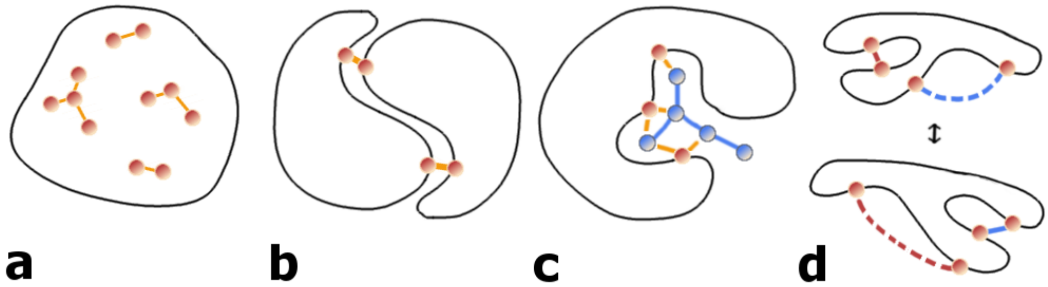


Figure 1.13: Possible sources of coevolutionary signals. **a)** Physical interactions between intra-domain residues. **b)** Interactions across the interface of predominantly homo-oligomeric complexes. **c)** Interactions mediated by ligands or metal atoms. **d)** Transient interactions (dashed lines) due to conformational flexibility.

fit manner [190]. Several studies demonstrated successfully that coevolutionary signals can capture interactions specific to different distinct conformations [96,120,152,154].

2

Interpretation of Coupling Matrices

Contact prediction methods learning a *Potts model* for the MSA of a protein familiy, map the inferred 20 x 20 dimensional coupling matrices w_{ij} onto scalar values to obtain contact scores for each residue pair as outlined in section 1.3.6. 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.

2.1 Single Coupling Values Carry Evidence of Contacts

Given the success of DCA methods, it is clear that the inferred couplings $\mathbf{w}_{ij}$ are good indicators of spatial proximity for residue pairs. As described in section 1.3.6, 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 2.1 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 2.6.6). 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 2.1. 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 2.2 show the correlation of raw coupling

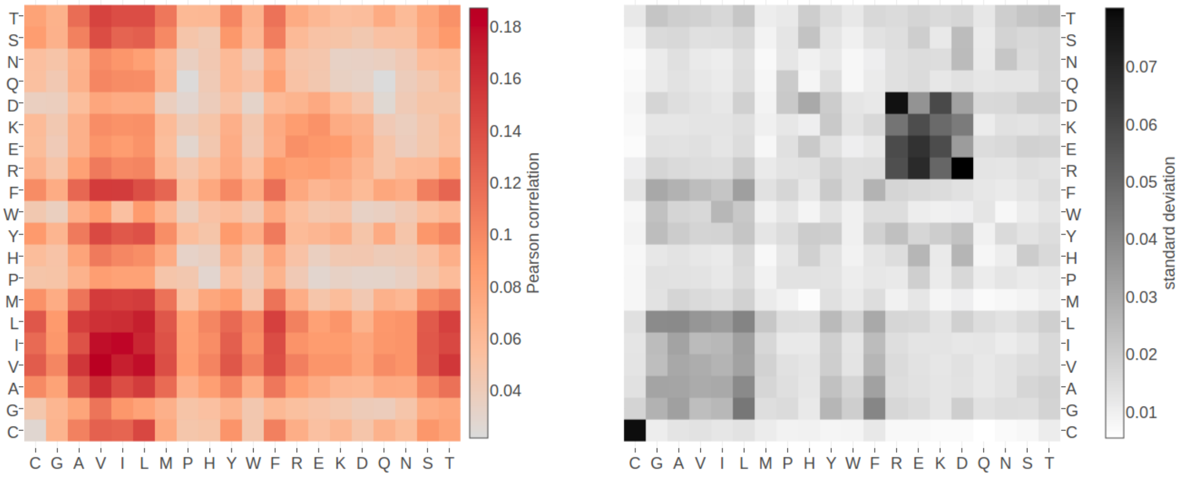


Figure 2.1: **Left** Pearson correlation of squared coupling values $(w_{ijab})^2$ with contact class (contact=1, non-contact=0). **Right** Standard deviation of squared coupling values for residue pairs in contact. Dataset contains 100.000 residue pairs per class (for details see methods section 2.6.6). Amino acids are abbreviated with one-letter code and 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.1). 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.

2.2 Coupling Profiles Vary with Distance

Analyses in the previous section showed that certain coupling values correlate more or less strong with contact class.

More insights can be obtained by looking at the distribution of distinct coupling values for contacts, non-contacts and arbitrary populations of residue pairs. Figure 2.3 shows the distribution of selected couplings for filtered residue pairs with $C_\beta - C_\beta$ distances $< 5\mathring{A}$ (see methods section 2.6.7 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

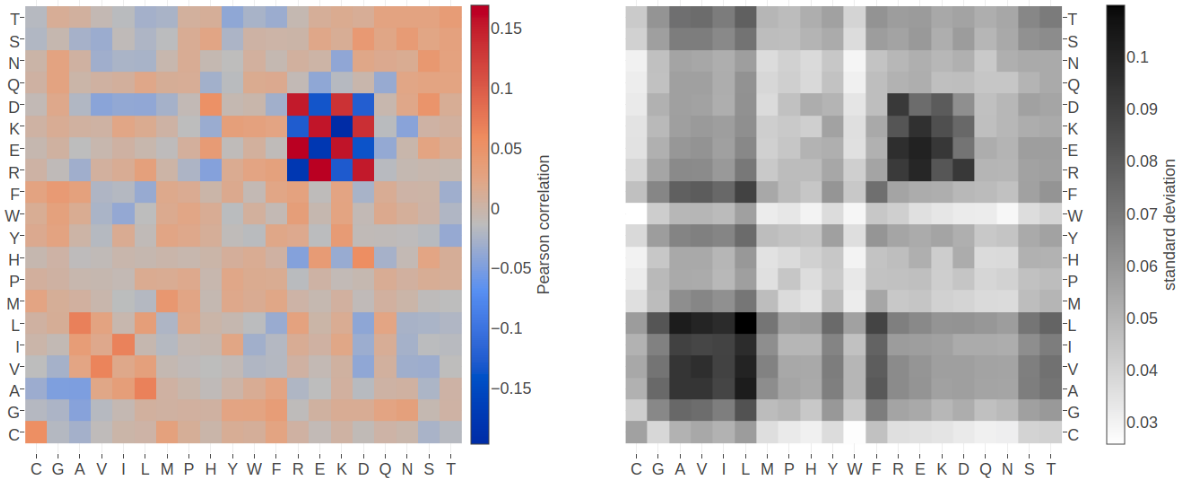


Figure 2.2: **Left** Pearson correlation of raw signed coupling values w_{ijab} with contact class (contact=1, non-contact=0). **Right** Standard deviation of coupling values for residue pairs in physical contact. Dataset contains 100.000 residue pairs per class (for details see section 2.6.6). Amino acids are abbreviated with one-letter code and they are broadly grouped with respect to physico-chemical properties listed in Appendix B.

positive values, reflecting the strong signals obtained from covalent disulphide bonds. The broad distribution for C-C, R-E and E-E agrees with the observation in section 2.1 that these specific coupling values have large standard deviations and that for charged residue pairings the signed coupling value is a strong indicator of a contact.

Hydrophobic pairs like V-I have an almost symmetric coupling distribution, confirming the finding that the direction of coupling is not indicative of a true contact whereas the strength of the coupling is. The hydrophobic effect that determines hydrophobic interactions is not specific or directed. Therefore, hydrophobic interaction partners can commonly be substituted by other hydrophobic residues, which explains the not very pronounced positive coupling signal compared to more specific interactions, e.g ionic interactions. 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 left plot in Figure 2.4). 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 $8\AA$ and $12\AA$ 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 $< 5\AA$. 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 2.6 shows the distribution of selected couplings for residue pairs far apart in the protein

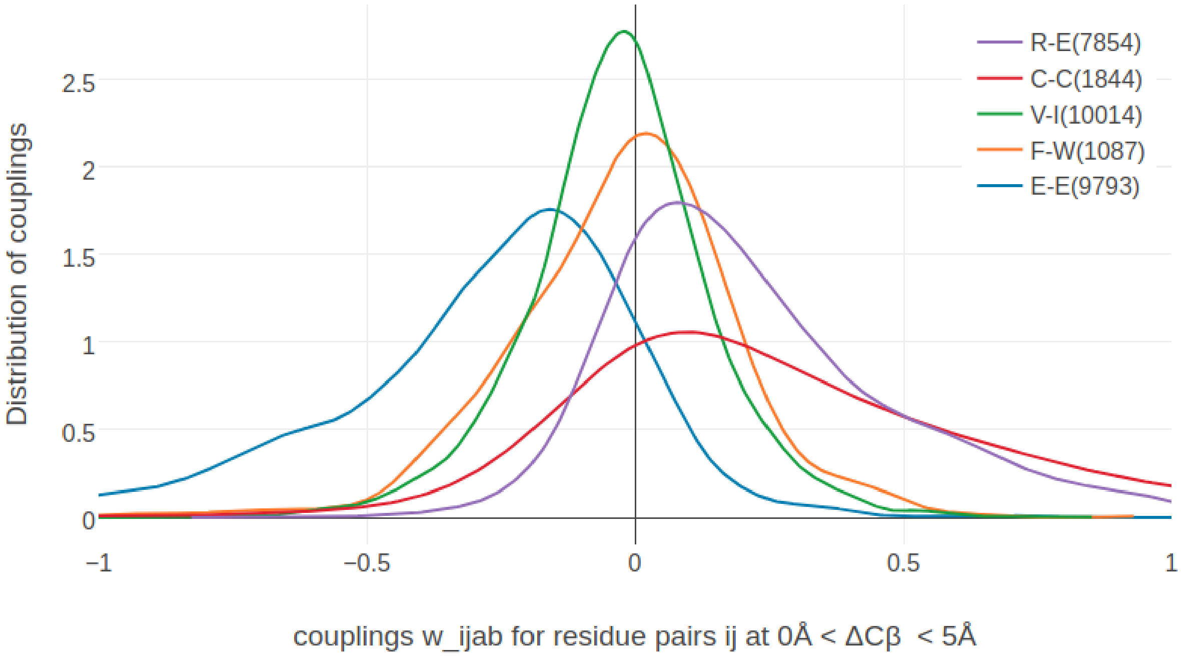


Figure 2.3: Distribution of selected couplings for filtered residue pairs with $C_\beta - C_\beta$ distances $< 5\mathring{A}$ (see methods section 2.6.7 for details). Number of coupling values used to determine the distribution is given in brackets in the legend. R-E = couplings for arginine and glutamic 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.

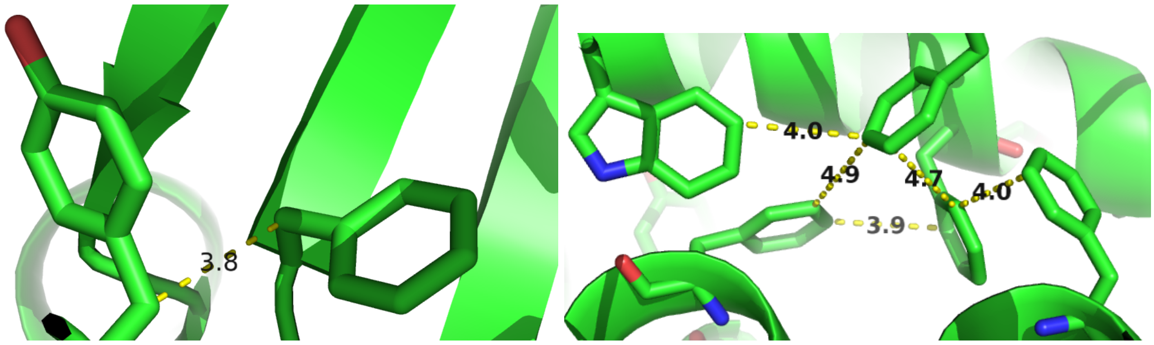


Figure 2.4: Peculiarities of aromatic residues. **Left** The planar ring system of aromatic sidechains at short C_β-C_β distances (e.g. $\Delta C_\beta < 5\mathring{A}$) often points away from each other to avoid steric hindrance. **Right** Network-like structure of aromatic residues in the protein core. 80% of aromatic residues are involved in such networks that are important for protein stability [191].

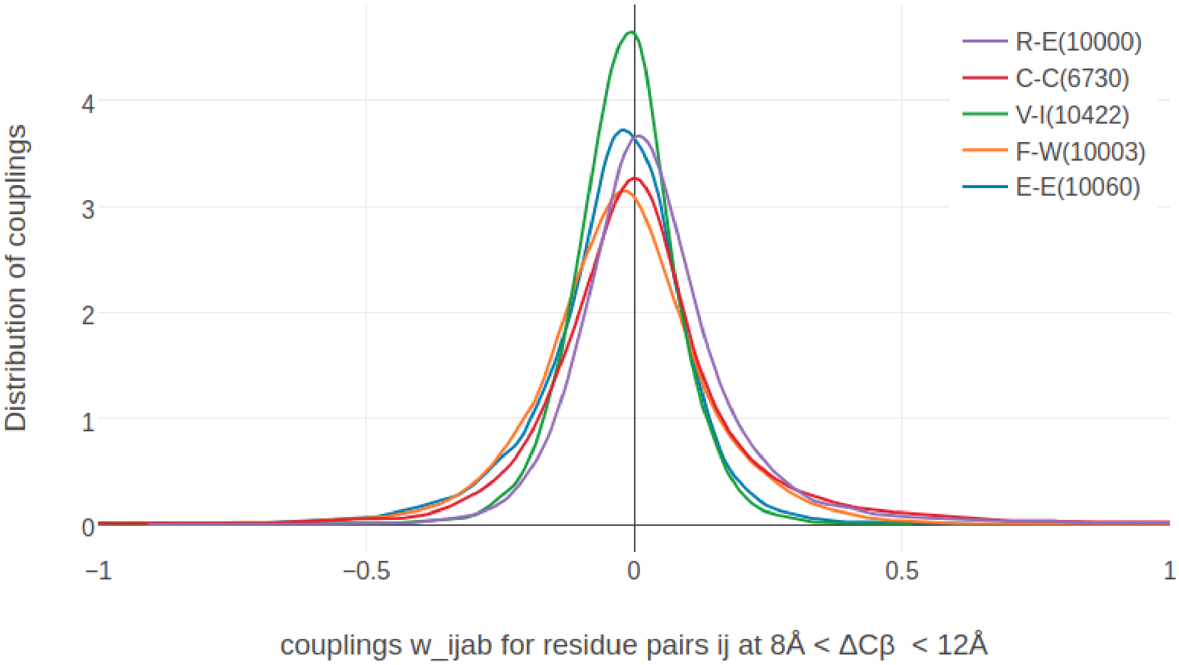


Figure 2.5: Distribution of selected couplings for filtered residue pairs with $C_\beta - C_\beta$ distances between $8\mathring{A}$ and $12\mathring{A}$ (see methods section 2.6.7 for details). Number of coupling values used to determine the distribution is given in brackets in the legend. Couplings are the same as in Figure 2.3.

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 2.1 that C-C coupling values, albeit having large standard-deviations, correlate only weakly with contact class. The same arguments apply to couplings of aromatic pairs that have a comparably broad distribution and do not correlate strongly with 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 [191]. An example is given in the right plot in Figure 2.4. 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.

2.3 Physico-Chemical Fingerprints in Coupling Matrices

The previous analysis showed that individual couplings have characterstic distributions that reflect the biophysical and steric interaction properties between amino acids. Individual coupling matrices for a residue pair that is in physcial contact often display striking patterns that agree with these findings. These patterns allow a biological interpretation of the coupling values that reveal details of the physico-chemical interdependency between both residues.

Figure 2.7 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

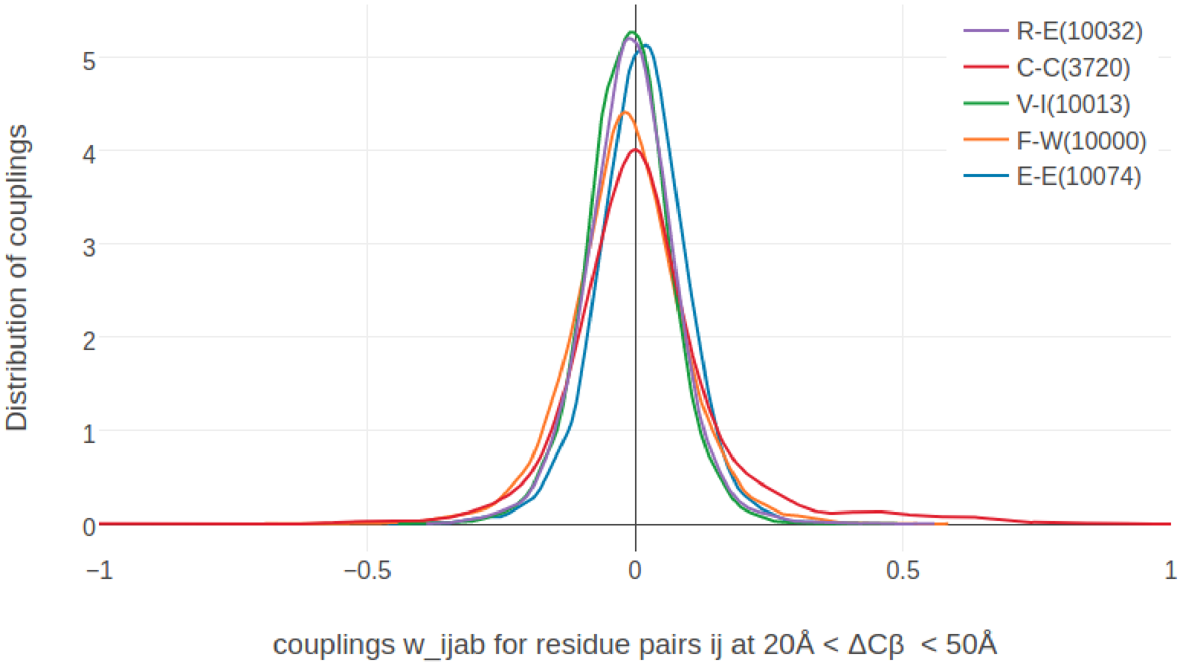


Figure 2.6: Distribution of selected couplings for filtered residue pairs with $C_\beta - C_\beta$ distances between 20$\AA$ and 50$\AA$ (see methods section 2.6.7 for details). Number of coupling values used to determine the distribution is given in brackets in the legend. Couplings are the same as in Figure 2.3.

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 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 2.9.

Figure 2.8 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 2.9 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 Figure 2.10), aromatic-proline interactions (see Figure 2.11), or disulphide bonds (see Figure 2.12).

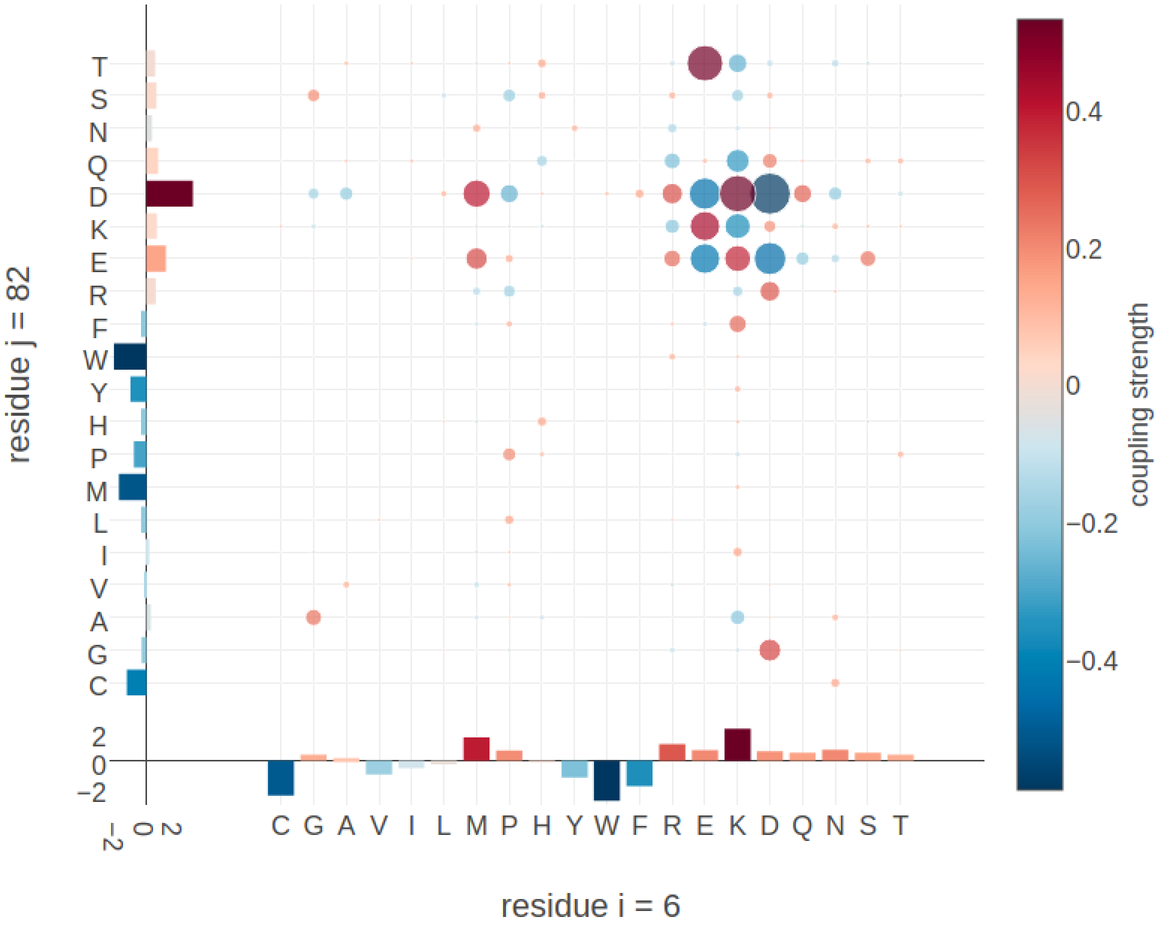


Figure 2.7: Couplings w_{ijab} and single potentials v_{ia} and v_{ja} computed with pseudo-likelihood for residues 6 and 82 in the carbamoyl phosphate synthetase protein (PDB id 1a9x_A domain 5). The matrix shows the 20x20 couplings w_{ijab} with color representing coupling strength and direction (red = 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.

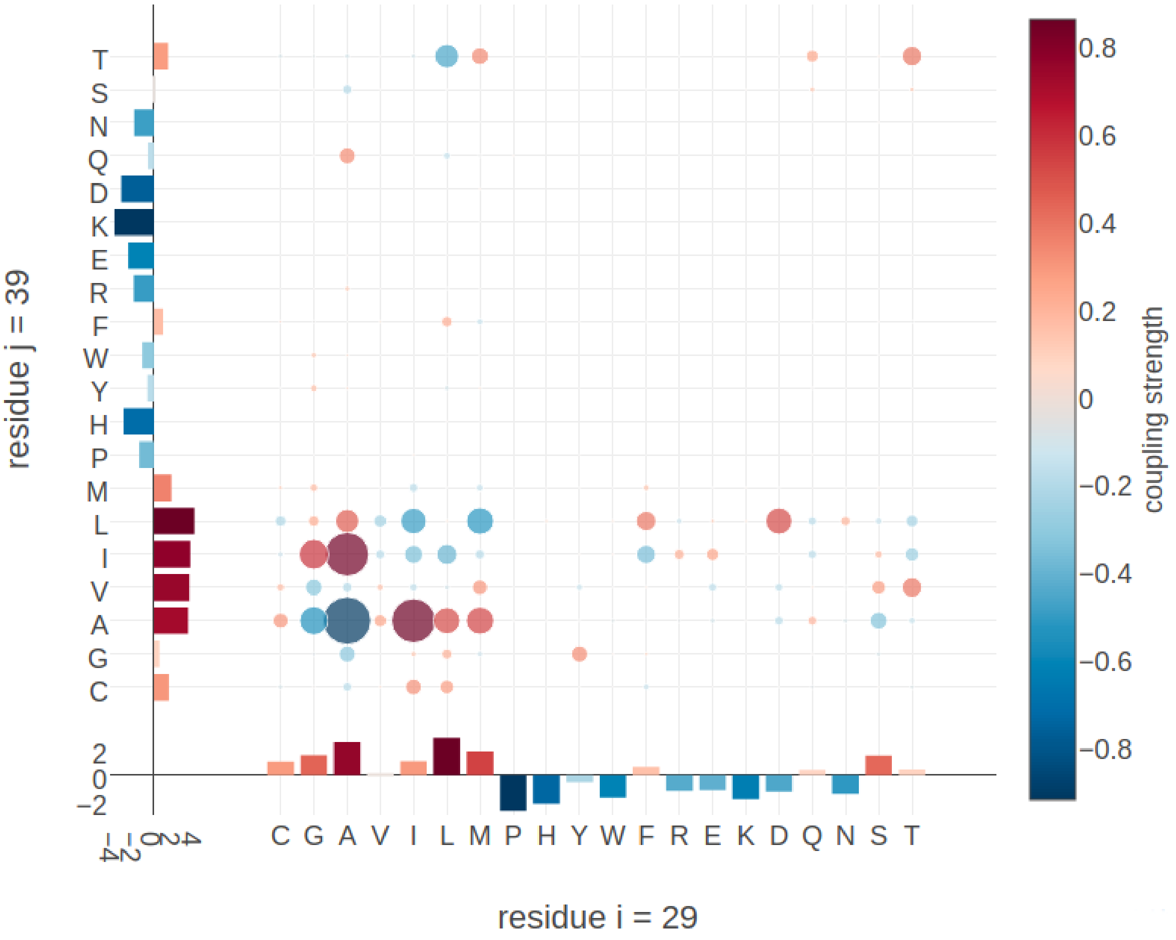


Figure 2.8: Couplings w_{ijab} and single potentials v_{ia} and v_{ja} computed with pseudo-likelihood for residues 29 and 39 in the lambda integrase protein (PDB id 1ae9_A). The matrix shows the 20x20 couplings w_{ijab}. Bars at the x-axis and y-axis correspond to the *Potts* model single potentials v_i and v_j respectively. Color coding is the same as in Figure 2.7

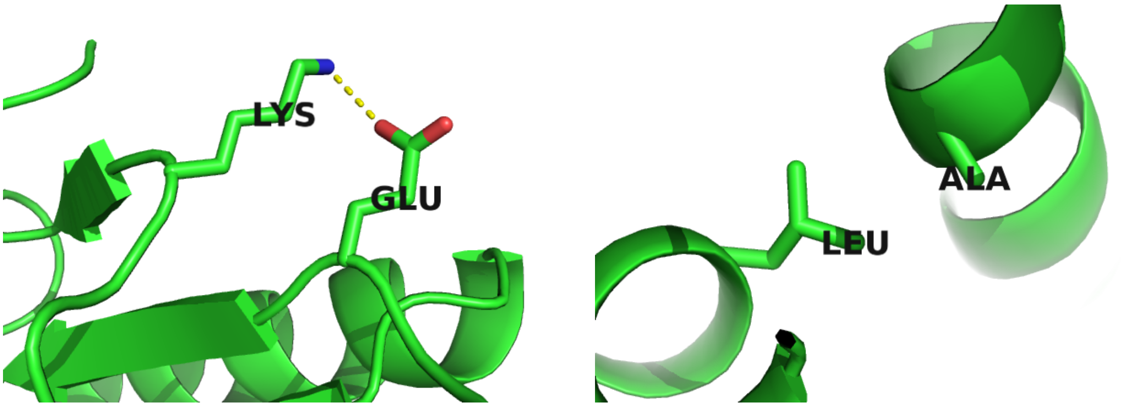


Figure 2.9: Interactions between protein side chains. **Left**: Glutamic acid (residue 6) forms a salt bridge with lysine (residue 82) in the carbamoyl phosphate synthetase protein (PDB id 1a9x_A domain 5). **Right**: Alanine (residue 29) and leucine (residue 39) within the hydrophobic core of the lambda integrase protein (PDB id 1ae9_A).

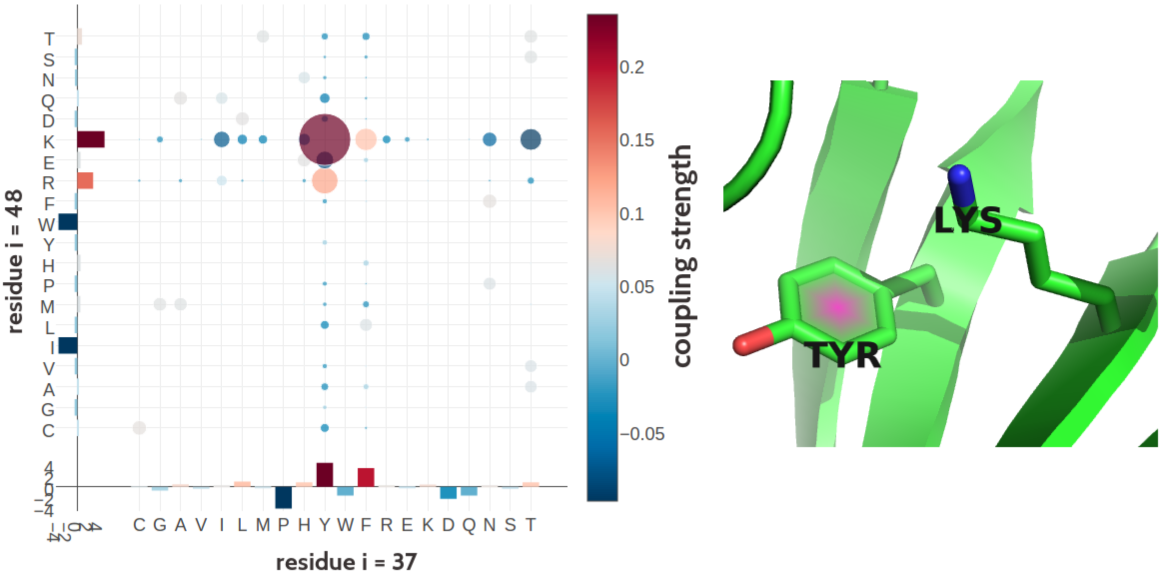


Figure 2.10: Tyrosine (residue 37) and Lysine (residue 48) forming a cation-π interaction in the C-terminal WRKY domain of Arabidopsis thaliana (PDB id 2ayd_A). **Left** Coupling matrix $\mathbf{w}_{ij}$ for residue $i = 37$ and residue $j = 48$. The matrix shows the 20x20 couplings w_{ijab}. Bars at the x-axis and y-axis correspond to the *Potts* model single potentials v_i and v_j respectively. Color coding is the same as in Figure 2.7 **Right** Cation-π interaction between Tyrosine (residue 37) and Lysine (residue 48).

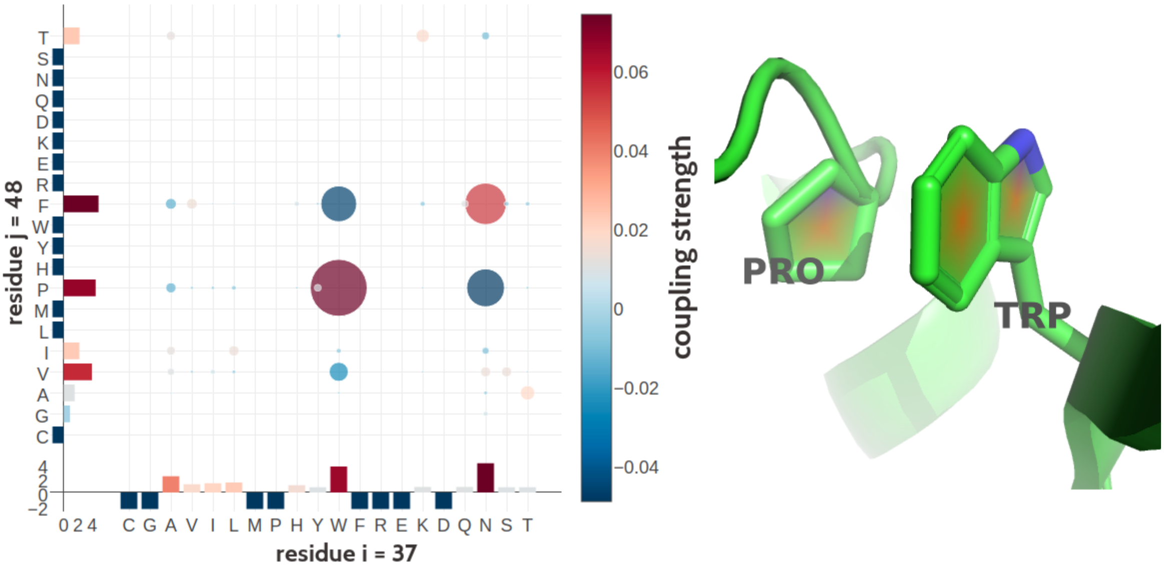


Figure 2.11: Proline and tryptophan (residues 17 and 34) forming a CH/π interaction in the murine leukemia virus receptor-binding glycoprotein (PDB id 1aol_A). **Left** Coupling matrix $\mathbf{w}_{ij}$ for residue $i = 17$ and residue $j = 34$. The matrix shows the 20x20 couplings w_{ijab}. Bars at the x-axis and y-axis correspond to the *Potts* model single potentials v_i and v_j respectively. Color coding is the same as in Figure 2.7 **Right** Proline (residues 17) and tryptophan (residues 34) stacked on top of each other engaging in a CH/π interaction.

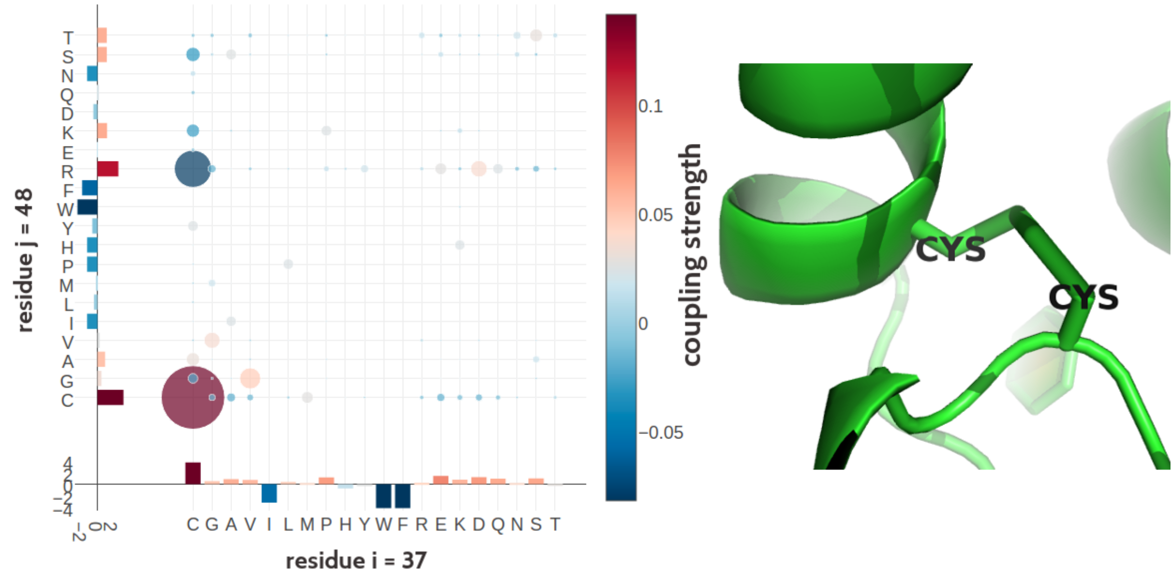


Figure 2.12: Two cystein residues (residues 54 and 64) forming a covalent disulfide bond in human interleukin-6 (PDB id 1alu_A). **Left** Coupling matrix $\mathbf{w}_{ij}$ for residue $i = 54$ and residue $j = 64$. The matrix shows the 20x20 couplings w_{ijab}. Bars at the x-axis and y-axis correspond to the *Potts* model single potentials v_i and v_j respectively. Color coding is the same as in Figure 2.7 **Right** Disulfide bond between the cystein residues 54 and 64 in the structure.

2.4 Higher Order Dependencies Between Couplings

The analyses in the last section showed that the contact matrices for residue pairs in physical contact often contain informative patterns regarding the underlying structural constraint. Therefore it can be expected that there are biological meaningful inderdependencies between individual coupling values 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 2.13 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 2.13 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 2.13).

2.5 Discussion

The analysis in this chapter proved that the 20x20 dimensional coupling matrices $\mathbf{w}_{ij}$ contain a wealth of information that is irretrievably lost when computing the heuristic contact score in form of the Frobenius norm of the coupling matrix. For several amino acid pairs (e.g. E-R, E-E) the direction of the corresponding coupling value is a strong indicator for a contact. More quantitatively, the distribution of individual couplings reflect physico-chemical interaction preferences between amino acids. Furthermore, characteristic patterns in the coupling matrices often point at the undelrying structural constraint that is subject to evolutionary pressure. The patterns also illustrate that there are higher order dependencies between the individual coupling values that also are in accordance with physico-chemical interaction preferences between amino acids.

Coucke and collegues performed a thorough quantitative analysis of coupling matrices selected from confidently predicted residue pairs [192]. 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.

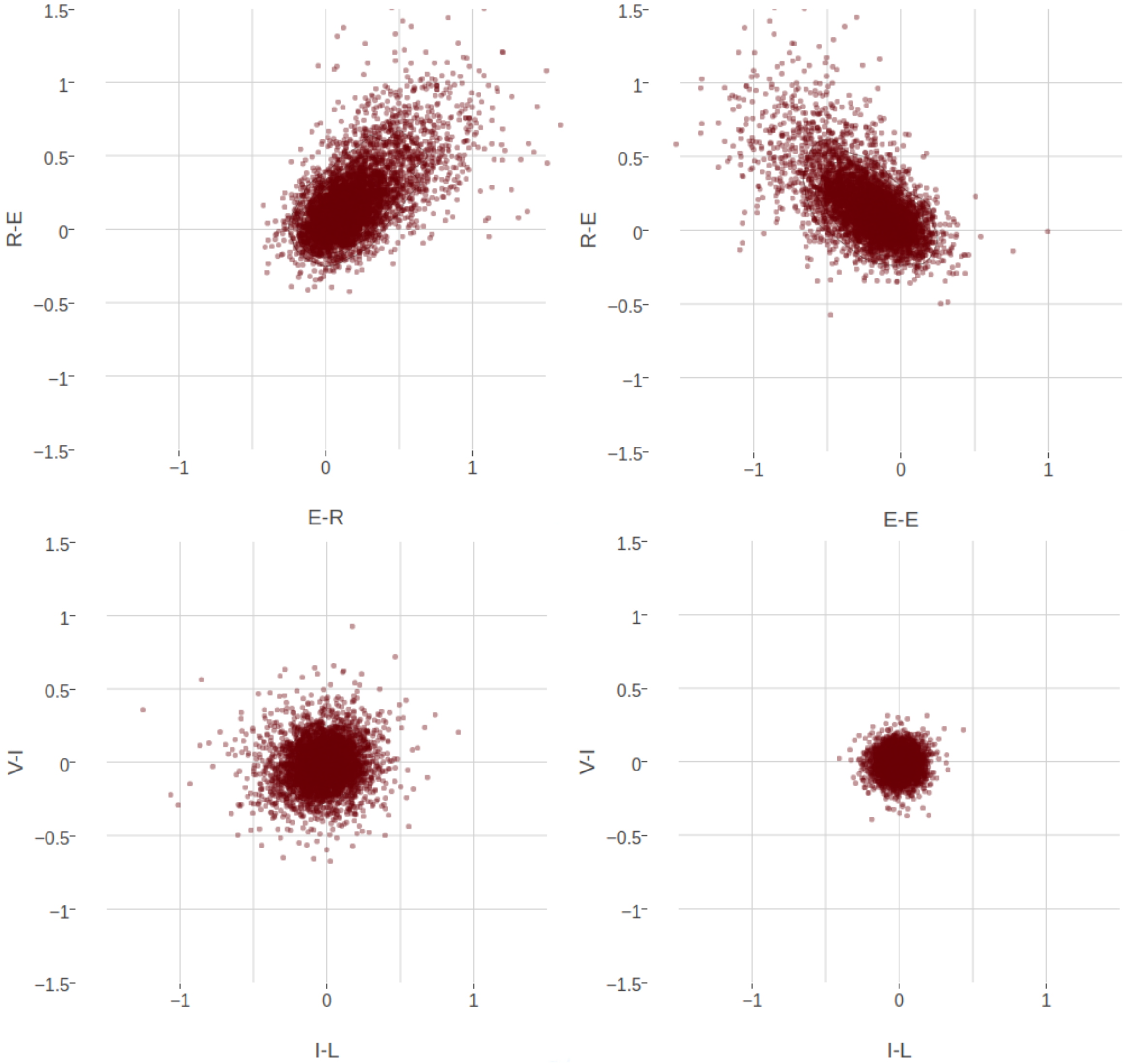


Figure 2.13: Two-dimensional distribution of approximately 10000 coupling values computed with pseudo-likelihood. **Top Left** The 2-dimensional distribution of couplings E-R and R-E for residue pairs with $C_\beta - C_\beta$ distances $< 8\mathring{A}$ is almost symmetric and the coupling values are 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.

2.6 Methods

2.6.1 Dataset

A protein dataset has been constructed from the CATH (v4.1) [193] 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[184] 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 [184] 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 2.14).

2.6.2 Computing Pseudo-Likelihood Couplings

Dr Stefan Seemayer has reimplementated the open-source software CCMpred [101] 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 is available upon request at `https://bitbucket.org/svorberg/ccmpred-new`. All computations in this thesis are performed with CCMpredPy unless stated otherwise.

CCMpredPy differs from CCMpred [101] which is available at `https://github.com/soedinglab/CCMpred` in several details:

Initialization of potentials **v** and **w**: - CCMpred initializes single potentials $\mathbf{v}_i(a) = \log f_i(a) - \log f_i(a = "-")$ with $f_i(a)$ being the frequency of amino acid a at position i and $a = "-"$ representing a gap. A single pseudo-count has been added before computing the frequencies. Pair potentials **w** are intialized at 0. - CCMpredPy initializes single potentials **v** with the

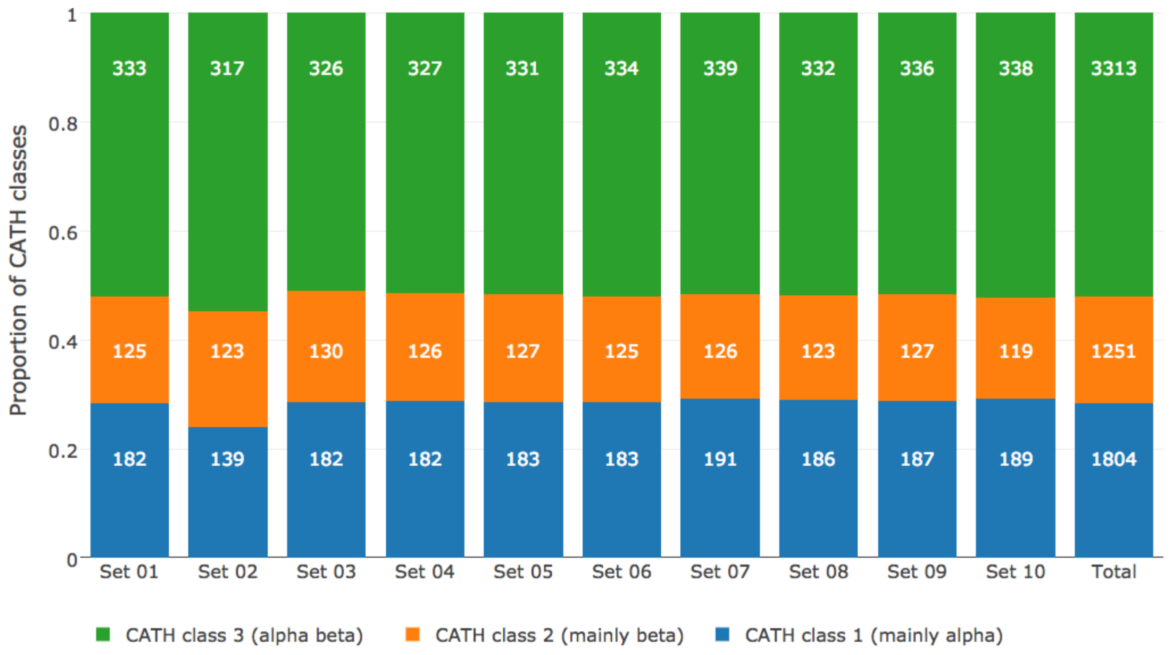


Figure 2.14: Distribution of CATH classes (1=mainly α, 2=mainly β, 3=$\alpha-\beta$) in the dataset and the ten subsets.

ML estimate of single potentials (see section 3.8.4) using amino acid frequencies computed as described in section 2.6.4. 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 3.8.4) using amino acid frequencies computed as described in section 2.6.4. 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 2.15).

Pseudo-likelihood couplings used in this thesis have been computed with CCMPredPy using the following flags:

```
--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
```

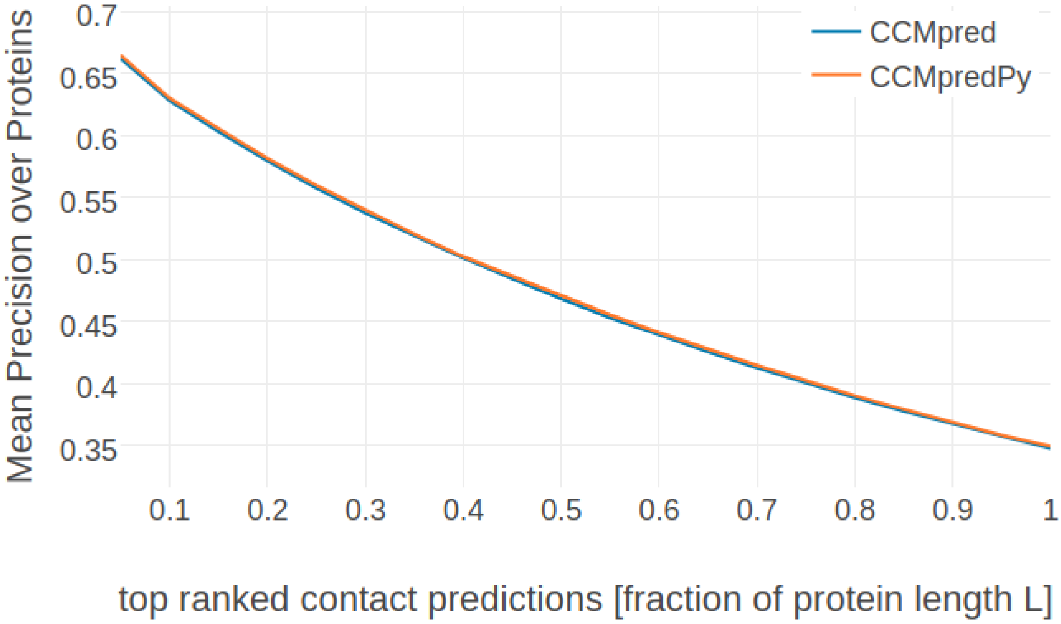


Figure 2.15: Mean precision over 3124 proteins of top ranked contacts computed as APC corrected Frobenius norm of couplings from pseudo-likelihood maximization. Couplings have been computed with CCMpred [101] and CCMpredPy as specified in the legend. Specific flags that have been used to run both methods are described in detail in the text (see section 2.6.2).

```
                    # (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
```

For the comparison of CCMpred and CCMPredPy in Figure 2.15, CCMpred was run with the following flags:

```
-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
```

2.6.3 Sequence Reweighting

As discussed in section 1.6.1, sequences in a MSA do not represent independent draws from a probabilistic model. To reduce the effects of redundant sequences, a popular sequence reweighting strategy has been found to improve contact prediction performance. 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:

$$w_n = \frac{1}{m_n} \tag{2.1}$$

$$m_n = \sum_{m=1}^{N} I\left(ID(x_n, x_m) \geq 0.8\right) \tag{2.2}$$

$$ID(x_n, x_m) = \frac{1}{L} \sum_{i=1}^{L} I(x_n^i = x_m^i) \tag{2.3}$$

An identity threshold of 0.8 has been used for all analyses in this thesis.
The number of effective sequences $\mathbf{N}_{\text{eff}}$ of an alignment is then the number of sequence clusters computed as:

$$N_{\text{eff}} = \sum_{n=1}^{N} w_n \tag{2.4}$$

2.6.4 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 2.6.3 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

$$q(x_i\!=\!a) :=(1-\tau)\; q_0(x_i\!=\!a) + \tau \tilde{q}(x_i\!=\!a) \tag{2.5}$$

$$\begin{aligned} q(x_i\!=\!a, x_j\!=\!b) :=&(1-\tau)^2\; \left[q_0(x_i\!=\!a, x_j\!=\!b) - q_0(x_i\!=\!a) q_0(x_j\!=\!b)\right] + \\ & q(x_i\!=\!a)\; q(x_j\!=\!b) \end{aligned} \tag{2.6}$$

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

$$\tau = \frac{N_{\text{pc}}}{(N_{\text{eff}} + N_{\text{pc}})} \tag{2.7}$$

where $N_{pc} > 0$. The formula for $q(x_i\!=\!a, x_j\!=\!b)$ in eq (2.6) 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 $q_0(x_i\!=\!a, x_j\!=\!b) = q_0(x_i\!=\!a) q_0(x_j\!=\!b)$.

2.6.5 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,

$$\begin{aligned} R(\mathbf{v}, \mathbf{w}) &= \log\left[\mathcal{N}(\mathbf{v}|\mathbf{v}^*, \lambda_v^{-1} I) \mathcal{N}(\mathbf{w}|\mathbf{w}^*, \lambda_w^{-1} I)\right] \\ &= -\frac{\lambda_v}{2} ||\mathbf{v} - \mathbf{v}^*||_2^2 - \frac{\lambda_w}{2} ||\mathbf{w} - w^*||_2^2 + \text{const.}\,, \end{aligned} \tag{2.8}$$

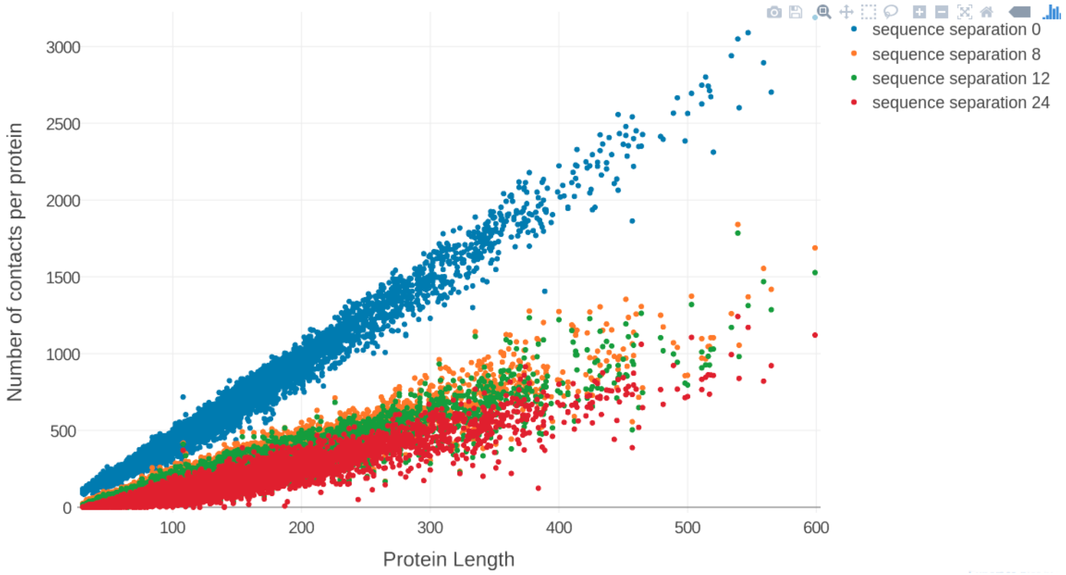


Figure 2.16: Number of contacts ($C_\beta < 8\mathring{A}$) with respect to protein length and sequence separation has a linear relationship.

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 2.16.

Most previous pseudo-likelihood approaches using L2-regularization for pseudo-likelihood optimization set $\mathbf{v}^* = \mathbf{w}^* = \mathbf{0}$ [101–103]. A different choice for v^* is discussed in section 3.8.4 that is is used per default with *CCMpredPy*. The single potentials will not be optimized when using contrastive divergence (CD) but will be fixed at v^* given in eq. (3.27). Furthermore, *CCMpredPy* uses regularization coefficients $\lambda_v = 10$ and $\lambda_w = 0.2(L-1)$ for pseudo-likelihood optimization and the choice for λ_w used with CD is discussed in section 3.3.

2.6.6 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: $> 25\mathring{A}$

2.6.7 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 2.6.4)

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 $N_{ij} \cdot q_i(a) \cdot q_j(b) > 80$.

3 Optimizing the Full Likelihood

Section 1.3 introduced the *Potts model* for contact prediction that is able to distinguish between directly and indirectly coupled residue pairs by jointly modelling the probability 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 predicting residue-residue contacts (see section 1.3.5). 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 1.12).

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 [194]. This section elaborates on how *contrastive divergence* (CD) 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 3.8.1.

3.1 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

$$\frac{\partial LL_{\text{reg}}}{\partial w_{ijab}} = N_{ij} q(x_i = a, x_j = b) - N_{ij} \, p(x_i = a, x_j = b | \mathbf{v}, \mathbf{w}) - \lambda_w w_{ijab} \, , \tag{3.1}$$

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 [95,195]. 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 [104]. 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 successful 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 *contrastive divergence* (CD) as an approximation to MCMC methods [194]. It was originally developed for training products of experts models but it can generally be applied to maximizing log likelihoods and has become popular for training restricted Boltzmann machines [95,196,197]. 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 even though the gradient estimate is 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 3.1. 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 (already updated) sequence positions:

$$p(\mathbf{x}_i = a | (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_L), \mathbf{v}, \mathbf{w}) \propto \exp\left(v_i(a) + \sum_{\substack{j=1 \\ i \neq j}}^{L} \mathbf{w}_{ij}(a, x_j) \right) \tag{3.2}$$

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. (3.1).

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

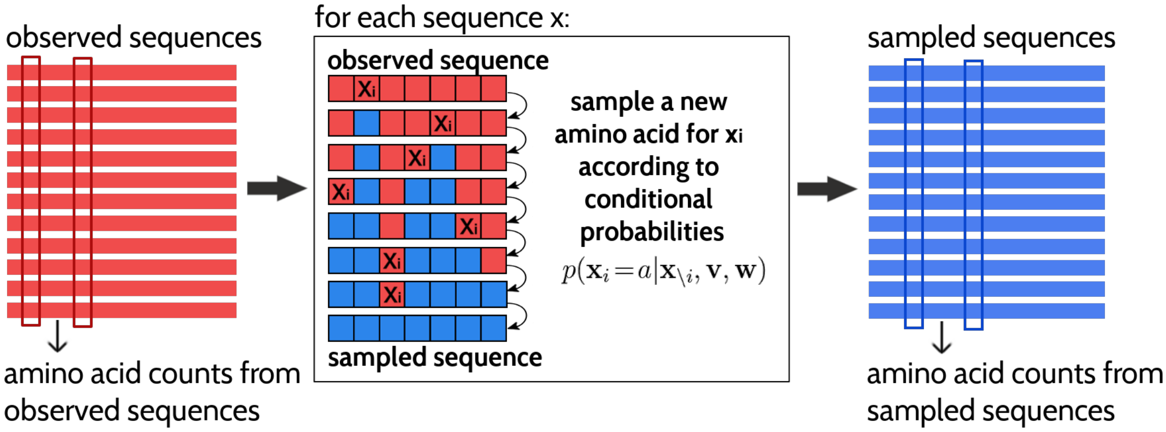

observed sequences
amino acid counts from observed sequences
for each sequence x:
observed sequence
Xi
sample a new amino acid for xi according to conditional probabilities
$p(\mathbf{x}_i = a | \mathbf{x}_{\backslash i}, \mathbf{v}, \mathbf{w})$
sampled sequence
sampled sequences
amino acid counts from sampled sequences

Figure 3.1: 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.

3.2 Optimizing the Full Likelihood

Given the likelihood gradient estimates obtained with *contrastive divergence* (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 parameterization by iteratively updating the parameters values in the opposite direction of the gradient of the objective function with respect to these parameters. *Stochastic* gradient descent (SGD) is a variant thereof that uses a 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 [198–200]. However, choosing an optimal step size for parameter updates as well as finding the optimal annealing schedule offers a challenge and needs manual tuning [201,202]. 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 3.2). 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 of the

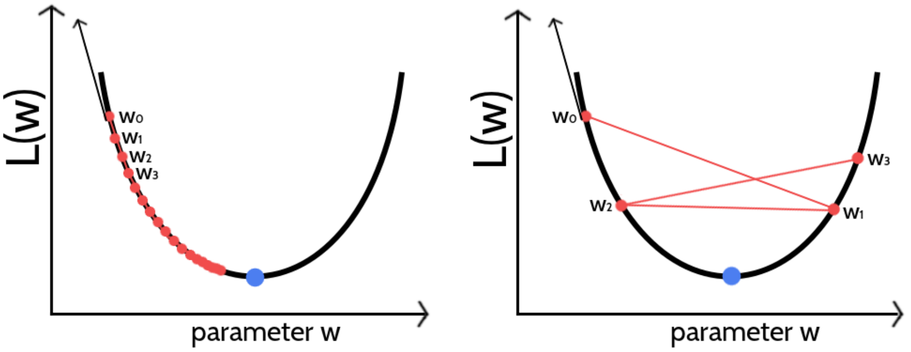


Figure 3.2: Visualization of gradient descent optimization of an objective function $L(w)$ for different step sizes α. The blue dot marks the minimum of the objective function. The direction of the gradient at the initial parameter estimate w_0 is given as black arrow. The updated parameter estimate 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.

Potts model. 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.

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.

3.2.1 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 *sontrastive divergence* (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 [203,204]. 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 [205]. However, when using a finite number of sequences for sampling, the norm of the gradient does not converge to zero but towards a certain offset as it is described in section 3.4.1. Convergence could also be monitored as the relative change of the norm of gradients within a certain number of iterations. Optimization will be stopped when the relative change becomes negligibly small, that is when the gradient norm has reached a 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

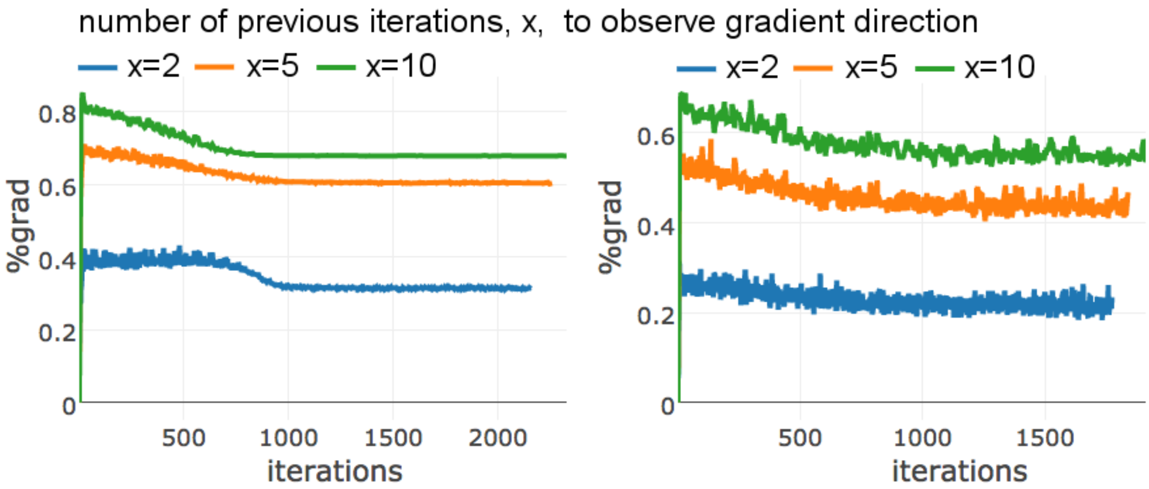


Figure 3.3: Percentage of parameters for which the derivative has changed its direction (i.e. the sign) during the previous x iterations (x is specified in the legend). Optimization is performed with SGD using the optimal hyperparameters defined in section 3.2.2 and using a regularization coefficient $\lambda_w = 0.1L$ (see section 3.3) 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 x iterations falls below the threshold of $\epsilon = 1e-8$. Development has been monitored for two different proteins, **Left** 1c75A00 (protein length = 71, number sequences = 28078, Neff = 16808) **Right** 1ahoA00 (protein length = 64, number sequences = 378, Neff = 229).

parameter estimates $||\mathbf{w}||_2$ over several iterations and stop learning when it falls below a small threshold ϵ,

$$\frac{||\mathbf{w}_{t-x}||_2 - ||\mathbf{w}_t||_2}{||\mathbf{w}_{t-x}||_2} < \epsilon \; . \tag{3.3}$$

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

For stochastic gradient descent the optimum is a moving target and the gradient will start oscillating when approaching the optimum. Therefore, another idea is to monitor the direction of the partial derivatives. 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 3.2. When optimizing high-dimensional problems using the same learning rate for all dimensions, it is likely that parameters converge at different speeds [198] leading to oscillations that could either originate from convergence or yet too large learning rates. As can be seen in Figure 3.3, the percentage of parameters for which the derivative changes direction within the last x 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.

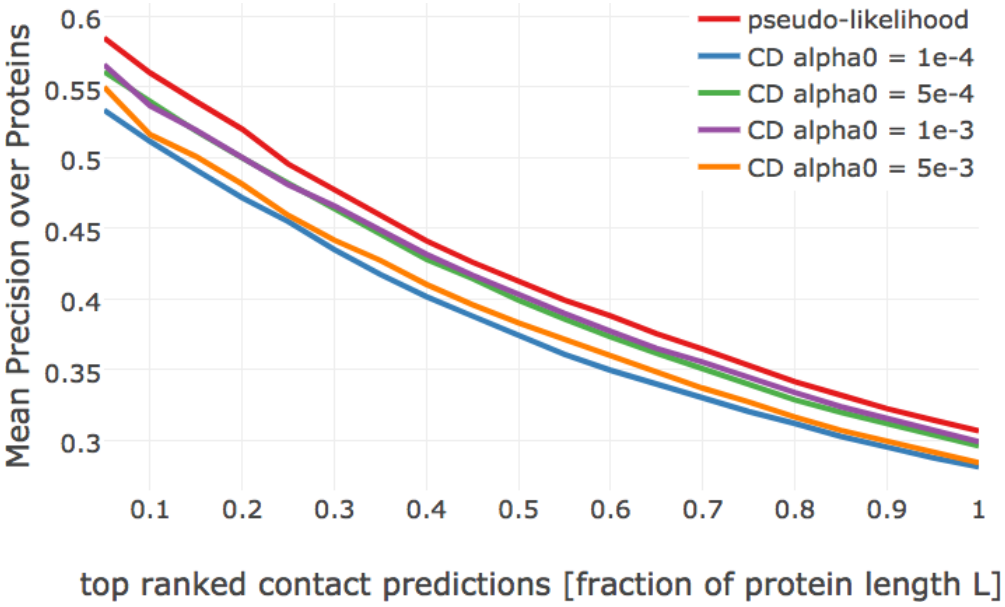


Figure 3.4: Mean precision for top ranked contact predictions over 300 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. **pseudo-likelihood**: couplings computed with pseudo-likelihood. **CD alpha0 = X**: couplings computed with CD using stochastic gradient descent with different initial learning rates α_0 (see legend).

3.2.2 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,

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \cdot \nabla_w LL_{\text{reg}}(\mathbf{v}^*, \mathbf{w}) \; . \tag{3.4}$$

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[199].

Figure 3.4 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 largest ($\alpha_0 = 5\text{e}{-3}$) learning rate.

By looking at individual proteins it becomes evident that the optimal learning rate depends on alignment size. Figure 3.5 displays the development of the L2 norm of the coupling parameters, $||\mathbf{w}||_2$, during optimization using different learning rates for two proteins with different alignment sizes. The left plot shows protein 1c75A00 that has a large alignment with 28078 sequences (Neff = 16808) while the right plot shows protein 1ahoA00 that has a small alignment with 378 sequences (Neff = 229). For protein 1ahoA00 and using a small initial learning rate $\alpha_0 = 1\text{e} - 4$, the optimization runs very slowly and does not converge within the maximum number of 5000 iterations. Using a large initial learning rate $\alpha_0 = 5\text{e} - 3$ results in slightly overshooting the optimum at the beginning of the optimization but with the learning rate decaying over time the parameter estimates converge. In contrast, for protein 1c75A00, the choice of learning rate has a more pronounced effect. With a small initial learning rate

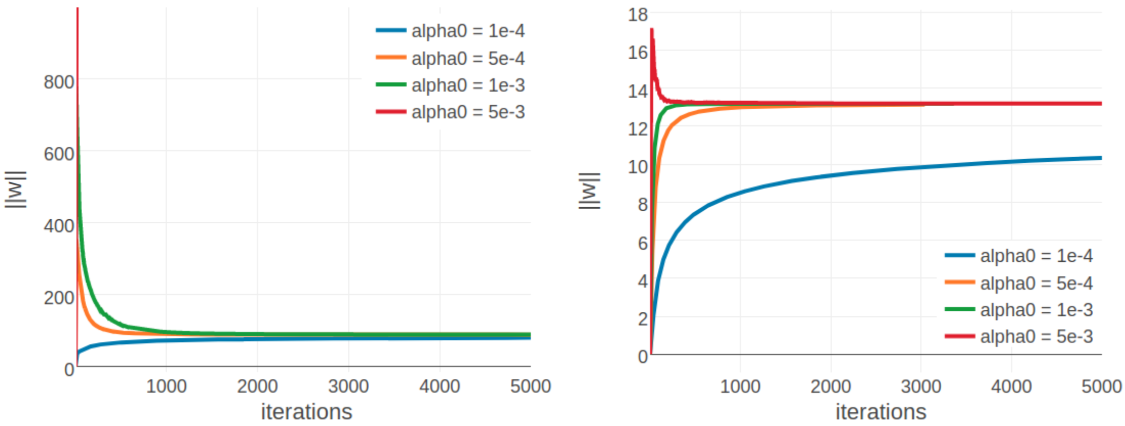


Figure 3.5: Convergence plots for two proteins during SGD optimization with different learning rates and convergence measured as L2-norm of the coupling parameters $||\mathbf{w}||_2$. Linear learning rate annealing schedule has been used with decay rate $\gamma = 0.01$ and initial learning rates α_0 have been set as specified in the legend. **Left** 1c75A00 (protein length = 71, number sequences = 28078, Neff = 16808). Figure is cut at the yaxis at $||\mathbf{w}||_2 = 1000$, but learning rate of 5e$-$3 reaches $||\mathbf{w}||_2 \approx 9000$. **Right** 1ahoA00 (protein length = 64, number sequences = 378, Neff = 229)

$\alpha_0 = 1\mathrm{e} - 4$ the optimization runs slowly but almost converges within 5000 iterations. A large initial learning rate $\alpha_0 = 5\mathrm{e} - 3$ lets the parameters diverge quickly and the optimum cannot be recovered. With learning rates $\alpha_0 = 5\mathrm{e} - 4$ and $\alpha_0 = 1\mathrm{e} - 3$, 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 before. 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,

$$\alpha_0 = \frac{5\mathrm{e}{-2}}{\sqrt{N_{\mathrm{eff}}}} \; . \tag{3.5}$$

For small Neff, e.g. 5th percentile of the distribution in the data set ≈ 50, this definition of the learning rate yields $\alpha_0 \approx 7\mathrm{e}{-3}$ and for large Neff, e.g. 95th percentile ≈ 15000, this yields $\alpha_0 \approx 4\mathrm{e}{-4}$. These values for α_0 lie in the optimal range that has been observed for the two representative proteins in Figure 3.4. With the initial learning rate defined as a function of Neff, precision slightly improves over the previous fixed learning rates (see Appendix Figure E.1). All following analyses are conducted using the Neff-dependent initial 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. (3.5):

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

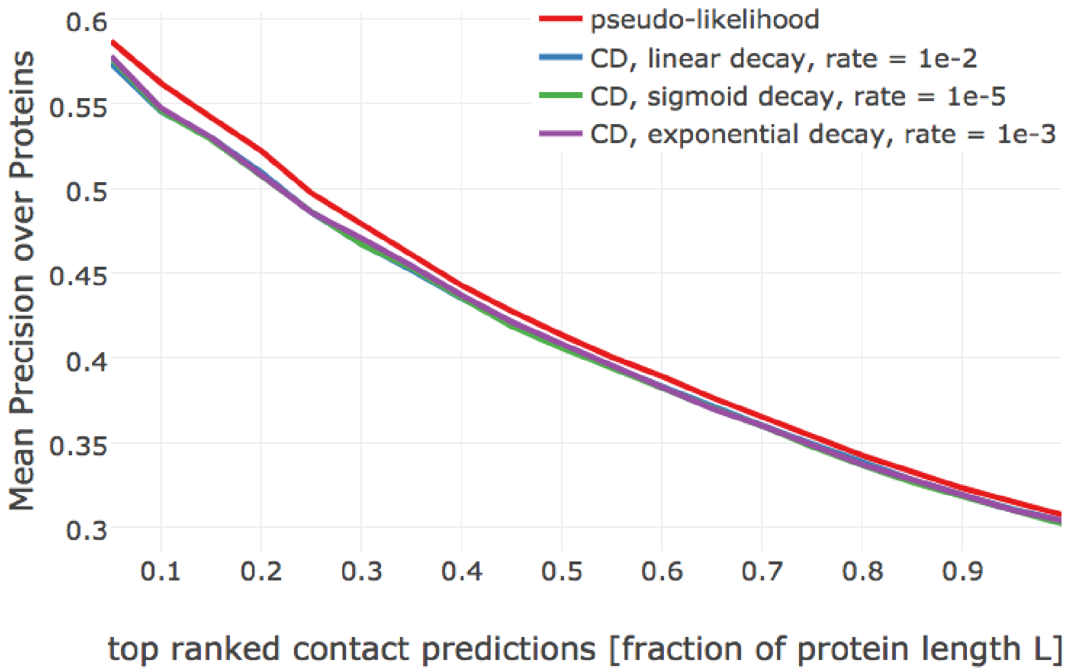


Figure 3.6: Mean precision for top ranked contact predictions over 300 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. **pseudo-likelihood**: couplings computed with pseudo-likelihood. **CD**: couplings computed with CD using stochastic gradient descent with an initial learning rate defined with respect to Neff. Learning rate annealing schedules and decay rates are specified in the legend.

The learning rate annealing schedules are visualized for different decay rates in Appendix Figure E.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 and yield a mean precision that is about one to two percentage below the mean precision for the pseudo-likelihood contact score (see Figure 3.6): a linear learning rate schedule with decay rate $\gamma = 1\mathrm{e}{-2}$, a sigmoidal learning rate schedule with decay rates $\gamma = 1\mathrm{e}{-5}$ or $\gamma = 1\mathrm{e}{-6}$ and an exponential learning rate schedule with decay rates $\gamma = 1\mathrm{e}{-3}$ or $\gamma = 1\mathrm{e}{-5}$. The square root learning rate schedule gives ovarall bad results and does not lead to convergence because the learning rate decays slowly at later time steps. The benchmark plots for all learning rate schedules are shown in Appendix Figures E.3, E.4, E.5, E.6.

In contrast to the findings regarding the initial learning rate earlier, an optimal decay rate can be defined independent of the alignment size. Figure 3.7 shows the development of the L2 norm of the coupling parameters, $||\mathbf{w}||_2$, during optimization for the same two representative proteins with small and large alignments as before. Convergence for protein 1ahoA00, having small Neff=229, is robust against the particular choice of learning rate schedule and decay rate and the presumed optimum at $||w||_2 \approx 13.2$ is reached regardless of the learning rate annealing schedule (see right plot in Figure 3.7). For protein 1c75A00, with high Neff=16808, the choice of the learning rate schedule has a notable impact on the rate of convergence. Using a linear schedule, the learning rate decays quickly but then converges to a certain offset, which effectively prevents further optimization progress and the presumed optimum at $||w||_2 \approx 90$ is not reached within 5000 iterations. Learning rate schedules that decay slower but decay continuously for 5000 iterations, such as an exponential schedule with $\gamma = 1\mathrm{e}{-3}$ or a sigmoidal schedule with $\gamma = 1\mathrm{e}{-6}$, guide the parameter estimates close to the expected optimum. Therefore, learning rate schedules with an exponential or sigmoidal decay can be

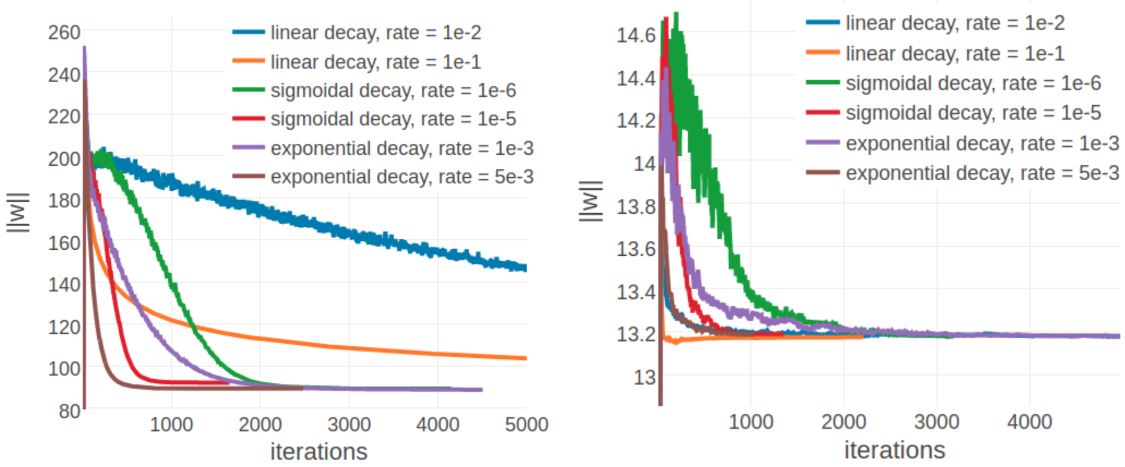


Figure 3.7: L2-norm of the coupling parameters $||\mathbf{w}||_2$ during stochastic gradient descent optimization with different learning rates schedules. The initial learning rate α_0 is defined with respect to Neff as given in eq. (3.5). Learning rate schedules and decay rates are used according to the legend. **Left** 1c75A00 (protein length = 71, number sequences = 28078, Neff = 16808). **Right** 1ahoA00 (protein length = 64, number sequences = 378, Neff = 229)

used with proteins having low Neffs as well as high Neffs.

Another aspect worth considering is run time and it can be observed that the different learning rate annealing schedules differ in convergence speed. Figure 3.8 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 $\gamma = 1\text{e}{-5}$ or an exponential learning rate annealing schedule with decay rate $\gamma = 5\text{e}{-3}$, On the contrary, the distribution of iterations until convergence has a median of 5000 when using a linear learning rate annealing schedule with $\gamma = 1\text{e}{-2}$ or an exponential schedule with decay rate $\gamma = 1\text{e}{-3}$. Under these considerations, I chose a sigmoidal learning rate schedule with $\gamma = 5\text{e}{-6}$ 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, $||\mathbf{w}||_2$, over the last 5 iterations falls below a small value $\epsilon < 1e-8$ as denoted in eq. (3.3). Figure 3.9 shows that the mean precision over proteins is robust to different settings of the number of iterations over which the relative change is computed. The convergence rate is mildly affected by the different settings. Optimization converges on average within 1697, 1782 and 1917 iterations, when computing the relative change of the parameter norm over the previous 2,5 and 10 iterations, respectively (see Appendix Figure E.11). For all following analysis, I chose 10 to be the number of iterations over which the convergence criterion is computed.

3.3 Tuning the Regularizer of Coupling Parameters

For tuning the hyperparameters of the stochastic gradient descent optimizer in the last section 3.2.2, 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 described in methods section 2.6.5. It is conceivable that CD achieves optimal performance using stronger or weaker regularization than used for pseudo-likelihood optimization. Therefore, I evaluated

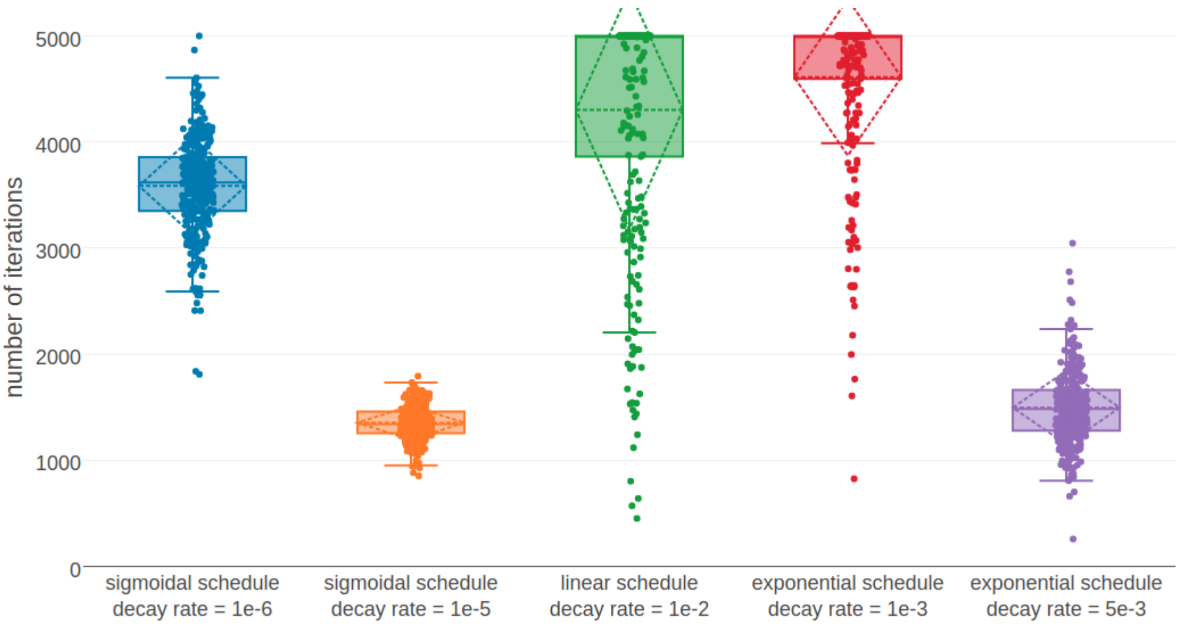


Figure 3.8: Distribution of the number of iterations until convergence for SGD optimizations of CD for different learning rate schedules. Convergence is reached when the relative difference of parameter norms, $||\mathbf{w}||_2$, over the last five iterations falls below $\epsilon = 1e-8$. Initial learning rate α_0 is defined with respect to Neff as given in eq. (3.5) and maximum number of iterations is set to 5000. Learning rate schedules and decay rates are specified in the legend.

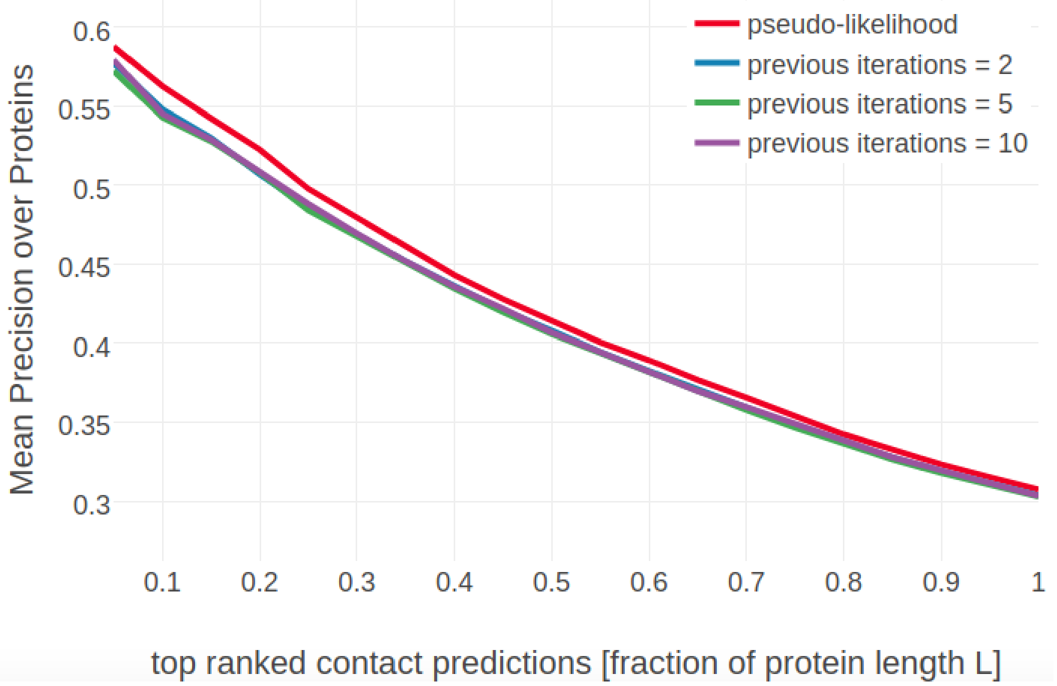


Figure 3.9: Mean precision for top ranked contact predictions over 300 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. **pseudo-likelihood**: couplings computed with pseudo-likelihood. **#previous iterations = X**: couplings computed with CD using stochastic gradient descent with an initial learning rate defined with respect to Neff and the sigmoidal learning rate schedule with $\gamma = 5e{-6}$. The relative change of the L2 norm over coupling parameters, $||\mathbf{w}||_2$, is evaluated over the previous X iterations (specified in the legend) and convergence is assumed when the relative change falls below a small value $\epsilon = 1e-8$.

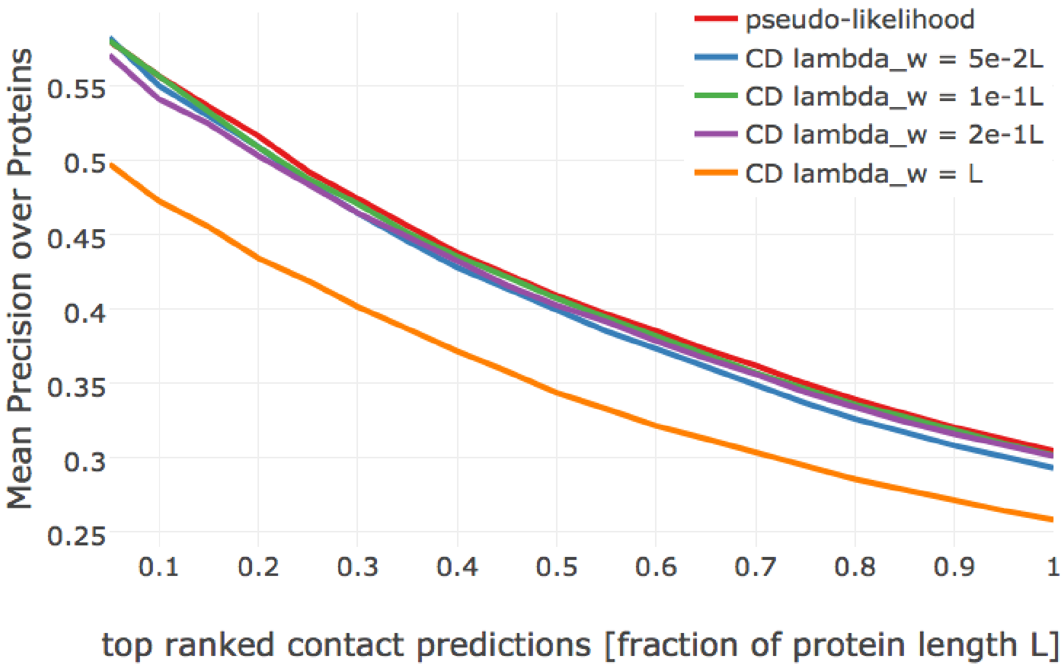


Figure 3.10: Mean precision for top ranked contact predictions over 300 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. **pseudo-likelihood**: couplings computed with pseudo-likelihood. **CD lambda_w = X**: couplings computed with CD using L2-regularization on the couplings $\mathbf{w}$ with regularization coefficient λ_w specified in the legend and keeping the single potentials v_i fixed at their MLE optimum v_i^* denoted in eq. (3.27).

performance for different regularization coefficients $\lambda_w \in \{5\text{e}{-}2L, 1\text{e}{-}1L, 1\text{e}{-}2L, L\}$ using the previously identified hyperparameters for SGD. The single potentials $\mathbf{v}$ are not subject to optimization and are kept fixed at their maximum-likelihood estimate v^* that is derived in eq. (3.27).

As can be seen in Figure 3.10, using strong regularization for the couplings, with $\lambda_w = L$, results in a drastic drop of mean precision. Using weaker regularization, with $\lambda_w = 5\text{e}{-}2L$, 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 $\lambda_w = 1\text{e}{-}1L$ than the default $\lambda_w = 1\text{e}{-}2L$ 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, in contrast to pseudo-likelihood optimization 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 (3.27). When the single potentials $\mathbf{v}$ are optimized with CD using the same regularization coefficient $\lambda_v = 10$ as for pseudo-likelihood optimization, performance is almost indistinguishable compared to keeping the single potentials $\mathbf{v}$ fixed (see Appendix Figure E.12).

3.4 Modifying 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 [194]. As described earlier, CD-1 provides a biased estimate of the true gradient because the Markov chains have not reached the stationary distribution [196]. Bengio and

Delalleau show that the bias for CD-k can 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 [197,206]. As the number of Gibbs steps, 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, even though 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 [197]. Furthermore it has been shown, that bias also depends on the mixing rate (rate of convergence) of the chains whereby the mixing rate decreases when model parameters increase [207]. This can lead to divergence of the CD-k solution from optimal solution in a sense that the model systematically gets worse as optimization progresses [208]. 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 [197]. These studies analyzing 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 using one Gibbs update step on a randomly selected variable is exactly equivalent to a stochastic maximum pseudo-likelihood estimation [209,210]. Asuncion and colleagues showed further that an arbitrary good approximation to the full likelihood can be reached by applying blocked-Gibbs sampling [211]. CD based on sampling an arbitrary number of variables, has an equivalent stochastic composite likelihood, which is a higher-order generalization of the pseudo-likelihood.

Another variant of CD is PCD, such that the Markov chain is not reinitialized at a data sample every time a new gradient is computed [207]. 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. Consequently, the Markov chains will not be pushed too far from equilibrium after each update but rather stay close to the stationary distribution [95,196,207]. Tieleman and others observed that PCD performs better than CD in all practical cases tested, even though CD can be faster in the early stages of learning and thus should be preferred when run time is the limiting factor [95,207,212].

The next sections discuss various modifications of the CD algorithm, such as 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 eventually 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.

3.4.1 Varying the Sample Size

The default Gibbs sampling scheme outlined in method section 3.8.6 involves the random selection of $10L$ sequences from the input alignment, with L being protein length, at every iteration of the optimization procedure. These selected sequences are used to initialize the same number of Markov chains. 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, at least for alignments with less than 10^3 diverse sequences [179]. It has been argued that roughly $5L$ non redundant sequences are required to obtain confident predictions that can bet used for protein structure prediction [103].

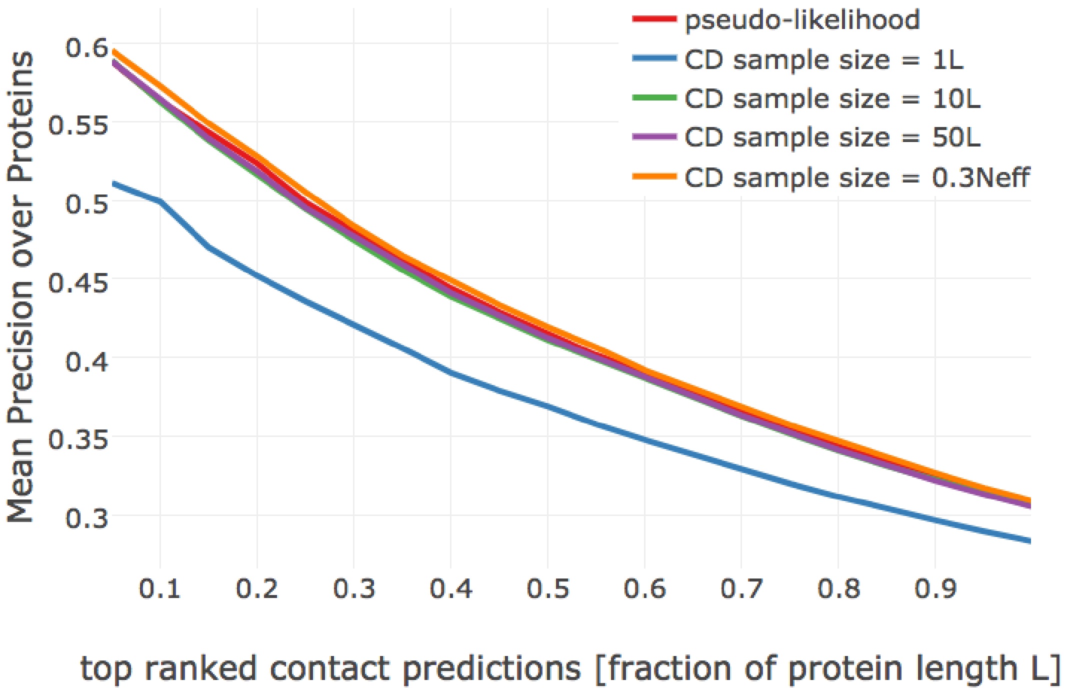


Figure 3.11: Mean precision for top ranked contact predictions over 300 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. **pseudo-likelihood**: couplings computed with pseudo-likelihood. **CD sample size** $=$ **X** : contact scores computed from CD with SGD. At every iteration, a particular number of sequences is randomly selected from the input alignment to initialize the Markov chains for Gibbs sampling. The number of randomly selected sequences is specified in the legend. It is defined either as multiples of protein length L or as fraction of the effective number of sequences Neff.

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 run time at each iteration. I evaluated different schemes for the random selection of sequences:

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

Figure 3.11 illustrates performance for several of the choices. Randomly selecting L sequences for sampling results in a visible drop in performance. There is no benefit in using more than $10L$ sequences, especially as sampling more sequences increases run time 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. By sampling 0.3Neff sequences, CD does slightly improve over pseudo-likelihood.

When evaluating performance with respect to the number of effective sequences Neff, it can clearly be noted that the optimal number of randomly selected sequences should be defined as a fraction of Neff. Selecting too many sequences, e.g. $50L$ for small alignments (left plot in Figure 3.12), or selecting too few sequences, e.g $1L$ for large alignments (right plot in Figure 3.12), results in a decrease in precision compared to defining the number of sequences as

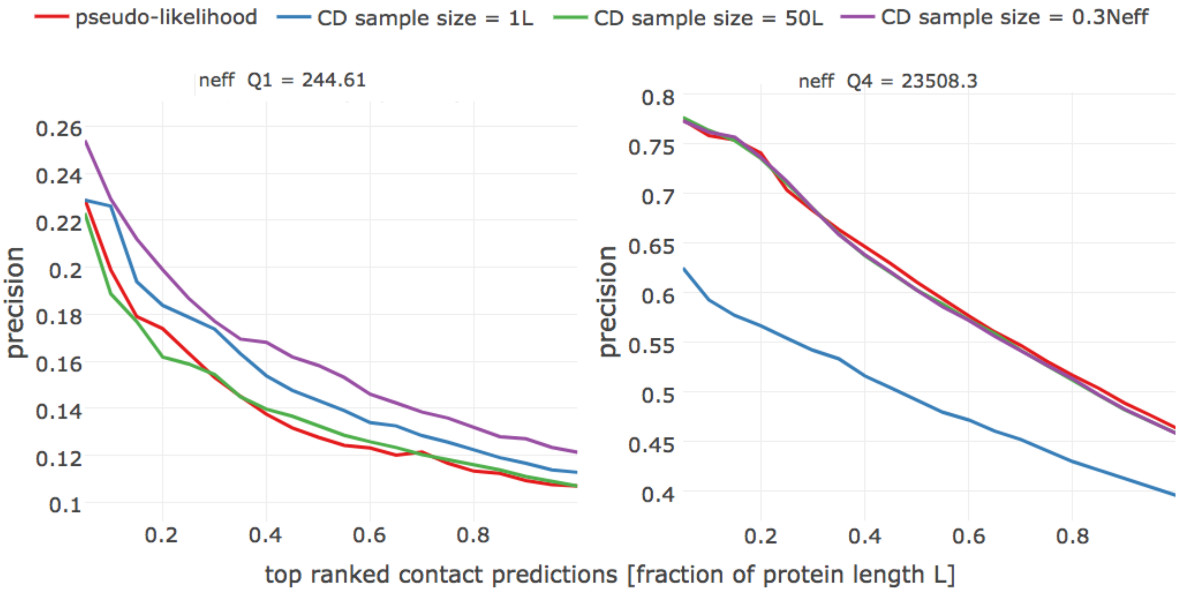


Figure 3.12: Mean precision for top ranked contact predictions over subsets of 75 proteins, defined according to Neff quartiles. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. **pseudo-likelihood**: contact scores computed from pseudo-likelihood. **CD sample size = X** : contact scores computed from CD with SGD. The number of randomly selected sequences for the Gibbs sampling process is specified in the legend. It is defined either as multiples of protein length L or as fraction of the effective number of sequences Neff. **Left** Subset of 75 proteins with Neff $<$ Q1. **Right** Subset of 75 proteins with Q3 $<=$ Neff $<$ Q4.

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 3.13 shows the development of the L2 norm of the gradient for couplings, $||\nabla_{\mathbf{w}} LL(\mathbf{v}^*, \mathbf{w})||_2$, for protein chain 1c75A00 that is of length 71 and has Neff $=$ 16808. The norm of the gradient decreases during optimization and for increasing choices of the sample size it saturates at decreasing levels. For example, increasing the sample size by a factor 100 (from L to $100L$) leads to an approximately 10-fold reduction of the norm of the gradient at convergence (1e+5 compared to 1e+4), 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!

The previous benchmark showed, that precision of the top ranked contacts does not improve to the same amount as the norm of the gradient decreases when the sample size is increased. Probably, the improved gradient when using a larger sample size helps to fine tune the parameters, which only has a negligible effect on the contact score computed as APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. For example, the difference between the parameter norm at convergence for sampling $10L = 710$ sequences or $50L = 3550$ sequences is only marginal (see right plot in Figure 3.13), despite a larger difference of the norm of gradients.

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 3.12. Protein 1ahoA00, that has length 64 and an alignment of 378 sequences (Neff=229), achieves a mean precision of 0.44 over the top $0.1L$ - L contacts when using all $N = 378$ sequences for sampling. When only $0.3N_{\text{eff}} = 69$ sequences are used in

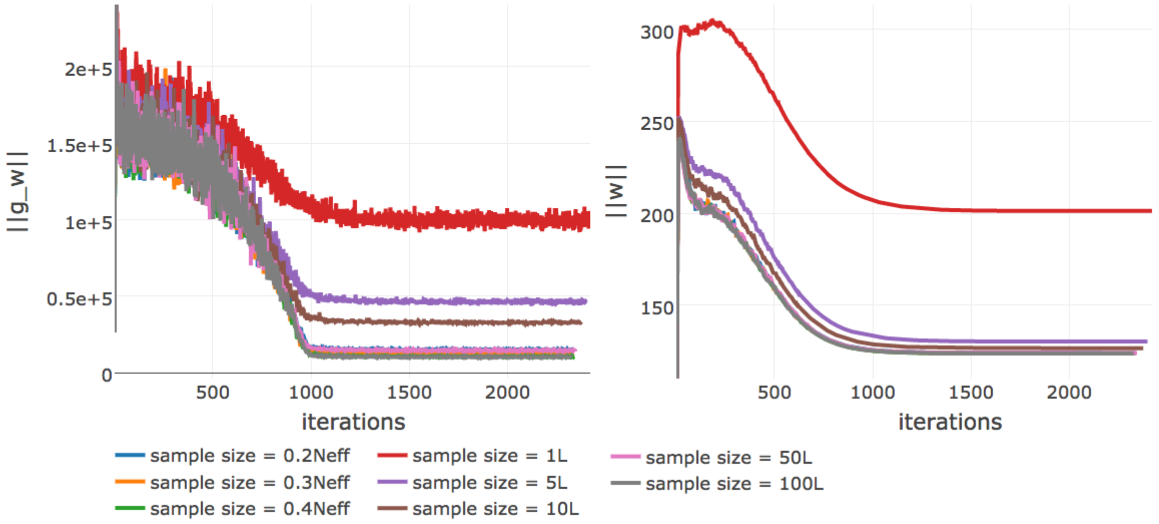


Figure 3.13: Monitoring parameter norm and gradient norm for protein 1c75A00 during SGD using different sample sizes. Protein 1c75A00 has length L=71 and 28078 sequences in the alignment (Neff=16808). The number of sequences, that is used for Gibbs sampling to approximate the gradient, is given in the legend with 1L = 71 sequences, 5L = 355 sequences, 10L = 710 sequences, 50L = 3550 sequences, 100L = 7100 sequences, 0.2Neff = 3362 sequences, 0.3Neff = 5042 sequences, 0.4Neff = 6723 sequences. **Left** L2-norm of the gradients for coupling parameters, $||\nabla_{\mathbf{w}} LL(\mathbf{v}^*, \mathbf{w})||_2$ (without contribution of regularizer). **Right** L2-norm of the coupling parameters $||\mathbf{w}||_2$.

the sampling procedure, 1ahoA00 achieves a mean precision of 0.62. Appendix Figure E.13 shows the course of the norm of the gradient and the norm of coupling parameters during optimization for this protein. Similarly as it has been observed for protein 1c75A00, the norm of the gradient converges towards smaller values when more sequences are used in the Gibbs sampling process and the improved gradient is supposed to lead to a better approximation of the likelihood. One explanation for this obvious discrepancy could be some effect of overfitting. Even though a regularizer is used for optimization and the norm of coupling parameters actually is smaller when using a larger sample size (see the right plot in Appendix Figure E.13).

3.4.2 Varying the number of Gibbs Steps

As discussed earlier, it has been pointed out in the literature that using $k > 1$ Gibbs steps for sampling sequences gives more precise results at the cost of longer run times per gradient evaluation [197,207]. I analysed the impact on performance when the number of Gibbs steps is increased to 5 and 10. As can be seen in Figure 3.14, 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 E.14).

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 3.15). 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 sampling with one Gibbs step, the parameter norm

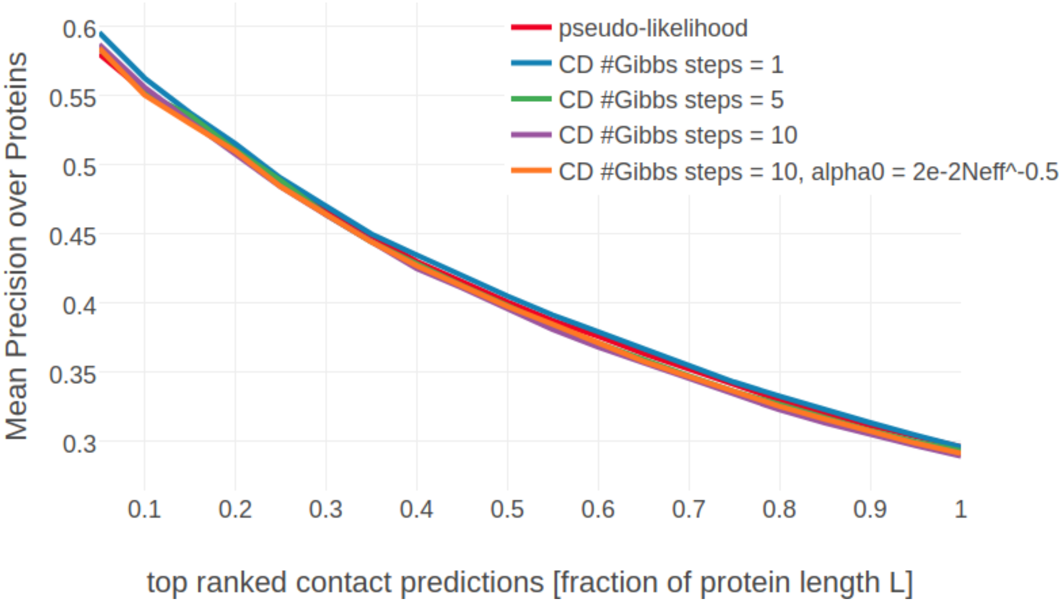


Figure 3.14: Mean precision for top ranked contact predictions over 300 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. **pseudo-likelihood**: contact scores computed from pseudo-likelihood. **CD #Gibbs steps = X**: contact scores computed from CD optimized with SGD and evolving each Markov chain using the number of Gibbs steps specified in the legend.

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 E.14. 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 3.15). 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 E.15). Optimizing CD with 10 Gibbs steps and using a smaller initial learning rate, $\alpha 0 = \frac{2e-2}{\sqrt{N_{\text{eff}}}}$, does not have an overall impact on mean precision as can be seen in Figure 3.14.

3.4.3 Persistent Contrastive Divergence

Finally I analysed, whether evolving the Markov chains over successive iterations, which is known as *persistent contrastive divergence* (PCD), does improve performance [207]. Several empirical studies have shown that PCD performs superior compared to CD-1 and also to CD-10 [207,212]. In the literature is has been pointed out that PCD needs to use small learning rates because in order to sample from a distribution close to the stationary distribution, the parameters cannot change too rapidly. However, using smaller learning rates not only increases run time but also requires tuning of the learning rate and learning rate schedule once again. 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, PCD is switched on. Concretely, PCD is switched on, when the relative change of the norm of coupling parameters, $||\mathbf{w}||_2$, falls below $\epsilon \in \{1e{-}3, 1e{-}5\}$ while the convergence criterion is not altered and convergence is assumed when the relative change

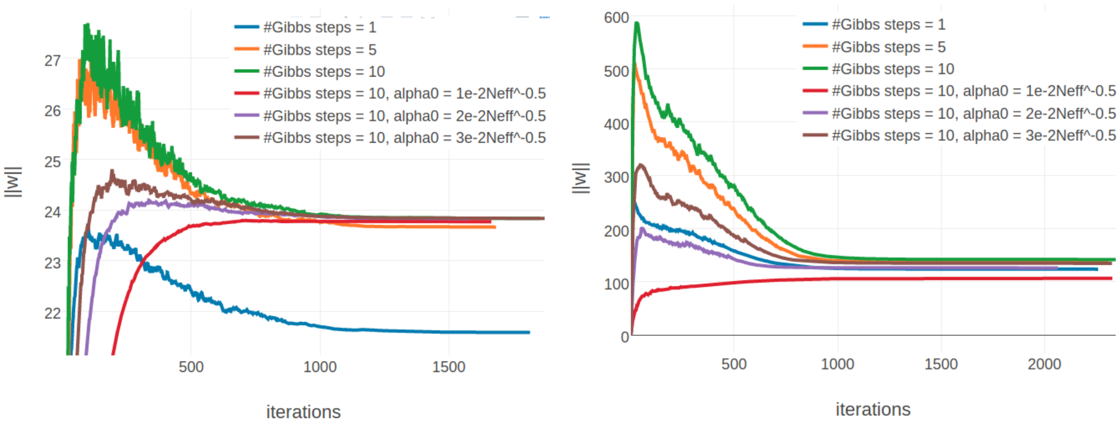


Figure 3.15: Monitoring parameter norm, $||\mathbf{w}||_2$, for protein 1aho_A_00 and 1c75_A_00 during SGD optimization using different number of Gibbs steps and initial learning rates, α_0. Number of Gibbs steps is given in the legend, as well as particular choices for the initial learning rate, when 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).

falls below ϵ=1e−8. As a result, the model will already have approached the optimum when PCD is switched on so that the coupling parameters $\mathbf{w}$ will mot change to quickly over many updates.

Figure 3.16 shows the mean precision of top ranked contacts on the validation set computed with several PCD variants that perform almost equally well. Evolving the Gibbs chains for k=10 steps results in a slight drop in performance, just as it has been observed for CD. Optimizing the full likelihood with CD and switching to PCD at a later stage of optimization does also not have a notable impact on performance.

Again it is insightful to observe the optimization progress for single proteins. For protein 1ahoA00, with low Neff=229, the PCD model converges to the same coupling norm offset ($||\mathbf{w}||_2 \approx 24$) as the CD model using 5 and 10 Gibbs steps (see left plot in Figure 3.17 compared to left plot in 3.15). It can also be seen that when PCD is switched on at a later stage of optimization the coupling norm jumps from the CD-1 level to the PCD level. The different optimum that is found for proteins with small alignments does not seem to affect predictive performance. Interestingly, convergence behaves differently for protein 1c75A00, that has high Neff=16808 (see right plot in Figure 3.17). PCD using one Gibbs step converges to a different coupling norm offset than CD-1 and PCD using ten Gibbs steps. However, when PCD is switched on later during optimization the model either ends up in the CD-1 (switch at $\epsilon = 1e-5$ or $\epsilon = 1e-6$) or in the PCD optimum (switch at $\epsilon = 1e-3$). The cause for this behavior is unclear, yet it has no noticeable impact on overall performance.

Against expectations from the findings in literature, neither CD-k with k>1 Gibbs steps nor PCD does improve performance with respect to precision of the top ranked contact predictions. Swersky and colleagues elaborated on various choices of hyperparameters (e.g momentum, averaging, regularization, etc.) for training Restricted Boltzmann Machines as classifiers with CD-k and PCD [212]. They found many subtleties that need to be explored and can play a crucial role for successful training. In section 3.2.2 I manually tuned the learning rate and annealing schedule for stochastic gradient descent to be used with CD-1. It is plausible, that these settings are not optimal for CD-k with k>1 Gibbs steps and PCD and require tuning once again. Because hyperparameter optimization with stochastic gradient descent is a time-consuming task, in the following, I applied the popular *ADAM* stochastic gradient descent

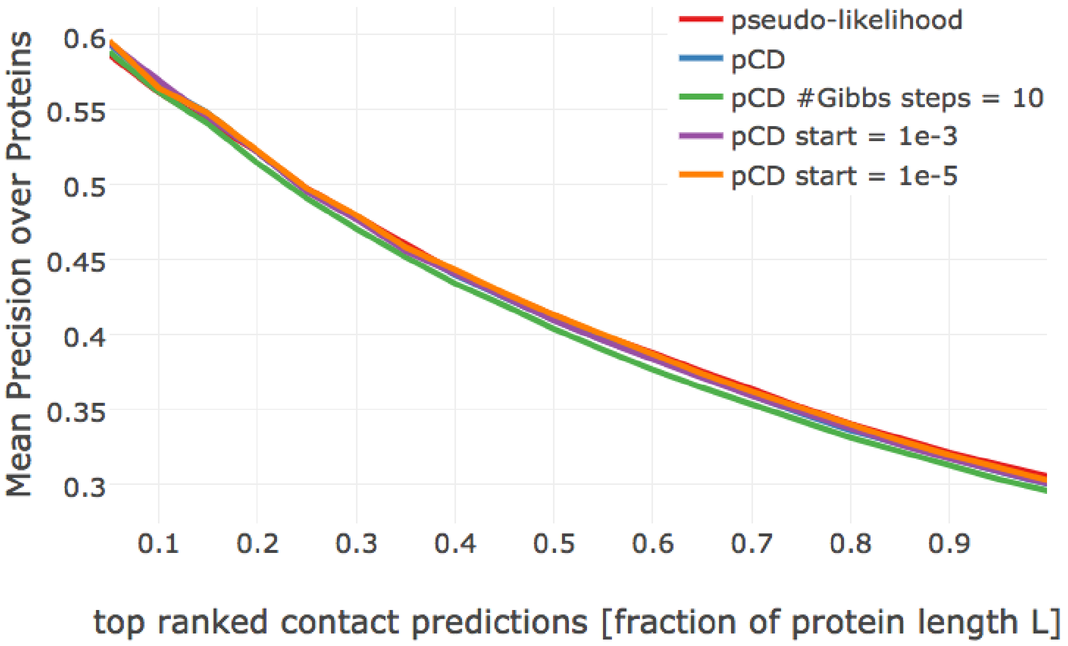


Figure 3.16: Mean precision for top ranked contact predictions over 300 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. **pseudo-likelihood**: contact scores computed from pseudo-likelihood. **pCD**: contact scores computed from PCD optimized with SGD using the hyperparameters that have been found to work optimal with CD as described throughout the last sections. **pCD #Gibbs steps = 10**: same as pCD, but evolving the Gibbs chain for 10 steps. **pCD start = 1e-3**: SGD optimization starts by optimizing the full likelihood using the CD gradient estimate and switches to the PCD gradient estimate once the relative change of L2 norm of parameters has fallen below $\epsilon = 1\text{e}{-3}$ evaluated over the last 10 iterations. **pCD start = 1e-5**: same as 'pCD start = 1e-3', but with $\epsilon = 1\text{e}{-5}$.

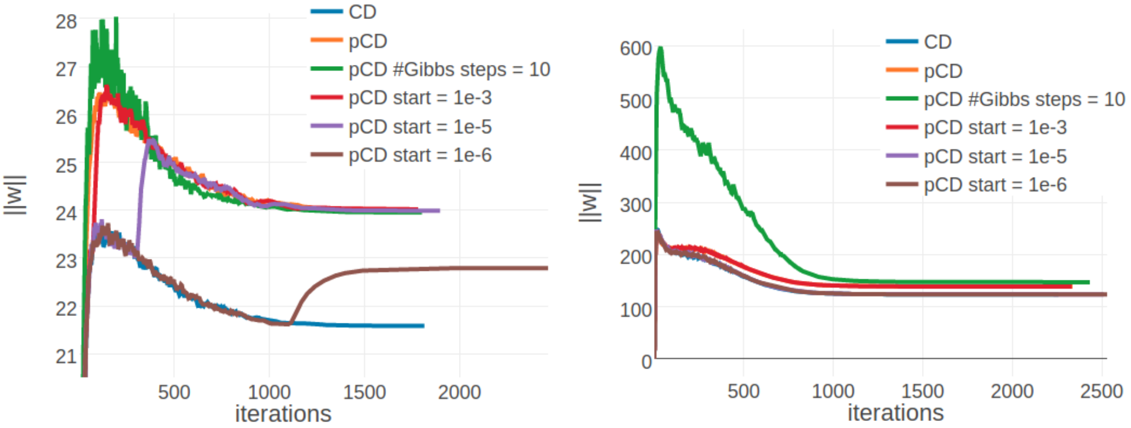


Figure 3.17: Monitoring parameter norm, $||\mathbf{w}||_2$, for protein 1ahoA00 and 1c75A00 during SGD optimization of different objectives. **Left** Protein 1ahoA00 has length L=64 and 378 sequences in the alignment (Neff=229) **Right** Protein 1c75A00 has length L=71 and 28078 sequences in the alignment (Neff=16808). **CD** contrastive divergence using 1 Gibbs step. **pCD** persistent contrastive divergence using 1 Gibbs step. **pCD #Gibbs steps = 10** persistent contrastive divergence using 10 Gibbs steps. **pCD start = 1e-3**, **pCD start = 1e-5**: same as in Figure 3.16 **pCD start = 1e-6**: same as 'pCD start = 1e-3', but with $\epsilon = 1\text{e}{-6}$.

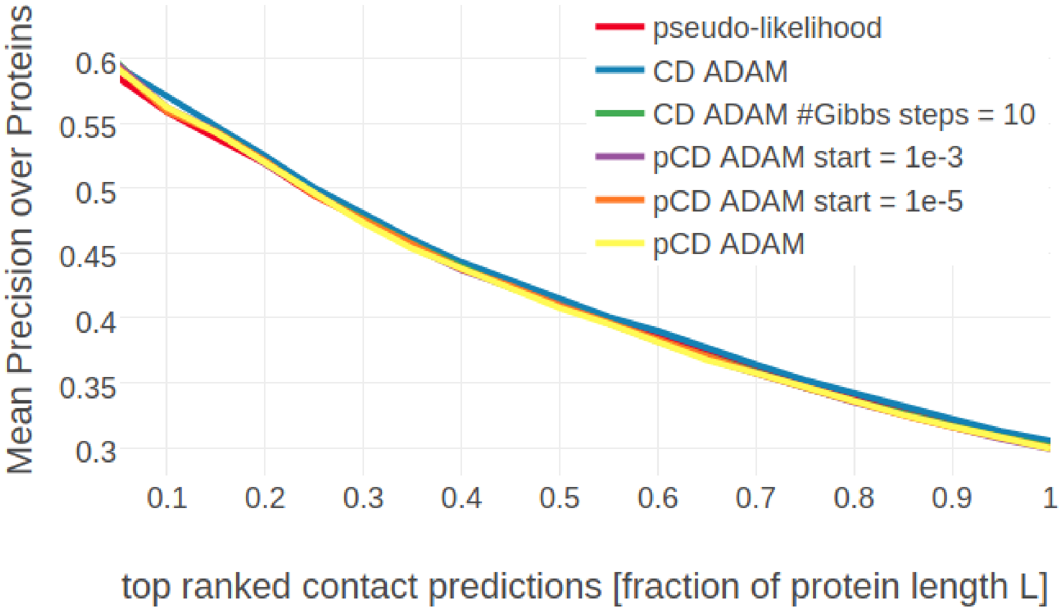


Figure 3.18: Mean precision for top ranked contact predictions over 300 proteins. Contact score is computed as APC corrected Frobenius norm of the couplings. **pseudo-likelihood**: couplings computed from pseudo-likelihood. **CD ADAM**: couplings computed from contrastive divergence using *ADAM* optimizer. **CD ADAM #Gibbs steps = 10**: couplings computed from contrastive divergence using *ADAM* optimizer and 10 Gibbs steps for sampling sequences. **pCD ADAM**: couplings computed from persistent contrastive divergence using *ADAM* optimizer. **pCD ADAM start = 1e-3**: *ADAM* starts by optimizing the full likelihood using the CD gradient estimate and switches to the PCD gradient estimate once the relative change of L2 norm of parameters has fallen below $\epsilon = 1\text{e}{-3}$ evaluated over the last 10 iterations. **pCD ADAM start = 1e-5**: same as "pCD ADAM start = 1e-3" but PCD is switched on for $\epsilon = 1\text{e}{-5}$

optimizer that does in theory not require tuning many hyperparameters [213].

3.5 Using the ADAM Optimizer with Contrastive Divergence

ADAM computes per-parameter adaptive learning rates including momentum. The default values have been found to work well in practice so that little parameter tuning is required (see methods section 3.8.5.1 for details) [198,213]. However, I tested *ADAM* with different learning rates for the optimization with CD-1 for protein 1mkcA00 (number of sequences = 142) and 1c75A00 (number of sequences = 28078) and found that both proteins are sensitive to the choice of learning rate. In contrast to plain stochastic gradient descent, with *ADAM* it is possible to use larger learning rates for proteins having large alignments, because the learning rate will be adapted to the magnitude of the gradient for every parameter individually. For protein 1mkcA00, with Neff=96, a learning rate of 5e-3 quickly leads to convergence whereas for protein 1c75A00, having Neff=16808, an even larger learning rate can be chosen to obtain quick convergence (see Appendix Figure E.16). Therefore, I again specified the learning rate as a function of Neff, $\alpha = 2\text{e}{-3} \log(\text{N}_{\text{eff}})$, such that for small Neff, e.g. 5th percentile of the distribution in the data set ≈ 50, this definition of the learning rate yields $\alpha_0 \approx 8\text{e}{-3}$ and for large Neff, e.g. 95th percentile ≈ 15000, this yields $\alpha_0 \approx 2\text{e}{-2}$.

It is interesting to note, that the norm of the coupling parameters, $||\mathbf{w}||_2$, converges towards different values depending on the choice of the learning rate (see Appendix Figure E.16. 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 [214] and Lasagne [215] packages allow the use of an annealing schedule. I therefore tested *ADAM* with a sigmoidal learning rate annealing schedule which already gave good results for SGD (see section 3.2.2). Indeed, as can be seen in Appendix Figure E.17, when *ADAM* is used with a sigmoidal decay of the learning rate, the L2-norm of the coupling parameters $||\mathbf{w}||_2$ converges roughly towards the same value. For the following analysis I used *ADAM* with a learning rate defined as a function of Neff and a sigmoidal learning rate annealing schedule with decay rate $\gamma = 5e-6$.

I evaluated CD-1, CD-10 and persistent contrastive divergence. As before, I will also evaluate a combination of both, such that PCD is switched on, when the relative change of the norm of coupling parameters, $||\mathbf{w}||_2$, falls below a small threshold. Figure 3.18 shows the benchmark for training the various modified CD models with the *ADAM* optimizer. Overall, the predictive performance for CD and PCD did not improve by using the *ADAM* optimizer instead of the manually tuned stochastic gradient descent optimizer. Therefore it can be concluded that adaptive learning rates and momentum do not provide an essential advantage for inferring *Potts* model parameters with CD and PCD.

The convergence analysis for the two example proteins 1ahoA00 and 1c75A00 reveals, that optimization with *ADAM* converges towards similar offsets as optimization with plain SGD with respect to the L2 norm of coupling parameters. For 1ahoA00, with low Neff=229, the L2 norm of the parameters converges towards $||\mathbf{w}||_2 \approx 21.6$ when using CD-1 and towards $||\mathbf{w}||_2 \approx 24$ when using PCD or CD-k with k>1 (compare left plots in Figures 3.19 and 3.17). For protein 1c75A00, with high Neff=16808, *ADAM* seems to find distinct optima that are clearly separated in contrast to using plain SGD. When using *ADAM* with CD-1 the algorithm converges towards $||\mathbf{w}||_2 \approx 120$, *ADAM* with CD-5 converges towards $||\mathbf{w}||_2 \approx 130$ and with CD-10 towards $||\mathbf{w}||_2 \approx 131$. And using *ADAM* with PCD, regardless of whether the PCD gradient estimate is used right from the start of optimization or only later, the algorithm converges towards $||\mathbf{w}||_2 \approx 134$. Therefore, *ADAM* establishes the clear trend that longer sampling, or sampling with persistent chains results in larger parameter estimates.

3.5.1 A *Potts* model specific convergence criterion

For the *Potts* model there exists a necessary but not sufficient condition that is satisfied at the optimum when the gradient is zero (derived in method section 3.8.4) and that is given by, $\sum_{a,b=1}^{20} w_{ijab} = 0$. This condition is never violated, as long as parameters satisfy this criterion at initialization and the same step size is used to update all parameters. To understand why, note that the 400 partial derivatives $\frac{\partial LL(\mathbf{v}^*, \mathbf{w})}{\partial w_{ijab}}$ for a residue pair (i, j) and for $a, b \in \{1, \ldots, 20\}$ are not independent. 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, $\sum_{a,b=1}^{20} N_{ij} q(x_i = a, q_j = b) = N_{ij}$.

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 3.20). Therefore, it holds $\sum_{a,b=1}^{20} \frac{\partial LL(\mathbf{v}^*, \mathbf{w})}{\partial w_{ijab}} = 0$. This symmetry is translated into parameter updates as long as the same step size is used to update all parameters. However, when using adaptive learning rates, e.g. with the *ADAM* optimizer, this symmetry is broken and the condition $\sum_{a,b=1}^{20} w_{ijab} = 0$ can be violated during the optimization process.

For proteins 1ahoA00 and 1c75A00, Figure 3.21 shows the number of residue pairs for which

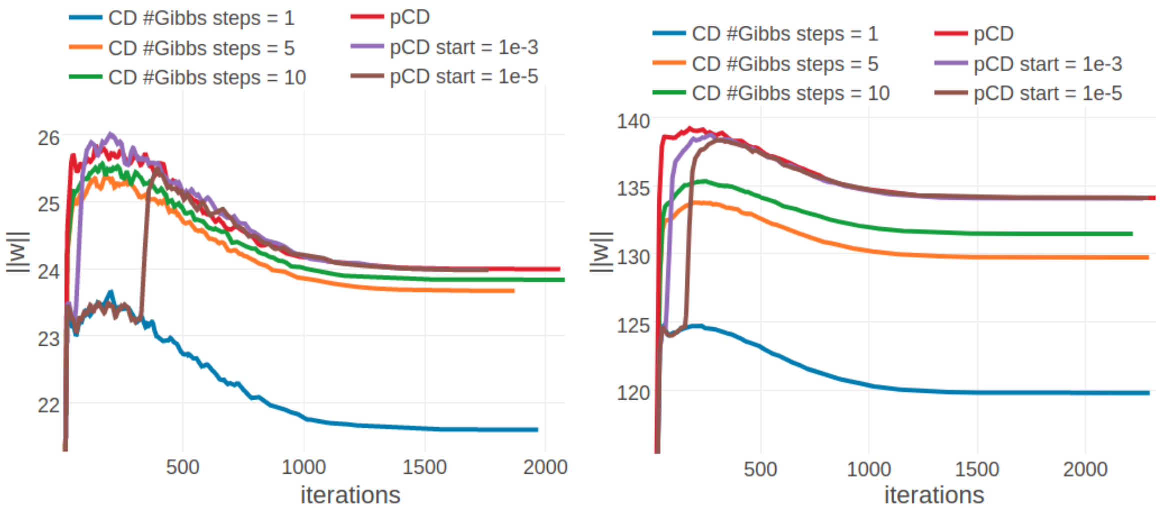


Figure 3.19: Monitoring the L2 norm of coupling parameters,$||\mathbf{w}||_2$, for protein 1ahoA00 and 1c75A00 during optimization of CD and PCD with the *ADAM* optimizer. Contrastive Divergence (CD in legend) is optimized employing a different number of Gibbs steps that are specified in the legend. Persistent contrastive divergence (pCD in legend) uses one Gibbs step. "pCD start= X" indicates that optimization starts by using the CD gradient estimate and switches to the PCD gradient estimate once the relative change of L2 norm of parameters has fallen below a small threshold over the last 10 iterations. The threshold is given in the legend. **Left** Protein 1ahoA00 has length L=64 and 378 sequences in the alignment (Neff=229) **Right** Protein 1c75A00 has length L=71 and 28078 sequences in the alignment (Neff=16808).

this condition is violated according to $|\sum_{a,b=1}^{20} w_{ijab}| > 1\mathrm{e}{-2}$, during optimization with *ADAM*. For about half out of the 2016 residue pairs in protein 1ahoA00 the condition is violated at the end of optimization. For protein 1c75A00 it is about 2300 out of the 2485 residue pairs. Whereas this is not a problem when computing the contact score based on the Frobenius norm of the coupling matrix, it is problematic when utilizing the couplings in the Bayesian framework presented in section 5, which requires the condition $\sum_{a,b=1}^{20} w_{ijab} = 0$ to hold.

3.6 Comparing Contrastive Divergence Couplings with Pseudo Likelihood Couplings

A final benchmark over a larger set of proteins (2000 proteins randomly selected from subsets 5 to 10 described in method section 2.6.1) reveals that contact predictions obtained by maximizing the pseudo-likelihood and by optimizing the full likelihood with contrastive divergence perform similar (see Figure 3.22). At any rate it is interesting to not only compare pseudo-likelihood and contrastive divergence based on overall performance, but to also have a look at single predictions. In the following, I will examine and compare the predictions made by both methods for two representative proteins, one with a small alignment and low corresponding Neff value and one with a large alignment and high corresponding Neff value.

3.6.1 Protein 1c75A00

Protein 1c75A00 has length L=71 and 28078 sequences in the alignment and is among the proteins with the highest number of effective sequences (Neff=16808 > 95th percentile). The

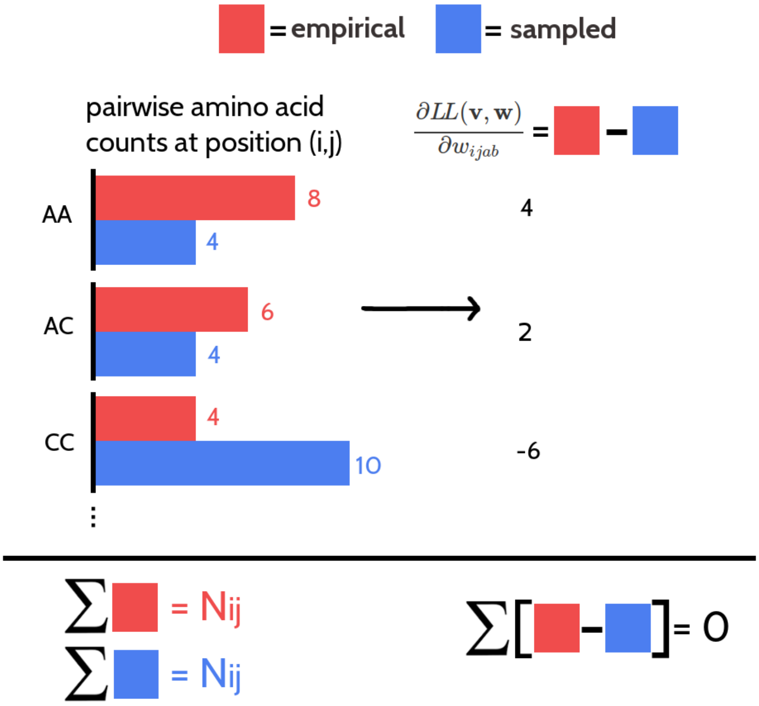

🟥 = empirical
🟦 = sampled
pairwise amino acid counts at position (i,j)
$\frac{\partial LL(\mathbf{v},\mathbf{w})}{\partial w_{ijab}}$ = 🟥 − 🟦
AA
8
4
4
AC
6
4
2
CC
4
10
-6
⋮
∑🟥 = Nij
∑🟦 = Nij
∑[🟥−🟦]= 0

Figure 3.20: 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 empirical alignment. Blue bars represent pairwise amino acid counts at position (i,j) for the sampled 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.

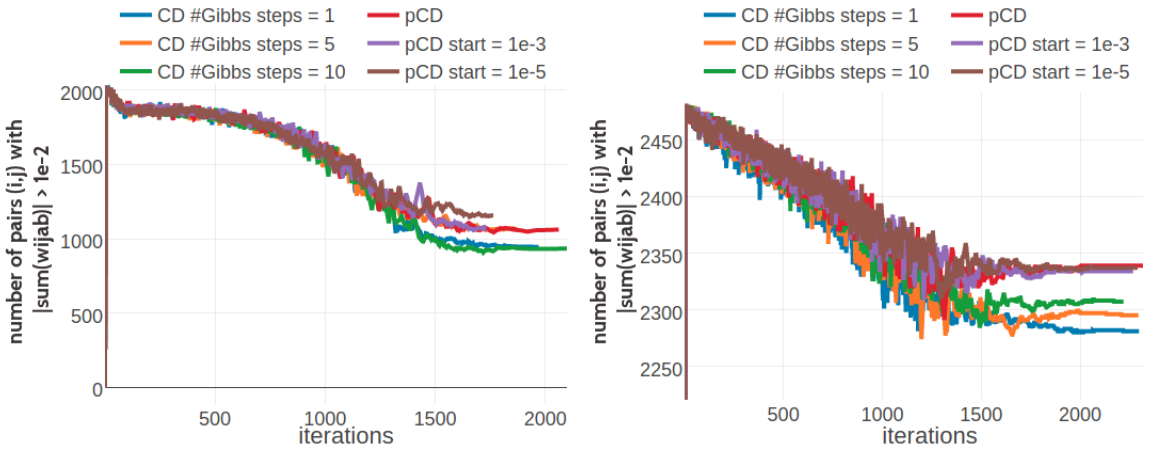


Figure 3.21: Monitoring the number of residue pairs for which $|\sum_{a,b=1}^{20} w_{ijab}| > 1\text{e}{-2}$. Legend is the same as in Figure 3.19. **Left** Protein 1ahoA00 has length L=64 and 378 sequences in the alignment (Neff=229) **Right** Protein 1c75A00 has length L=71 and 28078 sequences in the alignment (Neff=16808).

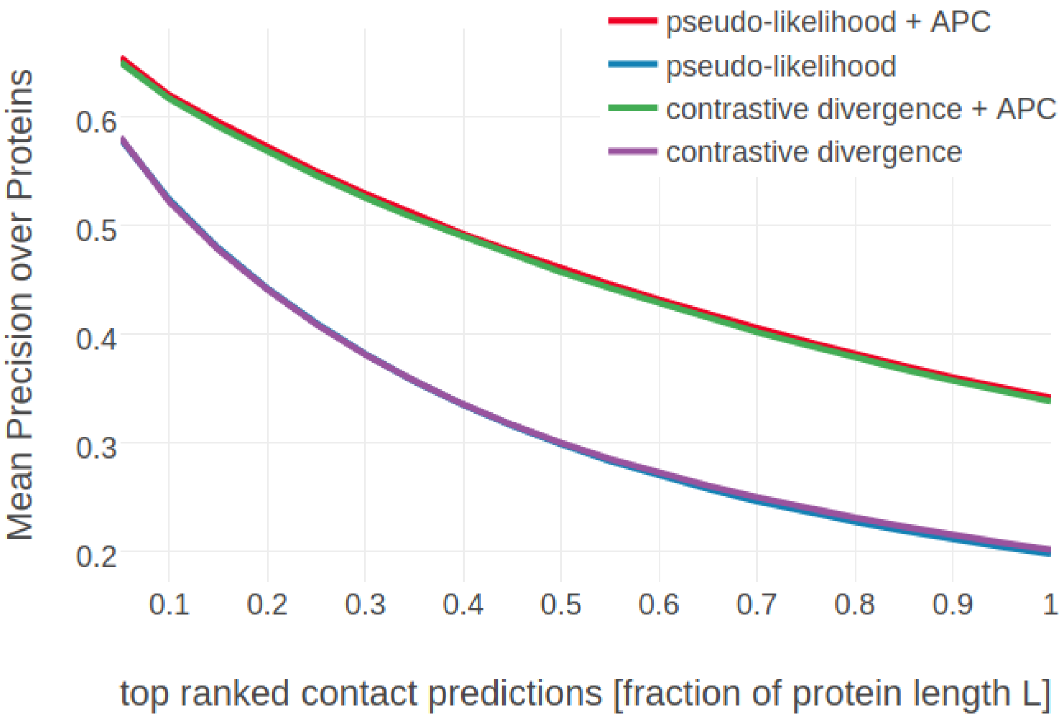


Figure 3.22: Mean precision for top ranked contact predictions over 2000 proteins. **pseudo-likelihood (APC)**: contact score is computed as APC corrected Frobenius norm of the couplings computed from pseudo-likelihood. **pseudo-likelihood**: same as "pseudo-likelihood (APC)" but without APC. **contrastive divergence (APC)**: contact score is computed as APC corrected Frobenius norm of the couplings computed from contrastive-divergence. **contrastive divergence**: same as "contrastive divergence (APC)" but without APC.

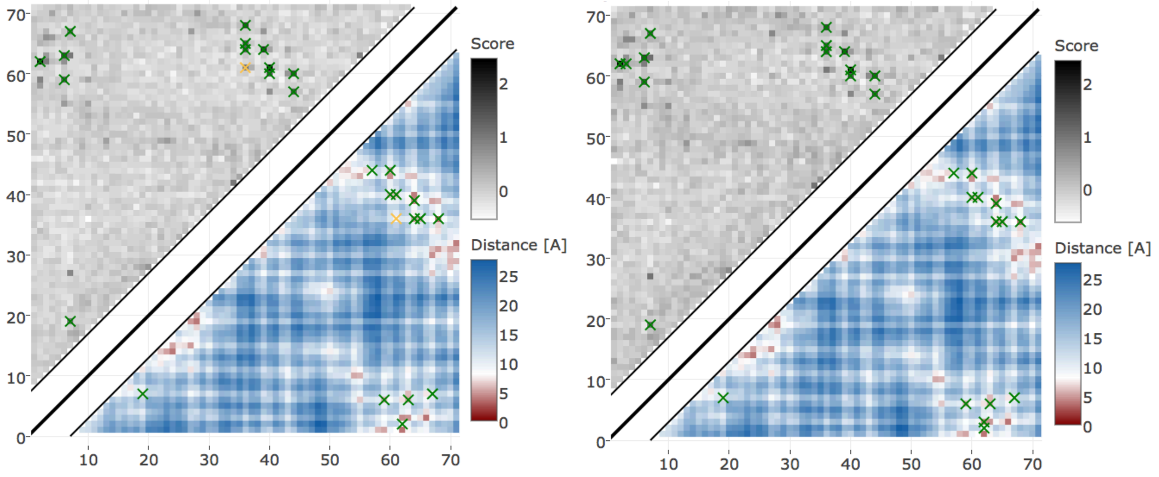


Figure 3.23: Contact maps predicted for protein 1c75A00. Upper left shows predicted contact map and lower right shows the native distance map. Contacts are defined according to a $8\mathring{A}$ C_β distance cutoff and have been computed as APC corrected Frobenius norm of the couplings. **Left** Couplings computed from pseudo-likelihood. **Right** Couplings computed from CD.

contact score (APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$) computed from pseudo-likelihood and contrastive divergence couplings performs equally well (see Appendix Figure E.18). The 14 (=L/5) highest scoring contacts predicted with CD are true positive contacts according to an $8\mathring{A}$ C_β distance cutoff compared to 13 true positive contacts predicted with pseudo-likelihood. Both methods predict very similar contact maps (see Figure 3.23). The highest scoring predictions (top L/5 contacts marked with crosses) are identical except for one contact, which is the false positive contact predicted by the pseudo-likelihood.

The contact maps suggest that both scores are very similar. Indeed, the correlation between both scores is very high (Pearson's correlation coefficient = 0.98) as can bee seen in the right plot in Figure 3.24. Of course, by applying the average product correction (APC), the scores are normalized with respect to the raw contact scores (=Frobenius norm of couplings $\mathbf{w}_{ij}$). The left plot in Figure 3.24 shows the contact scores before applying the average product correction. The raw contact scores computed from contrastive divergence couplings are systematically stronger than for pseudo-likelihood. Most likely this effect arises from the weaker regularization that is used with contrastive divergence ($\lambda_w = 0.1L$) than compared to pseudo-likelihood optimization ($\lambda_w = 0.2L$) (see section 3.3).

However, the contact scores have no meaning by themselves but merely reflect the confidence of the prediction.
It is more meaningful to compare the ranking of the residue pairs imposed by the scores. The left plot in Figure 3.25 compares the ordered scores of both methods that lie very close to the diagonal which indicates that both distribution are very similar (Kolmogorov-Smirnov pvalue = 0.0078, Spearman rho = 0.947536). A detailed view of the top ranked predictions is given in the right plot in Figure 3.25. The three most confident predictions are identical for both methods. Yet, the ranks of subsequent predictions are swapped by only a few positions which was already evident from the contact maps.

3.6.2 Protein 1ss3A00 and 1c55A00

When analysing sample size it was shown that by randomly selecting 0.3Neff sequences for Gibbs sampling improves performance especially for proteins with small Neff (see Figure 3.12)

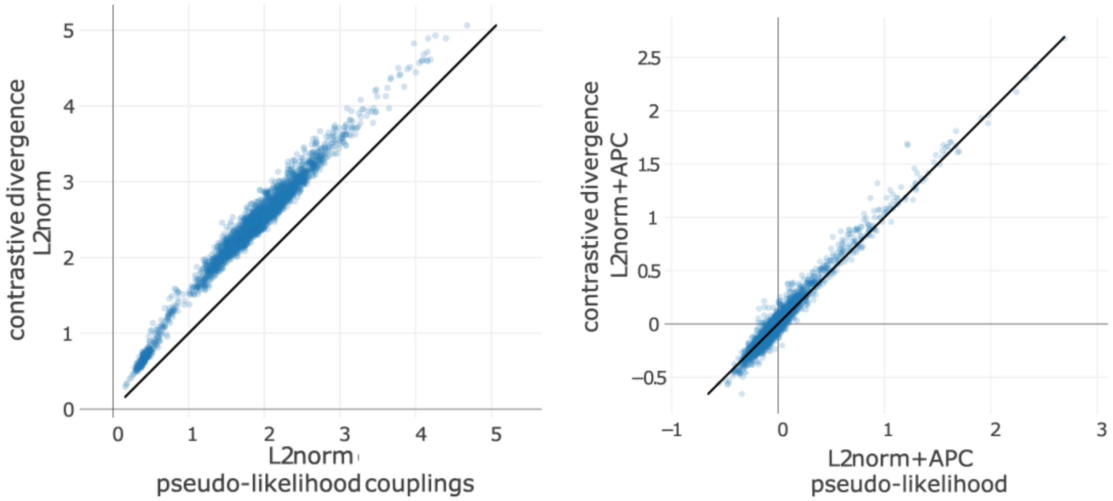


Figure 3.24: Contact scores computed from pseudo-likelihood and CD couplings for protein 1c75A00. **Left** Frobenius norm of couplings. **Right** Frobenius norm + APC of couplings.

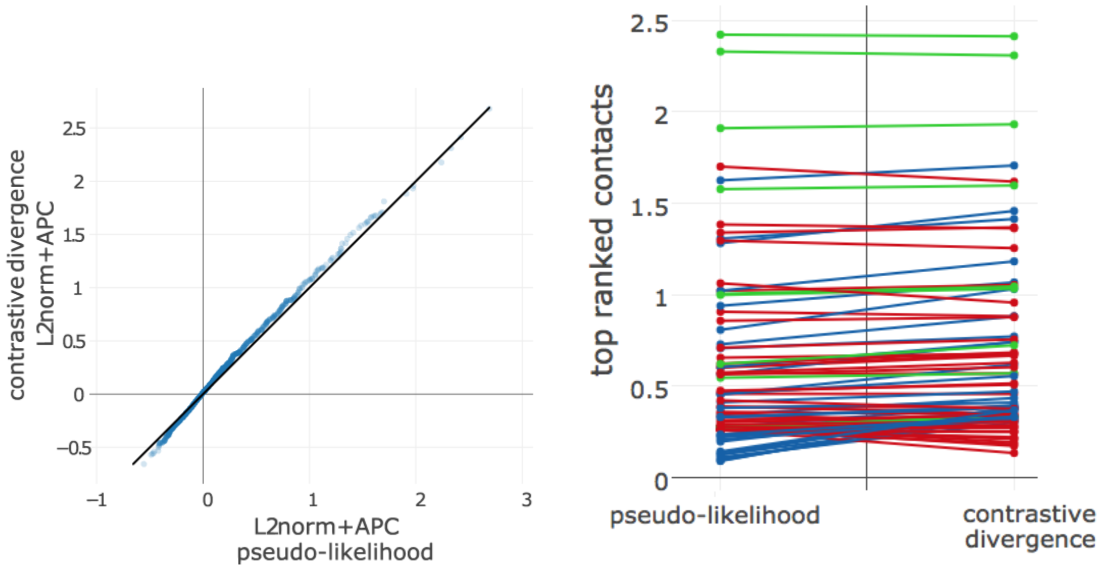


Figure 3.25: Comparing the ranking of highest scoring contacts predicted with pseudo-likelihood and contrastive divergence for protein 1c75A00. Contact scores are computed as APC corrected Frobenius norm of the couplings. **Left** Q-Q plot. **Right** Contact scores for the top 71 (=L) predictions from either method. Identical residue pairs are connected with a line. Green indicates identical ranking of the residue pair for both methods. Blue indicates higher ranking of the residue pair for contrastive divergence. Red indicates higher ranking of the residue pair for pseudo-likelihood.

on a small data set used for benchmarking (75 proteins per Neff quantile bin). This trend is still visible on the larger test data set but to a lesser extent (see Figure E.19).

By looking at some of these proteins with small Neff for which the contact score computed from CD couplings performs better than the score computed from pseudo-likelihood couplings, it is striking that CD mainly predicts strongly conserved positions that have high entropy.

For example, for protein 1ss3A00 (protein length=50, Neff=36), CD makes strong predictions for all pairings of the residues (8, 12, 16, 26, 30, 34) (see Figure 3.26). Five of the predicted contacts are actually true contacts. Taking a look at the structure it is revealed that these positions are disulfide bonds which are strongly conserved. Another example is protein 1c55A00 (protein length=40, Neff=78) for which CD makes strong predictions for pairings between residues (10, 16, 20, 31, 36, 38). Again, it turns out that the five true positive predictions are disulfide bonds (see bottom plot in Figure 3.26).

Interestingly, pseudo-likelihood does not predict the strongly conserved residues pairs and therefore misses some true contacts (see Appendix Figure E.20). However, when recapitulating the analysis from section 3.4.1 by increasing the sample size step-wise, the contact maps predicted with CD start to resemble those predicted by pseudo-likelihood and the predicted contacts between strongly conserved residues vanish (see Appendix Figure E.21). It was unclear from the analysis of the gradients for different samples sizes in section 3.4.1 why sampling less sequences and consequently a worse gradient estimate results in improved performance for proteins with small Neff. Now it can be hypothesized that the improved performance simply originates from the fact that contacts are predicted between strongly conserved columns.

This observation stresses the importance to complement coevolutionary analysis in low data scenarios by the use of other sequence derived information, like conservation. The most successful contact predictors presented in section 1.2.3 integrate features extracted from the MSA because it is known that sequence-based contact prediction is robust when only few sequences are available [87,88].

3.7 Discussion

It is not feasible to evaluate the full likelihood of the *Potts* model for proteins of typical length due to the complexity of the normalization constant. The most popular approach for protein contact prediction to get around this problem is to optimize the pseudo-likelihood instead. However, it is unknown how well the pseudo-likelihood solution approximates the full likelihood solution in case protein families have only few members. In this chapter I tested an alternative approach to infer the *Potts* model parameters, called *contrastive divergence* (CD). It optimizes the full likelihood of the *Potts* model by approximating the gradient with short Gibbs chains. However, a benchmark on a large test set showed that the predictive performance of CD does not improve over pseudo-likelihood with respect to the precision of top ranked contact predictions (see Figure 3.22). CD achieved minor improvements for small protein families, however this improvement could be traced back to amplified signals between strongly conserved residue pairs.

I elaborated in detail on the hyperparameter optimization for the stochastic gradient descent optimizer and the CD model itself. Even though the adaptive learning rate optimizer *ADAM* did not improve performance over plain stochastic gradient descent, it is still likely that appropriate modifications to the optimization procedure, e.g. averaging [206], might be beneficial for particular variants of CD. As discussed in section 3.2.1, the convergence criterion is a crucial aspect for optimization, not only affecting run time but it can also prevent overfitting. It might be worth to assess the convergence properties with more sophisticated convergence

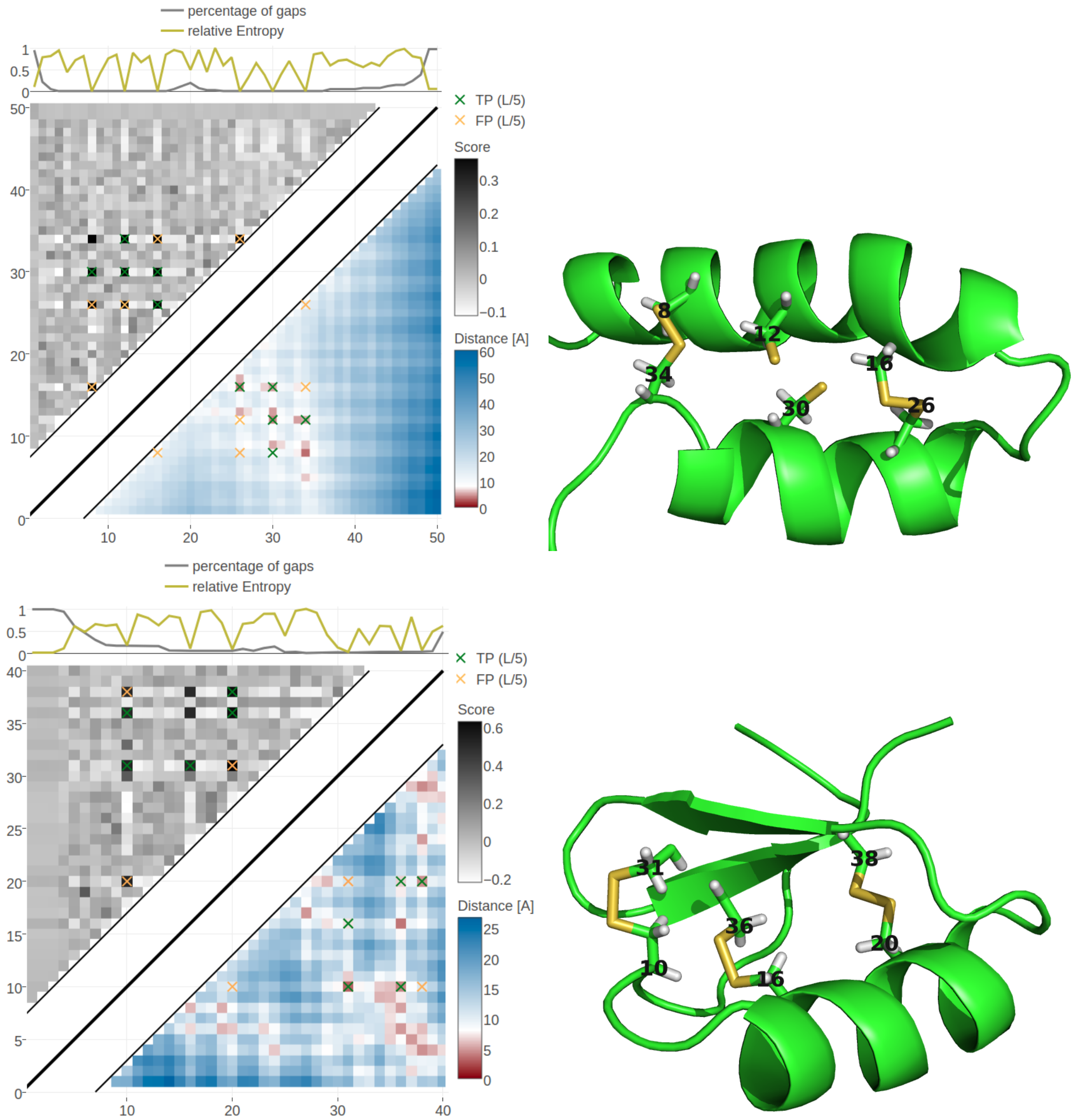


Figure 3.26: Contact maps and structures for protein 1ss3A00 and 1c55A00. Contact scores have been computed as APC corrected Frobenius norm of the CD couplings. Contacts are defined according to a $8\mathring{A}\ C_\beta$ distance cutoff. **Upper left**: predicted contact map and native distance map for protein 1ss3A00 (protein length=50, N=42, Neff=36). **Upper Right**: native protein structure of 1ss3A00 with disulfide bonds between residues pairs (8, 34), (12, 30), (16, 26). **Lower Left** predicted contact map and native distance map for protein 1c55A00 (protein length=40, N=115, Neff=78) **Lower Right** native protein structure of 1c55A00 with disulfide bonds between residues pairs (10, 31), (16, 36), (20, 38).

metrics, like the EB-criterion proposed by Mahsereci et al. [204], instead of using the L2 norm of the coupling parameters, $||\mathbf{w}||_2$.

Against expectations, the best performance with respect to the precision of the top ranked contacts was obtained by using the most simple variant of the *contrastive divergence* algorithm, CD-1. With CD-1, sequence samples are generated according to the current state of the model by evolving Gibbs chains, that have been initialized at data samples, for only one full step. Interestingly, better gradient estimates were obtained by running more Gibbs chains in parallel (see section 3.4.1), but did not carry over to better predictive performance. It is possible that the improved gradient helps to fine tune the parameters. Fine tuning would only have a negligible effect on the contact score, computed as the APC corrected Frobenius norm of the couplings, and the overall ranking of residue pairs.

Cocco and colleagues argued that for the purpose of contact prediction, where predictions only need to capture the topology of the network of coevolving positions, approximate methods such as pseudo-likelihood maximization might be sufficient to provide accurate results [97]. They showed that different approaches for *Potts* model parameter inference yield highly correlated contact scores, using the APC corrected Frobenius norm. In contrast, more quantitative applications, such as inferring mutation landscapes, where energies or probabilities have to be accurate, require precise approaches to fit the model parameters that can reproduce the fine statistics of the empirical data.

Therefore, it can be speculated that the heuristic contact score that has empirically been found to work very well for pseudo-likelihood couplings, might not be an appropriate choice for benchmarking the contrastive divergence approach. Perhaps the CD couplings need to be evaluated in a more sophisticated framework or for other purposes than contact prediction.

3.8 Methods

3.8.1 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 1.3:

$$
\begin{aligned}
p(\mathbf{X}|\mathbf{v},\mathbf{w}) &= \prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{v},\mathbf{w}) \\
&= \prod_{n=1}^{N} \frac{1}{Z(\mathbf{v},\mathbf{w})} \exp\left(\sum_{i=1}^{L} v_i(x_{ni}) \sum_{1 \leq i < j \leq L} w_{ij}(x_{ni}, x_{nj}) \right) \qquad (3.6)
\end{aligned}
$$

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

$$
Z(\mathbf{v},\mathbf{w}) = \sum_{y_1,...,y_L=1}^{20} \exp\left(\sum_{i=1}^{L} v_i(y_i) \sum_{1 \leq i < j \leq L} w_{ij}(y_i, y_j) \right) \qquad (3.7)
$$

The log likelihood is

$$
\begin{aligned}
LL(\mathbf{v},\mathbf{w}) &= \log p(\mathbf{X}|\mathbf{v},\mathbf{w}) \\
&= \sum_{n=1}^{N} \left[\sum_{i=1}^{L} v_i(x_{ni}) \sum_{1 \leq i < j \leq L} w_{ij}(x_{ni}, x_{nj}) \right] - N \log Z(\mathbf{v},\mathbf{w}). \qquad (3.8)
\end{aligned}
$$

The gradient of the log likelihood has single components

$$
\begin{aligned}
\frac{\partial LL(\mathbf{v},\mathbf{w})}{\partial v_{ia}} &= \sum_{n=1}^{N} I(x_{ni}\!=\!a) - N \frac{\partial}{\partial v_{ia}} \log Z(\mathbf{v},\mathbf{w}) \\
&= \sum_{n=1}^{N} I(x_{ni}\!=\!a) - N \sum_{y_1,...,y_L=1}^{20} \frac{\exp\left(\sum_{i=1}^{L} v_i(y_i) + \sum_{1 \leq i < j \leq L} w_{ij}(y_i, y_j)\right)}{Z(\mathbf{v},\mathbf{w})} I(y_i\!=\!a) \\
&= N q(x_i\!=\!a) - N p(x_i\!=\!a|\mathbf{v},\mathbf{w}) \qquad (3.9)
\end{aligned}
$$

and pair components

$$
\begin{aligned}
\frac{\partial LL(\mathbf{v},\mathbf{w})}{\partial w_{ijab}} =& \sum_{n=1}^{N} I(x_{ni} = a, x_{nj} = b) - N \frac{\partial}{\partial w_{ijab}} \log Z(\mathbf{v},\mathbf{w}) \\
=& \sum_{n=1}^{N} I(x_{ni}{=}a, x_{nj}{=}b) \\
& - N \sum_{y_1,\ldots,y_L=1}^{20} \frac{\exp\left(\sum_{i=1}^{L} v_i(y_i) + \sum_{1\le i<j\le L} w_{ij}(y_i,y_j)\right)}{Z(\mathbf{v},\mathbf{w})} I(y_i{=}a, y_j{=}b) \\
=& N q(x_i{=}a, x_j{=}b) - N \sum_{y_1,\ldots,y_L=1}^{20} p(y_1,\ldots,y_L|\mathbf{v},\mathbf{w})\, I(y_i{=}a, y_j{=}b) \\
=& N q(x_i{=}a, x_j{=}b) - N p(x_i{=}a, x_j{=}b|\mathbf{v},\mathbf{w})
\end{aligned} \tag{3.10}
$$

3.8.2 Treating Gaps as Missing Information

Treating gaps explicitly as 0'th letter of the alphabet will lead to couplings between columns that are not in physical contact. To see why, imagine a hypothetical alignment consisting of two sets of sequences as it is illustrated in Figure 3.27. 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 (3.10), 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 normalization of $p(\mathbf{x}_n|\mathbf{v},\mathbf{w})$ should not run over all positions $i \in \{1, ..., L\}$ but only over those i that are not gaps in $\mathbf{x}_n$. Therefore, the set of sequences S_n used for normalization of $p(\mathbf{x}_n|\mathbf{v},\mathbf{w})$ in the partition function will be defined as:

$$
S_n := \{(y_1, ..., y_L) : 0 \le y_i \le 20 \wedge (y_i{=}0 \text{ iff } x_{ni}{=}0)\} \tag{3.11}
$$

and the partition function becomes:

$$
Z_n(\mathbf{v},\mathbf{w}) = \sum_{\mathbf{y}\in S_n} \exp\left(\sum_{i=1}^{L} v_i(y_i) \sum_{1\le i<j\le L} w_{ij}(y_i,y_j)\right) \tag{3.12}
$$

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

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

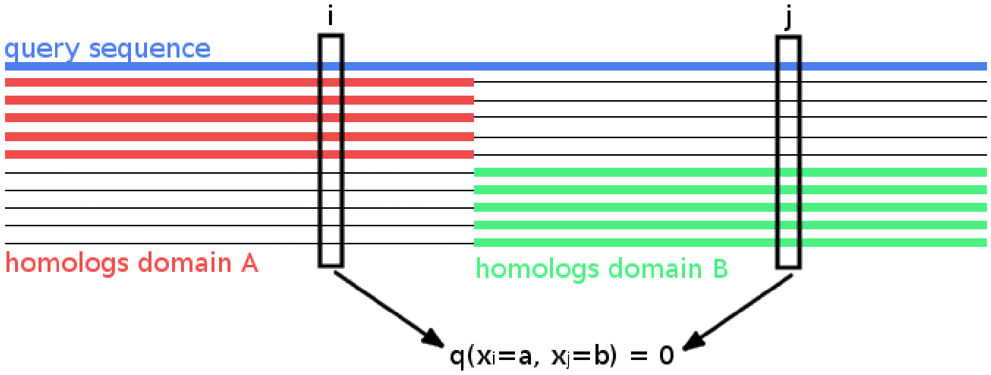


Figure 3.27: Hypothetical MSA consisting of two sets of sequences: the first set has sequences covering only the left half of columns, while the second set has sequences covering only the right half of columns. The two blocks could correspond to protein domains that were aligned to a single query sequence. 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.

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 normalized over $\{1, ..., 20\}$,

$$N_i := \sum_{n=1}^{N} w_n I(x_{ni} \neq 0) \qquad q_{ia} = q(x_i = a) := \frac{1}{N_i} \sum_{n=1}^{N} w_n I(x_{ni} = a) \tag{3.13}$$

$$N_{ij} := \sum_{n=1}^{N} w_n I(x_{ni} \neq 0, x_{nj} \neq 0) \quad q_{ijab} = q(x_i = a, x_j = b) := \frac{1}{N_{ij}} \sum_{n=1}^{N} w_n I(x_{ni} = a, x_{nj} = b) \tag{3.14}$$

with w_n being sequence weights calculated as described in methods section 2.6.3. With this definition, empirical amino acid frequencies are normalized without gaps, so that

$$\sum_{a=1}^{20} q_{ia} = 1 \; , \quad \sum_{a,b=1}^{20} q_{ijab} = 1. \tag{3.15}$$

3.8.3 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 3.8.4. By including the logarithm of this prior into the log likelihood the regularized log likelihood is obtained,

$$LL_{\text{reg}}(\mathbf{v}, \mathbf{w}) = \log \left[p(\mathbf{X}|\mathbf{v}, \mathbf{w}) \, \mathcal{N}(\mathbf{v}|\mathbf{v}^*, \lambda_v^{-1}\mathbf{I}) \, \mathcal{N}(\mathbf{w}|\mathbf{0}, \lambda_w^{-1}\mathbf{I}) \right] \tag{3.16}$$

or explicitly,

$$
\begin{aligned}
LL_{\text{reg}}(\mathbf{v},\mathbf{w}) = & \sum_{n=1}^{N} \left[\sum_{i=1}^{L} v_i(x_{ni}) + \sum_{1 \le i<j \le L} w_{ij}(x_{ni},x_{nj}) - \log Z_n(\mathbf{v},\mathbf{w}) \right] \\
& - \frac{\lambda_v}{2} \sum_{i=1}^{L} \sum_{a=1}^{20} (v_{ia} - v_{ia}^*)^2 - \frac{\lambda_w}{2} \sum_{1 \le i<j \le L} \sum_{a,b=1}^{20} w_{ijab}^2 .
\end{aligned} \tag{3.17}
$$

The gradient of the regularized log likelihood has single components

$$
\begin{aligned}
\frac{\partial LL_{\text{reg}}}{\partial v_{ia}} = & \sum_{n=1}^{N} I(x_{ni} = a) - \sum_{n=1}^{N} \frac{\partial}{\partial v_{ia}} \log Z_n(\mathbf{v},\mathbf{w}) - \lambda_v (v_{ia} - v_{ia}^*) \\
= & \, N_i q(x_i = a) \\
& - \sum_{n=1}^{N} \sum_{\mathbf{y} \in S_n} \frac{\exp\left(\sum_{i=1}^{L} v_i(y_i) + \sum_{1 \le i<j \le L}^{L} w_{ij}(y_i,y_j) \right)}{Z_n(\mathbf{v},\mathbf{w})} I(y_i = a) \\
& - \lambda_v (v_{ia} - v_{ia}^*)
\end{aligned} \tag{3.18}
$$

and pair components

$$
\begin{aligned}
\frac{\partial LL_{\text{reg}}}{\partial w_{ijab}} = & \sum_{n=1}^{N} I(x_{ni} = a, x_{nj} = b) - \sum_{n=1}^{N} \frac{\partial}{\partial w_{ijab}} \log Z_n(\mathbf{v},\mathbf{w}) - \lambda_w w_{ijab} \\
= & \, N_{ij} q(x_i = a, x_j = b) \\
& - \sum_{n=1}^{N} \sum_{\mathbf{y} \in S_n} \frac{\exp\left(\sum_{i=1}^{L} v_i(y_i) + \sum_{1 \le i<j \le L}^{L} w_{ij}(y_i,y_j) \right)}{Z_n(\mathbf{v},\mathbf{w})} I(y_i = a, y_j = b) \\
& - \lambda_w w_{ijab}
\end{aligned} \tag{3.19}
$$

Note that (without regularization $\lambda_v = \lambda_w = 0$) 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 data set in Appendix Figure C.2), the sums over $\mathbf{y} \in S_n$ in eqs. (3.18) and (3.19) can be approximated by $p(x_i = a|\mathbf{v},\mathbf{w}) I(x_{ni} \neq 0)$ and $p(x_i = a, x_j = b|\mathbf{v},\mathbf{w}) I(x_{ni} \neq 0, x_{nj} \neq 0)$, respectively, and the partial derivatives become

$$
\frac{\partial LL_{\text{reg}}}{\partial v_{ia}} = N_i q(x_i = a) - N_i \, p(x_i = a|\mathbf{v},\mathbf{w}) - \lambda_v (v_{ia} - v_{ia}^*) \tag{3.20}
$$

$$
\frac{\partial LL_{\text{reg}}}{\partial w_{ijab}} = N_{ij} q(x_i = a, x_j = b) - N_{ij} \, p(x_i = a, x_j = b|\mathbf{v},\mathbf{w}) - \lambda_w w_{ijab} \tag{3.21}
$$

Note that the couplings between columns i and j in the hypothetical MSA presented in the last section 3.8.2 will now vanish since $N_{ij} = 0$ and the gradient with respect to w_{ijab} is equal to $-\lambda_w w_{ijab}$.

3.8.4 The prior on single potentials

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. (3.21), yields

$$0 = N_{ij}\, q(x_i = a, x_j \neq 0) - N_{ij}\, p(x_i = a|\mathbf{v}, \mathbf{w}) - \lambda_w \sum_{b=1}^{20} w_{ijab}\ , \tag{3.22}$$

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,

$$\sum_{a,b=1}^{20} w_{ijab} = 0. \tag{3.23}$$

At the optimum the gradient with respect to v_{ia} vanishes and according to eq. (3.20), $p(x_i = a|\mathbf{v}, \mathbf{w}) = q(x_i = a) - \lambda_v(v_{ia} - v^*_{ia})/N_i$. This term can be substituted into equation (3.22), yielding

$$0 = N_{ij}\, q(x_i = a, x_j \neq 0) - N_{ij}\, q(x_i = a) + \frac{N_{ij}}{N_i}\lambda_v(v_{ia} - v^*_{ia}) - \lambda_w \sum_{b=1}^{20} w_{ijab}\ . \tag{3.24}$$

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

$$0 = \lambda_v(v_{ia} - v^*_{ia}) - \lambda_w \sum_{b=1}^{20} w_{ijab}. \tag{3.25}$$

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. (3.25), 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 behavior can be corrected by instead choosing a Gaussian prior centered around $\mathbf{v}^*$ obeying

$$\frac{\exp(v^*_{ia})}{\sum_{a'=1}^{20} \exp(v^*_{ia'})} = q(x_i = a). \tag{3.26}$$

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

$$v^*_{ia} = \log q(x_i = a) - \frac{1}{20} \sum_{a'=1}^{20} \log q(x_i = a'). \tag{3.27}$$

For this choice, $v_{ia} - v^*_{ia}$ will be approximately zero and will certainly be much smaller than v_{ia}, hence the sum over coupling coefficients in eq. (3.25) will be close to zero, as it should be.

3.8.5 Stochastic Gradient Descent

The couplings w_{ijab} are initialized at 0 and single potentials v_i will not be optimized but rather kept fixed at their maximum-likelihood estimate v_i^* as described in methods section 3.8.4. The optimization is stopped when a 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 parameterization and is described in detail in methods section 3.8.6. 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 validation set of 300 proteins, that is a subset of the data set described in methods section 2.6.1. Contact scores for couplings are computed as the APC corrected Frobenius norm as explained in section 1.3.6. Pseudo-likelihood couplings are computed with the tool *CCMpredPy* that is introduced in methods section 2.6.2 and the pseudo-likelihood contact score will serve as general reference method for tuning the hyperparameters.

3.8.5.1 The Adaptive Moment Estimation Optimizer *ADAM*

ADAM [213] stores an exponentially decaying average of past gradients and squared gradients,

$$m_t = \beta_1 m_{t1} + (1 - \beta_1) g \tag{3.28}$$

$$v_t = \beta_2 v_{t1} + (1 - \beta_2) g^2 \ , \tag{3.29}$$

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,

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{3.30}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \ . \tag{3.31}$$

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

$$x_{t+1} = x_t - \alpha \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{3.32}$$

Kingma et al. proposed the default values $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e8$ and a constant learning rate $\alpha = 1e - 3$ [213].

3.8.6 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 pairwise amino acid counts between the input alignment and a sampled alignment plus an additional regularization term as given in eq. (3.1). Pairwise amino acid counts are computed from the input alignment accounting for sequence weights (described in methods section 2.6.3) and including pseudo counts (described in methods section 2.6.4). 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 $S = \min(10L, N)$, 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 probabilities given in eq. (3.2). 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 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
```

4 Random Forest Contact Prior

The wealth of successful meta-predictors presented in section 1.2.3 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 simple local statistical methods as presented in section 1.2.1.

For example, predictions of secondary structure elements and solvent accessibility are used by almost all modern machine learning predictors, such as MetaPsicov [85], NeBCon [88], EPSILON-CP [87], PconsC3 [83]. Other frequently used features include pairwise contact potentials, sequence separation and conservation measures such as column entropy [85,88,216].

In the following sections I present a random forest classifier that uses sequence derived features to distinguish contacts from non-contacts. Method section 4.6.1 lists all features used to train the classifier including the aforementioned standard features as well as some novel features. The probabilistic predictions of the random forest model can be introduced directly as prior information into the Bayesian statistical model that will be presented in chapter 5 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.

4.1 Random Forest Classifiers

Random Forests are supervised machine learning methods that belong to the class of ensemble methods [217–219]. They are easy to implement, fast to train and can handle large numbers of features due to implicit feature selection [220]. 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)
2. every split of a node is evaluated on a random subset of features

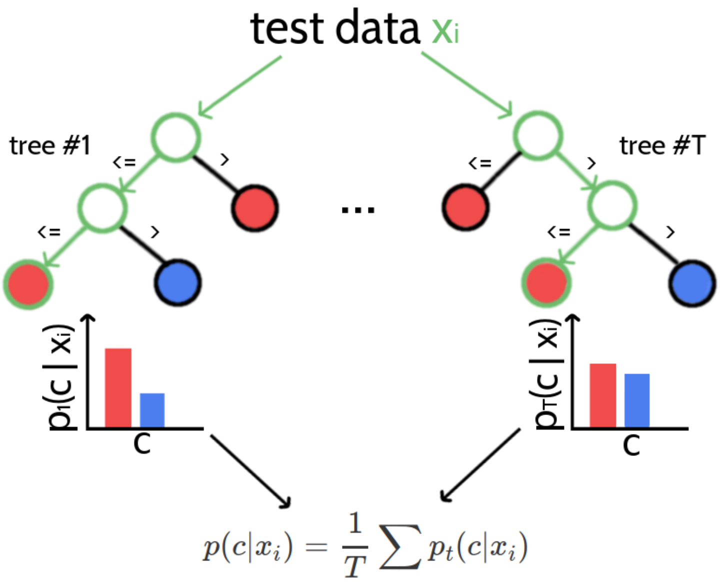


Figure 4.1: Classifying new data with random forests. A new data sample is run down every tree in the forest until it ends up in a leaf node. Every leaf node has associated class probabilities $p(c)$ reflecting the fraction of training samples at this leaf node belonging to every class c. The color of the leaf nodes reflects the class with highest probability. The predictions from all trees in form of the class probabilities are averaged and yield the final prediction.

A single decision tree, especially when it is grown very deep is highly susceptible to noise in the training set and therefore prone to overfitting which results in poor generalization ability. As a consequence of randomness and averaging over many decision trees, the variance of a random forest predictor decreases and therefore the risk of overfitting [221]. 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 4.1 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 [220]. 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 [222].

4.2 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 run time. While more trees in the random forest generally improve performance of the model, they will slow down training and prediction. A crucial hyperparameter is the number of features that is randomly selected for a split at each node in a tree [223]. 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. Over-fitting 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. [221]

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 4.6.3. 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 4.2 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 4.6.1.

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 4.3. I chose `max_features=0.30` and

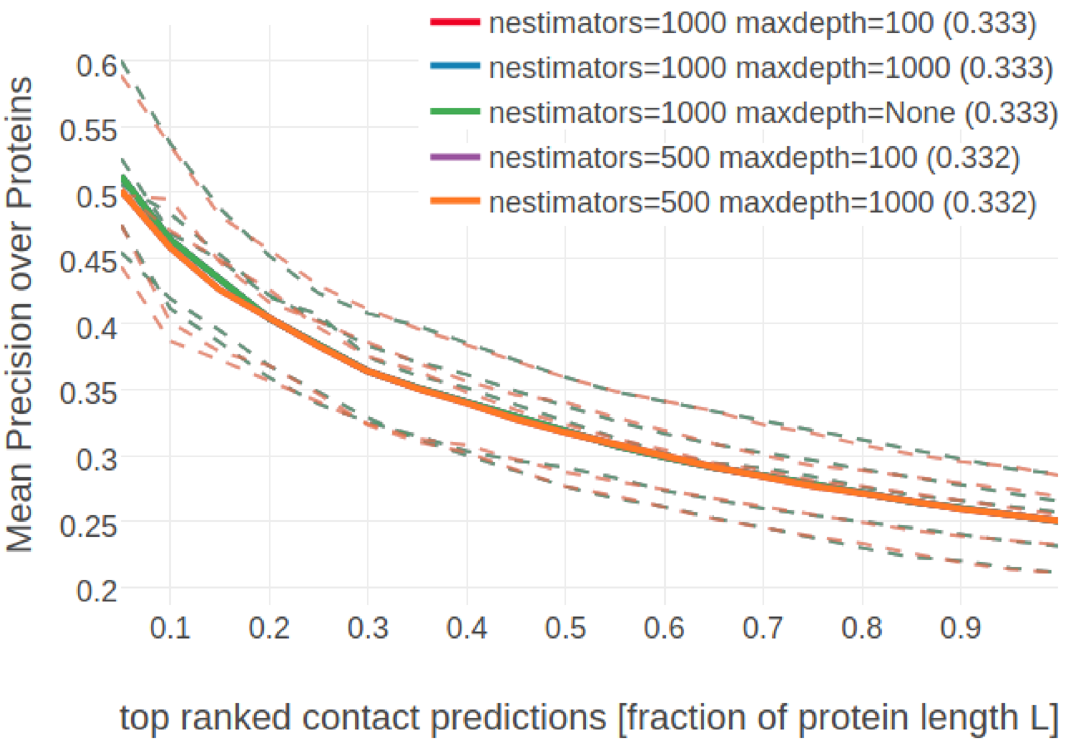


Figure 4.2: Mean precision over 200 proteins against highest scoring contact predictions from random forest models for different settings of *n_ estimators* and *max_ depth*. Dashed lines show the performance of models that have been learned on the five different subsets of training data. Solid lines give the mean precision over the five models. Only those models are shown that yielded the five highest mean precision values (given in parentheses 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.

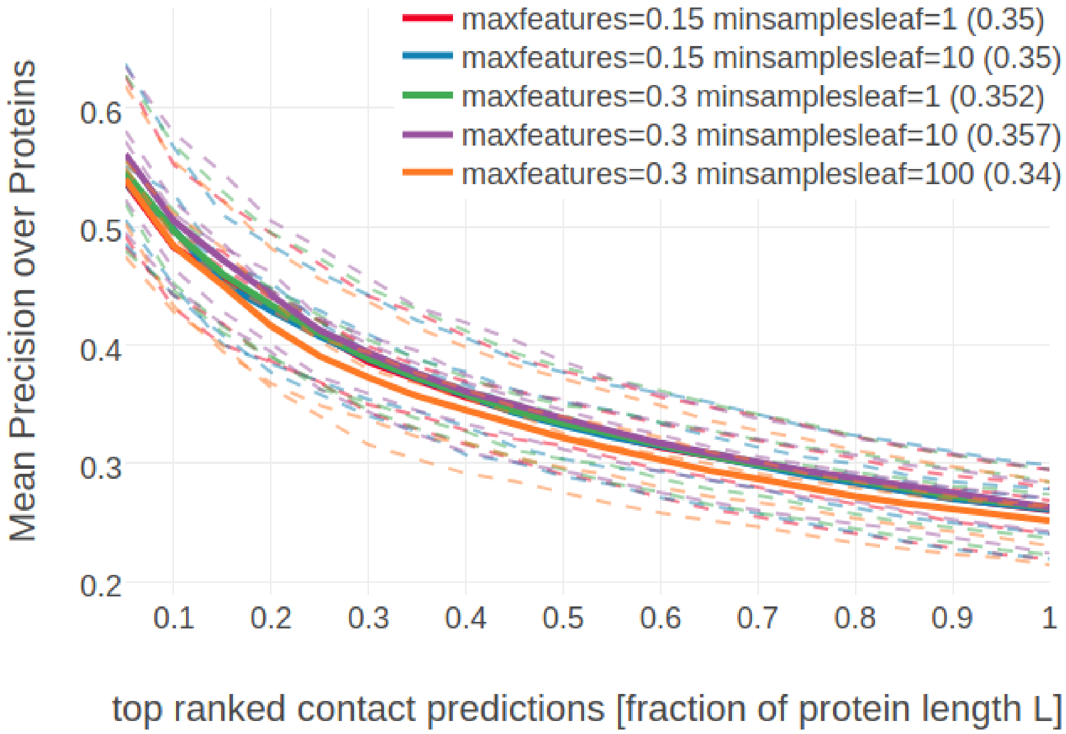


Figure 4.3: Mean precision over 200 proteins against highest scoring contact predictions from random forest models with different settings of *min_samples_leaf* and *max_features*. Dashed lines show the performance of models that have been learned on the five different subsets of training data. Solid lines give the mean precision over the five models. Only those models are shown that yielded the five best mean precision values (given in parentheses in the legend).

`min_samples_leaf=10` for further analysis. Tuning the hyperparameters in a different order or on a larger data set gives similar results.

In a next step I assessed data set 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 data set 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 Figures F.6 and F.7, the default choice of using a window size of five positions and the non-contact threshold of $8\AA$ 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 Figure F.8. These estimates might be biased in a way since the random forest hyperparameters have been optimized on a data set using exactly these optimal settings.

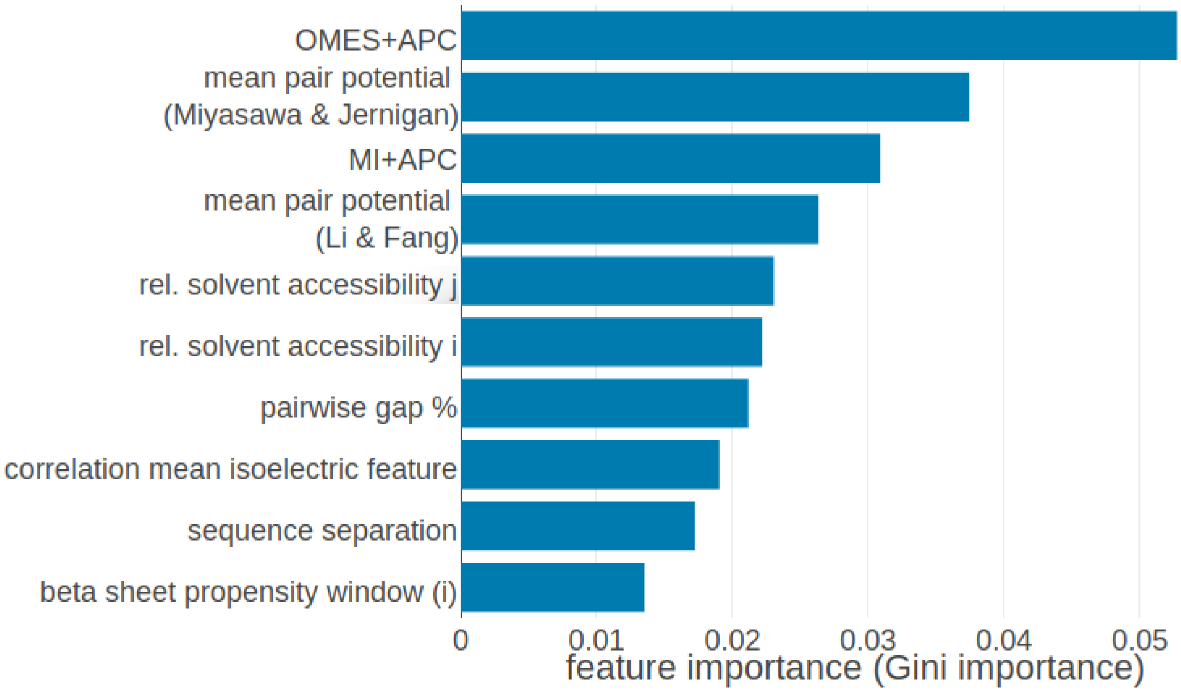


Figure 4.4: Top ten features ranked according to *Gini importance*. **OMES+APC**: APC corrected OMES score according to Fodor&Aldrich [224]. **mean pair potential (Miyasawa & Jernigan)**: average quasi-chemical energy of transfer of amino acids from water to the protein environment [225]. **MI+APC**: APC corrected mutual information between amino acid counts (using pseudo-counts). **mean pair potential (Li&Fang)**: average general contact potential by Li & Fang [70]. **rel. solvent accessibilty i(j)**: RSA score computed with Netsurfp (v1.0) [226] 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 [227] computed within a window of five positions around i. Features are described in detail in methods section 4.6.1.

4.3 Evaluating Random Forest Model as Contact Predictor

I trained a random forest classifier on the feature set described in methods section 4.6.1 and using the optimal hyperparameters identified with 5-fold cross-validation as described in the last section.

Figure 4.4 shows the ranking of the ten most important features according to *Gini importance*. Both local statistical contact scores, *OMES* [224] and MI (mutual information between amino acid counts), constitute the most important features besides the mean pair potentials cording to Miyazawa & Jernigan [225] and Li&Fang[70]. 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 data set. Removing irrelevant features from the data set is a convenient procedure to reduce model complexity. It has been found, that prediction performance might even increase after removing the most irrelevant features [220]. For example, during the development of *EPSILON-CP*, a deep neural network

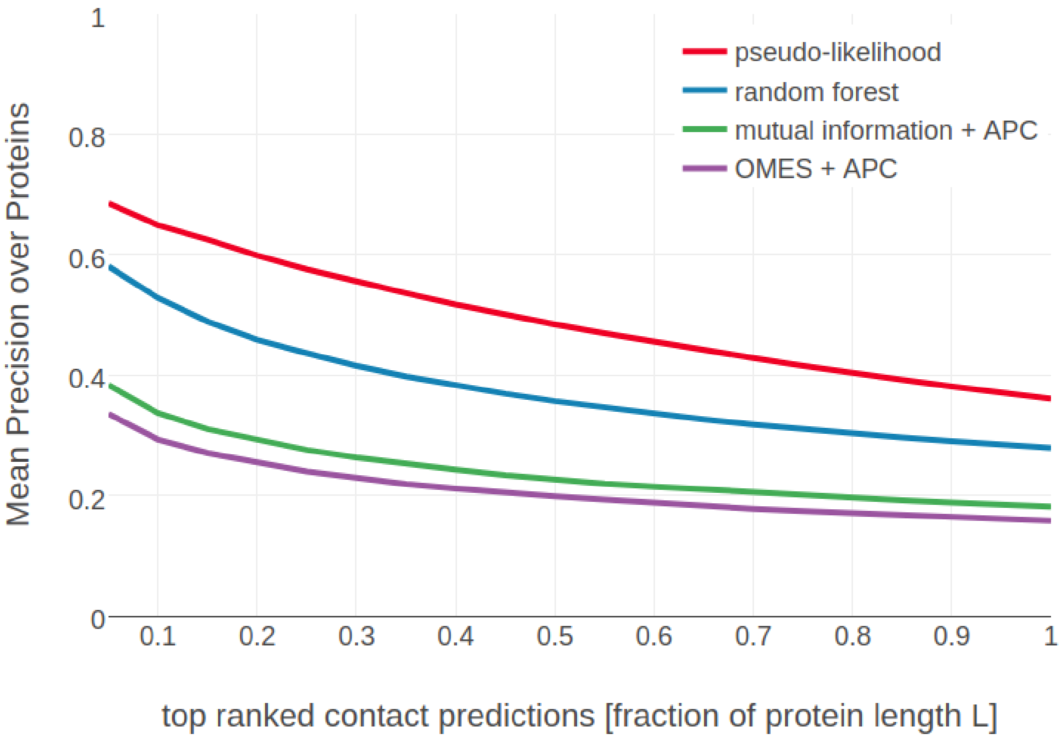


Figure 4.5: Mean precision for top ranked contacts on a test set of 1000 proteins. **pseudo-likelihood** = APC corrected Frobenius norm of couplings computed with pseudo-likelihood. **random forest** = random forest model trained on 75 sequence derived features. **OMES** = APC corrected *OMES* contact score according to Fodor&Aldrich [224]. **mutual information** = APC corrected mutual information between amino acid counts (using pseudo-counts).

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 acid composition), the performance of their predictor increased slightly [87]. Other studies have also emphasized the importance of feature selection to improve performance and reduce model complexity [68,70].

As described in methods section 4.6.4, 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 (see appendix Figure F.9). 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 4.5 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 4.4.

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 Appendix Figure F.2).
Both, local statistical models *OMES* and MI also perform weak on small alignments, leading

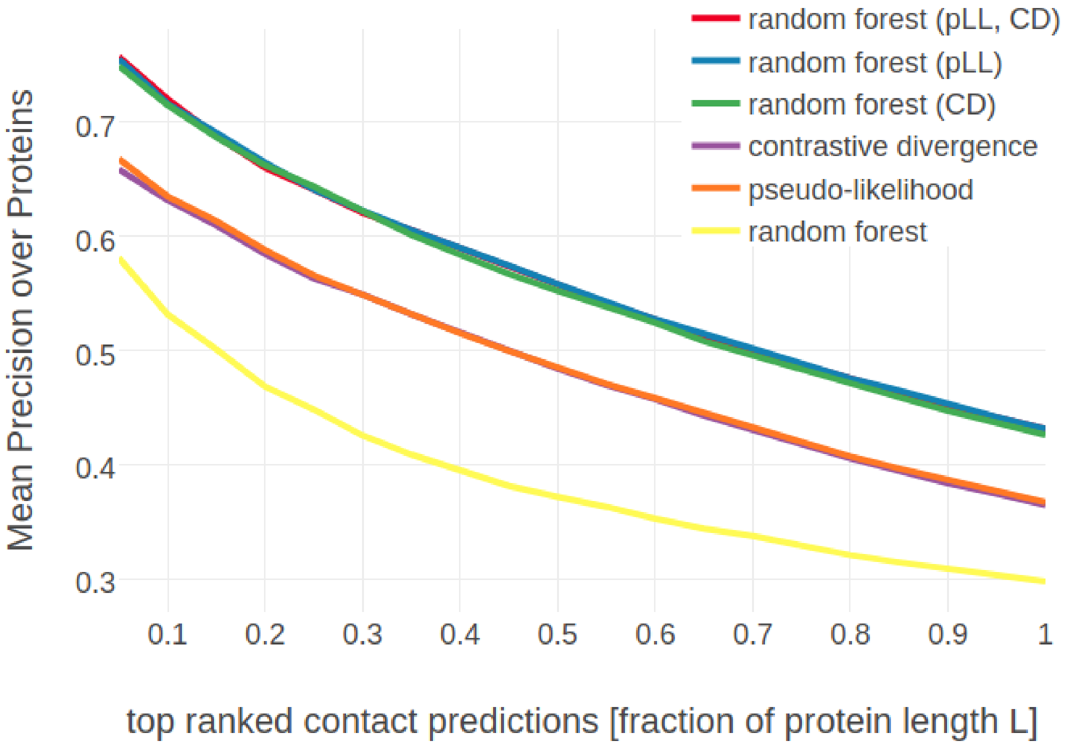


Figure 4.6: Mean precision for top ranked contacts on a test set of 1000 proteins. **random forest (pLL, CD)** random forest model trained on sequence features and the pseudo-likelihood and contrastive divergence contact scores. **random forest (pLL)** random forest model trained on sequence features and the pseudo-likelihood contact score. **random forest (CD)** random forest model trained on sequence features and the contrastive divergence contact score. **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.

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 [83,87].

4.4 Using Contact Scores as Additional Features

Figure F.2 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 F.3) followed by the same sequence features that were found in the previous analysis in Figure 4.4. The model trained on the 76 most relevant features performs as well as the model trained on the full feature set and was used in the benchmark shown in Figure 4.6. 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 (about 12%) as can be seen in in the left plot in Figure 4.7. In

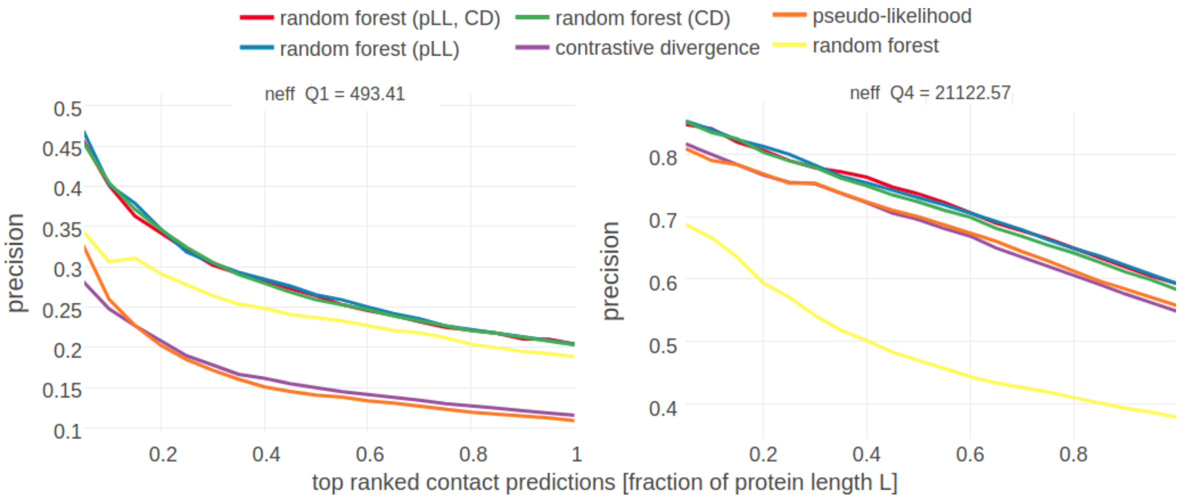


Figure 4.7: Mean precision for top ranked contacts on a test set of 1000 proteins splitted into four equally sized subsets with respect to Neff. Subsets are defined according to quantiles of Neff values. **Left**: Subset of proteins with Neff < Q1. **Right**: Subset of proteins with Q3 <= Neff < Q4. Methods are the same as in Figure 4.6

contrast, the improvement on large alignments (right plot in Figure 4.7) is smaller (about 5%), as the gain from simple sequence features compared to the much more powerful coevolution signals is neglectable.

Similarly, the contact scores derived from couplings computed with CD in chapter 3 can be added as a feature instead of the pseudo-likelihood score or besides the pseudo-likelihood contact score. Again, the contrastive-divergence and the pseudo-likelihood contact score comprise the most important features in the respective models (see Appendix Figures F.4 and F.5). The three models trained on additional coevolution features perform comparably (see Figure 4.6) and apparently, there is minor information gain by adding both coevolution contact scores. Since it has been shown in section 3.6 that pseudo-likelihood and contrastive divergence contact scores are highly correlated, resulting in very similar rankings for residue pairs, it is not surprising that the random forest model including both coevolution scores does not improve over the random forest model including only one of both scores.

4.5 Discussion

Much information about interacting protein residues is typically contained in simple protein sequence features. All popular machine learning and meta-predictors for contact prediction employ sequence derived features as additional source of information besides coevolution scores. In line with this knowledge I developed a random forest classifier for contact prediction that is trained on simple sequence features.

Random forests are a convenient choice for many machine learning applications as they require no input preparation, such as feature scaling, they perform implicit feature selection and provide a robust indicator of feature importance and can handle huge feature space. Furthermore they are quick and straight forward to train and have been shown to perform well for protein contact prediction.

As expected, the random forest model yielded a robust estimator that outperformed coevolution methods for small protein families where they suffer from the low signal-to-noise ratio. Furthermore, I integrated the predictions of the pseudo-likelihood and the constrastive divergence method as additional features for training. Again as expected, the individual methods greatly contribute and improve the predictive performance of the random forest classifier. Even for protein families with many sequences, where coevolutionary methods perform best, the combined random forest model improves over the individual coevolution approaches. Yet, including both coevolution scores as additional features into the random forest model does not help to boost performance further. Apparently, they do not seem to represent complementary information which was on the other hand already expected from the analysis in chapter 3.

4.6 Methods

4.6.1 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 2.6.3.

4.6.1.1 Global Features

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

Table 4.1: Features characterizing the total alignment

Feature	Description	Features per residue pair
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 2.6.3)	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)[227] given as average three state propensities	3
NetsurfP	secondary structure prediction by Netsurfp (v1.0)[226] 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.

4.6.1.2 Single Position Features

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

Table 4.2: Single Position Sequence Features

Feature	Description	Features per residue pair
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
kullback leibler divergence	between weighted observed and background amino acid frequencies [228]	2
jennson shannon divergence	between weighted observed and background amino acid frequencies [228]	2
PSSM	log odds ratio of weighted observed and background amino acid frequencies [228]	40
secondary structure prediction	three state propensities PSIPRED (v4.0) [227]	6
secondary structure prediction	three state propensities Netsurfp (v1.0) [226]	6
solvent accessibility prediction	RSA and RSA Z-score Netsurfp (v1.0) [226]	4
relative position in sequence	$\frac{i}{L}$ for a protein 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 [229]	10
Average polarity (Grantham)	Polarity according to Grantham [230]. Data taken from AAindex Database [231].	2
Average polarity (Zimmermann)	Polarity according to Zimmermann et al. [232]. Data taken from AAindex Database [231].	2
Average isoelectricity	Isoelectric point according to Zimmermann et al. [232]. Data taken from AAindex Database [231].	2
Average hydrophobicity (Wimley&White)	Hydrophobicity scale according to Wimley & White [233]. Data taken from UCSF Chimera [233].	2
Average hydrophobicity (Kyte&Dolittle)	Hydrophobicity index according to Kyte & Doolittle [234]. Data taken from AAindex Database [231].	2
Average hydrophobicity (Cornette)	Hydrophobicity according to Cornette [235].	2
Average bulkiness	Bulkiness according to Zimmerman et al. [232]. Data taken from AAindex Database [231].	2
Average volume	Average volumes of residues according to Pontius et al. [236]. Data taken from AAindex Database [231].	2

There are 48 single sequence 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 feature for center residue i will be computed as the mean feature over residues $[i–2, \ldots, i, \ldots, i+2]$. Whenever the window extends the range of the sequence (for $i < 2$ and $i > (L-2)$), the window feature will be computed only for valid sequence positions. This results in additional 96 window features per residue pair.

4.6.1.3 Pairwise Features

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

Table 4.3: Pairwise Sequence Features

Feature	Description	Features per residue pair
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 4.2	13
pairwise potential (buried)	Average quasi-chemical energy of interactions in an average buried environment according to Miyazawa&Jernigan [225]. Data taken from AAindex Database [231].	1
pairwise potential (water)	Average quasi-chemical energy of transfer of amino acids from water to the protein environment according to Miyazawa&Jernigan [225]. Data taken from AAindex Database [231].	1
pairwise potential (Li&Fang)	Average general contact potential by Li&Fang [70]	1
pairwise potential (Zhu&Braun)	Average statistical potential from residue pairs in beta-sheets by Zhu&Braun [237]	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 coevolution score	according to Fodor&Aldrich [224] with and without APC	2

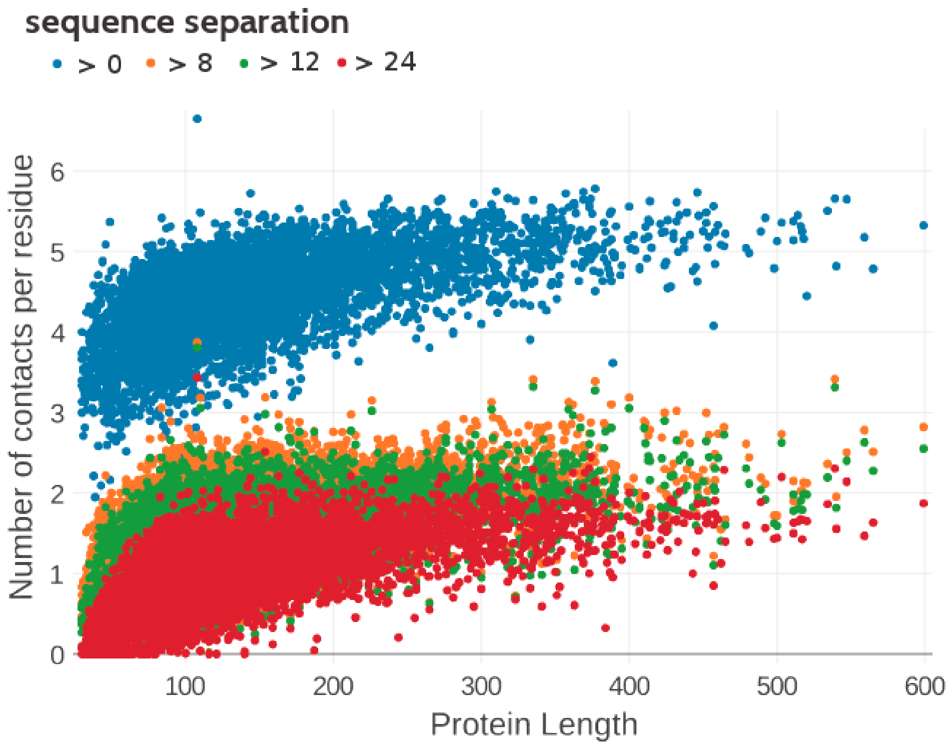


Figure 4.8: Observed number of contacts per residue has a non-linear relationship with protein length. Distribution is shown for several thresholds of sequence separation |j-i|.

There are in total 26 pairwise sequence features.

4.6.2 Simple Contact Prior with Respect to Protein Length

The last feature listed in table 4.1 ("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 contacts 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 4.8.

In log space, the average number of contacts per residue can be fitted with a linear regression and yields the following functions:

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

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

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

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

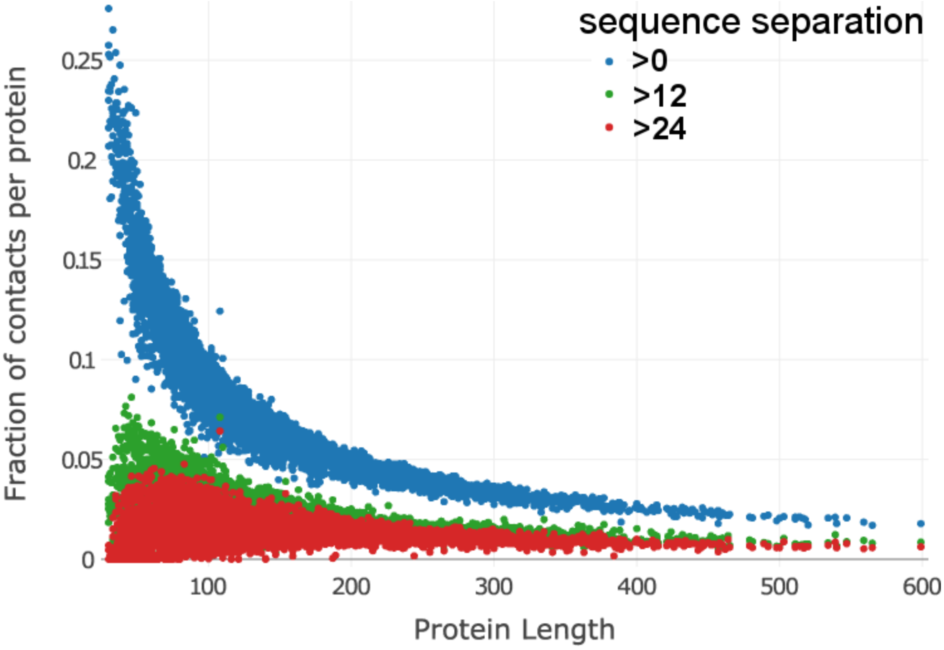


Figure 4.9: Fraction of contacts among all possible contacts in a protein against protein length L. The distribution has a non-linear relationship. At a sequence separation >8 positions the fraction of contacts for intermediate size proteins with length >100 is approximately 2%. Data set contains 6368 proteins and is explained in methods section @ref(data set).

4.6.3 Cross-validation for Random Forest Training

Proteins constitute highly imbalanced data sets with respect to the number of residue pairs that form and do not form physical contacts. As can be seen in Figure 4.9, 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 4.4 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 4.4: 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) [69]	SVM	1:4
Li et al. (2011) [70]	Random Forest	1:1, 1:2
Wang et al. (2011) [71]	Random Forest	1:4
DiLena et al. (2012) [79]	deep neural network	1:≈4 (sampling 20% of non-contacts)
Wang et al. (2013) [72]	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 2.6), so that the training set consists of five subsets that are non-redundant 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)` [238] and trained the models on features extracted from MSAs which are listed in methods section 4.6.1.

4.6.4 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.

5

A Bayesian Statistical Model for Residue-Residue Contact Prediction

All methods so far predict contacts by finding the one solution of parameters v_{ia} and w_{ijab} that maximizes a regularized version of the log likelihood of the MSA and in a second step transforming the MAP estimates of the couplings $\mathbf{w}^*$ into heuristic contact scores (see Introduction 1.3.5). Apart from the heuristic transformation that omits meaningful information comprised in the coupling matrices $\mathbf{w}_{ij}$ as discussed in section 2, using the MAP estimate of the parameters instead of the true distribution has the decisive disadvantage of concealing the uncertainty of the estimates.

The next sections present the derivation of a principled Bayesian statistical approach for contact prediction eradicating these deficiencies. The model provides estimates of the posterior probability distributions of contact states c_{ij} for all residues pairs i and j, given the MSA $\mathbf{X}$. A true contact (contact state $c_{ij} = 1$) 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 (contact state $c_{ij} = 0$). 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 explicitly model the dependence of coupling coefficients $\mathbf{w}_{ij}$ on contacts/non-contacts as a mixture of Gaussians with contact state dependent mixture weights and thus can even learn correlations between couplings. Furthermore, it provides probability estimates for the predicted contacts that could simplify the selection of constraints for *de novo* structure prediction by establishing suitable probability cutoffs.

5.1 Computing the Posterior Probabilty of a Contact

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 4.6.1), can be written as a hierarchical Bayesian model of the form:

$$p(\mathbf{c}, \mathbf{v}, \mathbf{w}|\mathbf{X}, \phi) \propto p(\mathbf{X}|\mathbf{v}, \mathbf{w}) p(\mathbf{v}, \mathbf{w}|\mathbf{c})\, p(\mathbf{c}|\phi)\,. \tag{5.1}$$

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,

$$p(\mathbf{c}|\mathbf{X}, \phi) \propto p(\mathbf{X}|\mathbf{c}) p(\mathbf{c}|\phi) \tag{5.2}$$

$$p(\mathbf{X}|\mathbf{c}) = \int\int p(\mathbf{X}|\mathbf{v}, \mathbf{w})\, p(\mathbf{v}, \mathbf{w}|\mathbf{c})\, d\mathbf{v}\, d\mathbf{w} \; . \tag{5.3}$$

The single potentials $\mathbf{v}$ will be fixed at their best estimate $\mathbf{v}^*$ (see method section 3.8.4) by using a very tight prior $p(\mathbf{v}) = \mathcal{N}(\mathbf{v}|\mathbf{v}^*, \lambda_v^{-1}\mathbf{I}) \to \delta(\mathbf{v} - \mathbf{v}*)$ for $\lambda_v \to \infty$ that acts as a delta function. This allows the replacement of the integral 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 5.2. The prior over the *Potts* model parameters then yields,

$$p(\mathbf{v}, \mathbf{w}|\mathbf{c}) = \mathcal{N}(\mathbf{v}|\mathbf{v}^*, \lambda_v^{-1}\mathbf{I}) \prod_{1 \le i < j \le L} p(\mathbf{w}_{ij}|c_{ij}) \; . \tag{5.4}$$

Furthermore, method section 5.7.2 proposes an approximation to the regularized 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. (5.3). The detailed derivation of the solution to the integral is covered in method section 5.7.3.

Finally, the marginals $p(c_{ij}|\mathbf{X}, \phi) = \int p(\mathbf{c}|\mathbf{X}, \phi) d\mathbf{c}_{\backslash ij}$, where $\mathbf{c}_{\backslash ij}$ is the vector containing all coordinates of $\mathbf{c}$ except c_{ij} can be computed to obtain the posterior probability distribution of the contact states (see method section 5.4).

5.2 Modelling the Prior Over Couplings Depending on Contact States

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 normalized weights $g_k(c_{ij})$ that depend on the contact state c_{ij},

$$p(\mathbf{w}_{ij}|c_{ij}) = \sum_{k=0}^{K} g_k(c_{ij})\, \mathcal{N}(\mathbf{w}_{ij}|\mu_k, \mathbf{\Lambda}_k^{-1}) \, . \tag{5.5}$$

The assumption that the contact-state dependent coupling prior can be modelled as a multivariate Gaussian is justified by the analysis of single and 2-dimensional coupling distributions presented in section 2.2 and in section 2.4. The couplings w_{ijab} for the analysis presented in those sections have been filtered, such that there is sufficient evidence for a and b in the alignment (see method section 2.6.6 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. The analysis showed that the univariate distributions of single couplings w_{ijab} are characteristic for the physico-chemical properties of the corresponding amino acid pairing (a, b) and vary with inter-residue distance. More than that, the 2-dimensional distributions suggest that there are higher order dependencies between the 400 couplings w_{ijab} that reflect amino acid specific preferences of the interaction between the corresponding residues i and j. By explicitly modelling the prior over couplings, $p(\mathbf{w}_{ij}|c_{ij})$, as a 400-dimensional Gaussian mixture, is is possible to capture these characteristic interdependencies between the couplings.

The K 400-dimensional Gaussian mixture components are defined by means $\mu_k \in \mathbb{R}^{400}$, precision matrices $\mathbf{\Lambda}_k \in \mathbb{R}^{400 \times 400}$, and normalized weights $g_k(c_{ij})$ that depend on the contact state $c_{ij} \in \{0, 1\}$. The zeroth component is expected to capture the majority of coupling parameters without a strong covariation signal, $w_{ijab} \approx 0$. Generally, the couplings are expected to vanish for non-contacts $(c_{ij} = 0)$ but couplings will also be close to zero for contacts $(c_{ij} = 1)$ when there is no covariation between residues i and j or when there is no evidence in the alignment originating from amino acid pairings a and b. Therefore, $\mu_0 = 0$ will be kept fixed. Furthermore, the precision matrices $\mathbf{\Lambda}_k$ will be modelled as diagonal matrices, thereby drastically reducing the computational complexity of the optimization problem. In order to ensure that interdependencies between couplings can be modelled with diagonal precision matrices, the number of components K is a crucial parameter.

5.3 Training the Hyperparameters in the Likelihood Function of Contact States

Solving the integral in eq. (5.3) as described in in detail in method section 5.7.3, yields the likelihood function of contact states, $p(\mathbf{X}|\mathbf{c})$. It contains the hyperparameters of the prior over couplings, $p(\mathbf{w}_{ij}|c_{ij})$, which is modelled as a mixture of K 400-dimensional Gaussians with component weights that depend on the contact state.

The hyperparameters are trained by minimizing the negative logarithm of the likelihood over a set of training MSAs as described in detail in method section 5.7.5. The MAP estimates of the coupling parameters $\mathbf{w}^*_{ij}$ are needed to compute the Hessian of the regularized *Potts* model likelihood, which again is needed for the Gaussian approximation to the regularized likelihood (see method section 5.7.2). For that purpose, I trained the hyperparameters by utilizing couplings $\mathbf{w}^*_{ij}$ obtained from pseudo-likelihood maximization as well as couplings $\mathbf{w}^*_{ij}$ obtained by maximizing the full likelihood with contrastive divergence (CD).

In the following I present the results of learning the hyperparameters for the coupling prior modelled as a Gaussian mixture with $K \in \{3, 5, 10\}$ Gaussian components with diagonal precision matrices $\mathbf{\Lambda}_k$ and a zero-component that is fixed at $\mu_0 = 0$ on data sets of different sizes (see method section 5.7.5.1 for details).

5.3.1 Training Hyperparameters for a Gaussian Mixture with Three Components

Training of the hyperparameters for three component Gaussian mixtures based on pseudo-likelihood and contrastive divergence couplings converged after several hundreds of iterations. The inferred hyperparameters obtained by several independent optimization runs and on the data sets of different size (10000, 100000, 3000000, 500000 residue pairs per contact class) are consistent. The following analysis is conducted for the training on the data set with 300,000 residue pairs per contact class and by using pseudo-likelihood couplings for estimation of the Hessian.

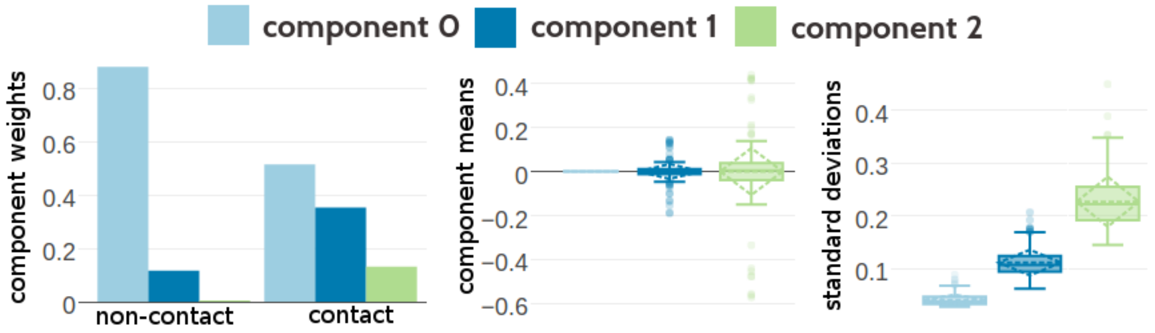


Figure 5.1: Statistics for the hyperparameters, $\gamma_k(c_{ij})$, μ_k and $\mathbf{\Lambda}_k$ of a three component Gaussian mixture obtained after 331 iterations. Trained on 300,000 residue pairs per contact class and using *pseudo-likelihood* couplings to estimate the Hessian. **Left** Component weights $\gamma_k(c_{ij})$ for residue pairs not in physical contact ($c_{ij}=0$) and true contacts ($c_{ij}=1$). **Center** Distribution of the 400 elements in the mean vectors μ_k. **Right** Distribution of the 400 standard deviations corresponding to the square root of the diagonal of $\mathbf{\Lambda}_k^{-1}$.

Figure 5.1 shows the statistics of the inferred hyperparameters. The zeroth component, with $\mu_0 = 0$ has a weight of 0.88 for the non-contact class, whereas it has only weight 0.51 for the contact class. This is expected given that the couplings w_{ijab} for non-contacts have a much tighter distribution around zero than contacts. Component 2 has on average the highest standard deviations and for several dimensions this component is located far off from zero, e.g. dimension EE has $\mu_2(\text{EE}) = -0.57$ or ER has $\mu_2(\text{ER}) = 0.43$. Therefore, it is not surprising that component 2 has a low weight for non-contacts ($g_2(0) = 0.0026$) but a higher weight for contacts ($g_2(1) = 0.13$). Statistics of the Gaussian mixture hyperparameters learned on the other data sets is shown in Appendix Figures G.2 and G.3. The inferred hyperparameters for the Gaussian mixture model based on couplings optimized with contrastive divergence are consistent with the estimates obtained by using pseudo-likelihood couplings as can be seen in Appendix Figures G.5 ans G.4.

Figure 5.2 shows several one-dimensional projections of the 400 dimensionl Gaussian mixture with three components. Generally, the Gaussian mixture learned for residue pairs that are not in contact is much narrower and almost symmetrically centered around zero. The Gaussian mixture for contacts, by contrast is much broader and often skewed. For the aliphatic amino acid pair (V,I), the Gaussian mixture for both contacts and non-contacts is very symmetrical and much narrower compared for example to the Gaussian mixtures for the aromatic amino acid pair (F,W), which also has symmetrical distributions. In contrast, the distribution of couplings for amino acid pairs (E,R) and (E,E) has strong tails for positive and negative values respectively. The one-dimensional projections of the Gaussian mixture model greatly resemble the empirical distributions of couplings illustrated in Figure 2.3 and in Figure 2.6 in chapter 2. The Gaussian mixtures learned on larger data sets produce very similar distributions (see Appendix Figures G.6 and G.7). Likewise, the distributions from the Gaussian mixture models that have been learned based on contrastive divergence couplings are also very similar and shown in Appendix Figure G.8.

Distributions from the two-dimensional projection of the Gaussian mixture model are shown in Figure 5.3 for several pairs of couplings. The type of paired couplings has been chosen to allow a direct comparison to the empirical distributions in Figure 2.13 in chapter 2. The top left plot shows the distribution of sampled coupling values according to the Gaussian mixture model for contacts for amino acid pairs (E,R) and (R,E). Component 2 has a weight of 0.13 for contacts and is mainly responsible for the positive coupling between (E,R) and (R,E). The

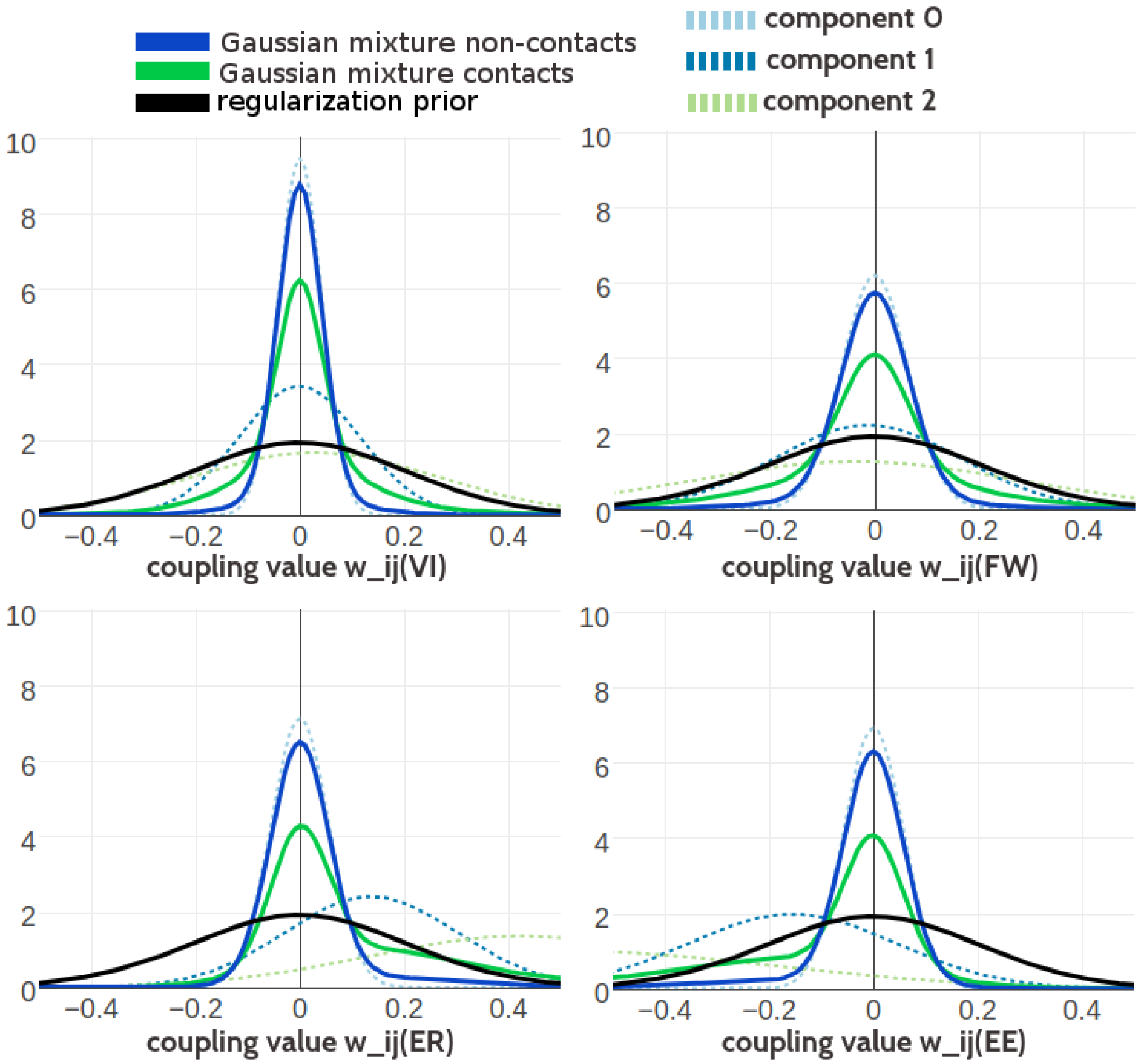


Figure 5.2: Visualization of one-dimensional projections of the three-component Gaussian mixture model for the contact-dependent coupling prior. Hyperparameters, $\gamma_k(c_{ij})$, μ_k and $\mathbf{\Lambda}_k$, have been trained on 300,000 residue pairs per contact class and using *pseudo-likelihood* couplings to estimate the Hessian. Green solid line: Gaussian mixture for contacts. Blue solid line: Gaussian mixture for non-contacts. Black solid line: regularization prior with $\lambda_1 = 0.2L$ with L being protein length and assumed $L = 150$. Dashed lines: unweighted probability densities of Gaussian components with color code specified in the legend. **Top Left** One dimensional projection for pair (V,I). **Top Right** One dimensional projection for pair (F,W). **Bottom Left** One dimensional projection for pair (E,R). **Bottom Right** One dimensional projection for pair (E,E).

amino acid pairs (E,E) and (R,E) are negatively coupled and again component 2 generates the strongest couplings far off zero as can be seen in the top right plot. The plots at the bottom of Figure 5.3 show the distribution of sampled couplings for amino acid pairs (I,L) and (V,I) according to the Gaussian mixture model for contacts (component weight $g_k(1)$) as well as for non-contacts (component weight $g_k(0)$). The coupling distribution for contacts is symmetrically centered around zero just as the distribution for non-contacts. Because of the higher weight of component 2, the distribution for contacts is much broader than the distribution for non-contacts. The two-dimensional distributions of sampled couplings obtained from the set of Gaussian mixture hyperparameters that have been learned based on contrastive divergence couplings, are very similar and shown in Appendix Figure G.9

5.3.2 Training Hyperparameters for a Gaussian Mixture with Five and Ten Components

The increased complexity of training five or even ten instead of three component Gaussian mixtures does not only result in longer runtimes until convergence but also slows down runtime per iteration. The optimization runs for five and ten component Gaussian mixtures did not converge within 2000 iterations. Nevertheless, the obtained hyperparameters and resulting Gaussian mixture are consistent, as will be shown in the following.

Figure 5.4 and 5.5 show the statistics of the inferred hyperparameters for a five and ten component Gaussian mixture, respectively. Similarly to three component Gaussian mixtures, the zeroth component receives a high weight for couplings from residue pairs that are not in physical contact ($g_0(0) = 0.93$ for five component mixture and $g_0(0) = 0.87$ for ten component mixture). There is a second component with a noteworthy contribution to the Gaussian mixture for non-contact couplings (component 3 with $g_3(0) = 0.07$ for five component mixture and component 9 with $g_9(0) = 0.13$ for ten component mixture). These two components are also the strongest components for the Gaussian mixture representing couplings from contacting residue pairs. The inferred hyperparameters for Gaussian mixture models based on couplings optimized with contrastive divergence are consistent with the estimates obtained by using pseudo-likelihood couplings as can be seen in Appendix Figures G.10 and G.11.

Figure 5.6 compares the one-dimensional projections of the 400 dimensionl Gaussian mixtures with five and ten components for the amino acid pairs (V,I) and (E,R). The general observations regarding the shape of the Gaussian mixture for couplings from contacts and non-contacts that have been found for the three component mixture also apply here. Generally, the Gaussian mixture for couplings from non-contacts is narrower in the five and ten component mixtures than in the three component Gaussian mixture model. Thereby, the differentiation between contacts and non-contacts is enhanced because the ratio between the Gaussian mixture probability distribution for contacts and non-contacts increases. Furthermore, whereas in the three component model only two components would contribute to defining the tails of the distribution for couplings from contacts, now there are more components that can refine the tails. For example, in the case of amino acid pair (E,R) all but the zeroth component, which is fixed at zero, are shifted towards positive values. In the case of amino acid pair (V,I) the components are shifted towards both positive and negative values. Overall, the Gaussian mixtures with five and ten components seem to refine the modelling of the coupling distributions compared to the simpler three component model. The same observations apply to the one-dimensional projections of the Gaussian mixtures inferred based on contrastive divergence couplings only that the resultant mixtures are even narrower (see Appendix Figure G.12).

Two-dimensional projections of the Gaussian mixture with five and ten components are shown in Figure 5.7 for different pairs of couplings. The distributions resemble the ones learned for

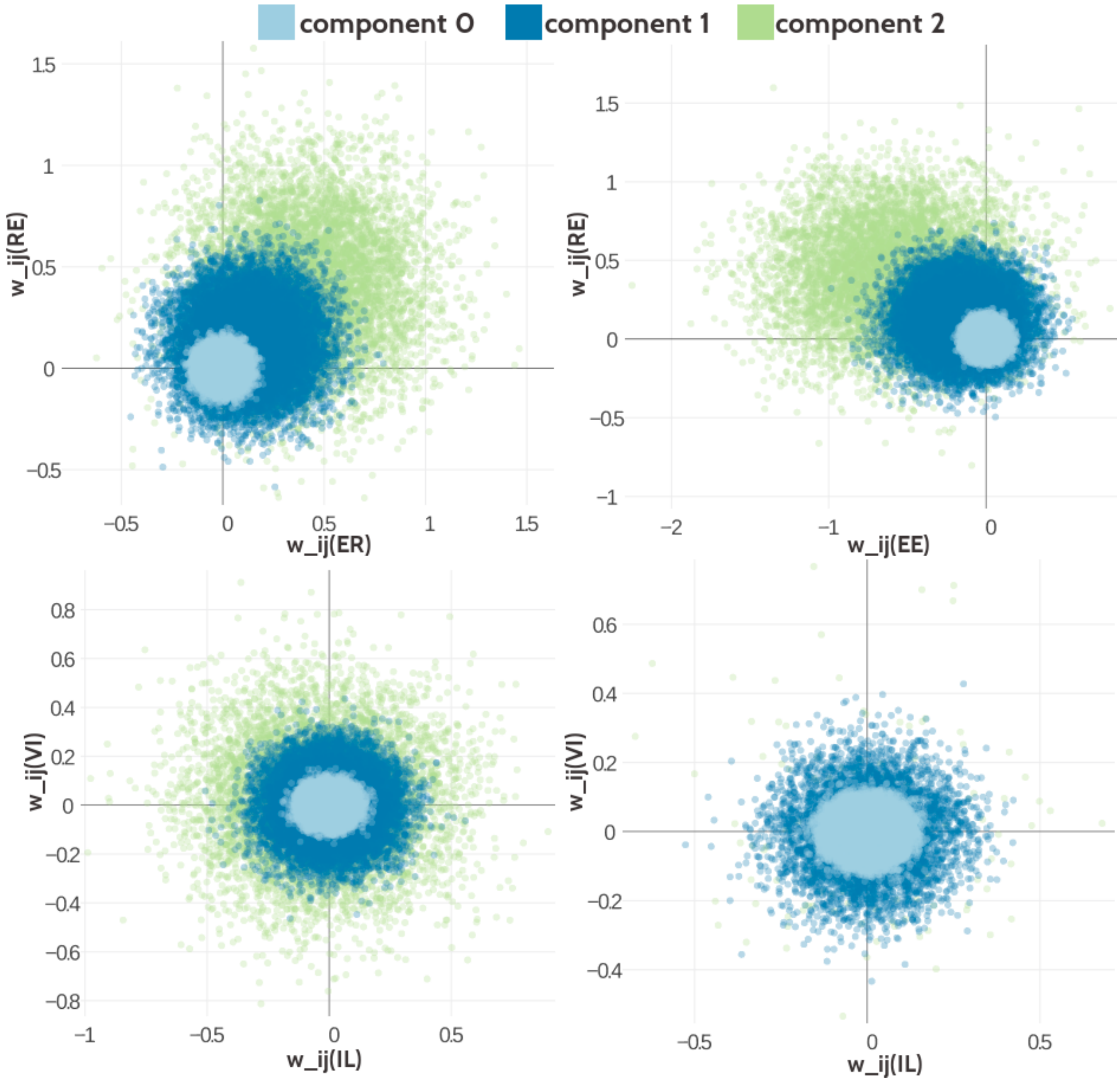


Figure 5.3: Visualization of two-dimensional projections of the three-component Gaussian mixture model for the contact-dependent coupling prior. Hyperparameters, $\gamma_k(c_{ij})$, μ_k and $\mathbf{\Lambda}_k$, have been trained on 300,000 residue pairs per contact class and using *pseudo-likelihood* couplings to estimate the Hessian. 10,000 paired couplings have been sampled from the Gaussian mixture model. The different colors represent the generating component and color code is specified in the legend. **Top Left** Two-dimensional projection for pairs (E,R) and (R-E) for contacts (using component weight $g_k(1)$). **Top Right** Two- dimensional projection for pairs (E,E) and (R,E) for contacts (using component weight $g_k(1)$). **Bottom Left** Two-dimensional projection for pairs (I,L) and (V,I) for contacts (using component weight $g_k(1)$). **Bottom Right** Two-dimensional projection for pair (I,L) and (V,I) for non-contacts (using component weight $g_k(0)$).

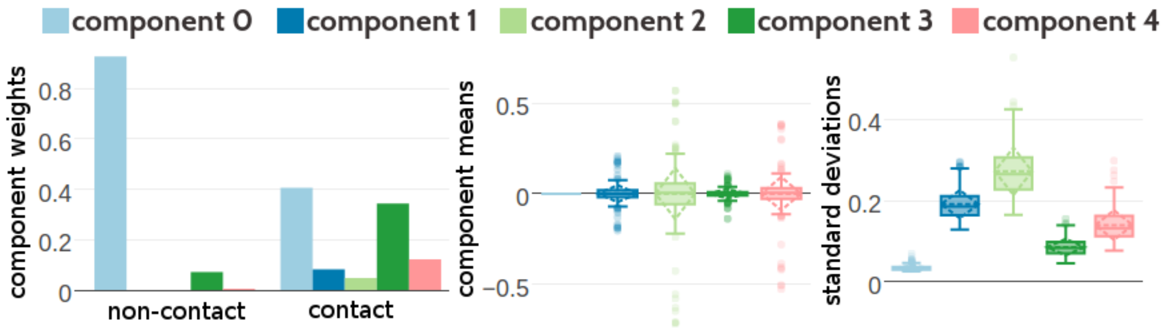


Figure 5.4: Statistics for the hyperparameters, $\gamma_k(c_{ij})$, μ_k and $\mathbf{\Lambda}_k$ of a five component Gaussian mixture obtained after 1134 iterations. Trained on 300,000 residue pairs per contact class and using *pseudo-likelihood* couplings to estimate the Hessian. **Left** Component weights $\gamma_k(c_{ij})$ for residue pairs not in physical contact ($c_{ij} = 0$) and true contacts ($c_{ij} = 1$). **Center** Distribution of the 400 elements in the mean vectors μ_k. **Right** Distribution of the 400 standard deviations corresponding to the square root of the diagonal of $\mathbf{\Lambda}_k^{-1}$.

the Gaussian mixture with three components. However, it is visible that the zeroth component is narrower for the five and ten component Gaussian mixture and that the additional components model particular parts of the distribution. For example, component 9 in the ten component Gaussian mixture model produces couplings for amino acid pairs (E-R) and (R-E) that are close to zero in both dimensions or even slightly negative. The Gaussian mixture of the coupling prior that has been learned based on couplings computed with contrastive divergence in general produces distributions that are narrower which is expected given the hyperparameter statistics and the observations from the univariate distributions.

In conclusion it can be found that training of the hyperparameters for the Gaussian mixtures of the contact-dependent coupling prior seems to be robust. Training consistently yields comparable hyperparameter settings and the Gaussian mixtures produce similar distributions regardless of the data set size and repeated independent runs. The Gaussian mixture repeatedly reproduce the empirical distribution of couplings shown in Figure 2.13 in chapter 2 very well. Of course it must be noted that the empirical distributions do not take the uncertainty of the inferred couplings into account. They are computed for high evidence couplings as explained in method section 2.6.7 and therefore do not provide a completely correct reference. Besides, looking at two dimensional projections of the 400 dimensional Gaussian mixture model can only provide a limited view of the high-dimensional interdependencies. Another restricting issue is run time. The more components define the Gaussian mixture, the longer it takes to train the model per iteration and the more iterations it takes to reach convergence. However, without reaching convergence it cannot be assured that the identified hyperparameters for five and ten component Gaussian mixtures represent optimal estimates.

5.4 Evaluating the Bayesian Models for Contact Prediction

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

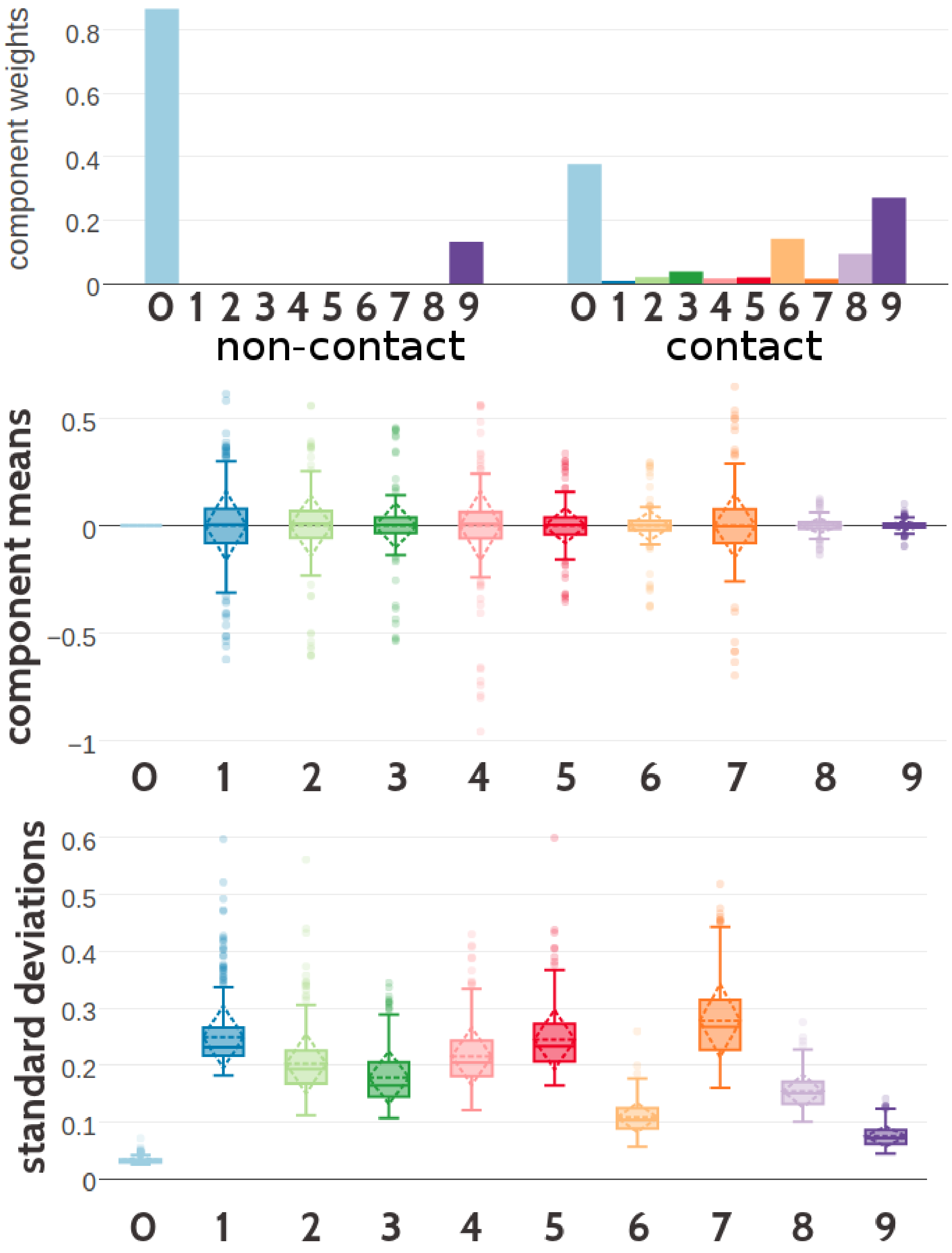


Figure 5.5: Statistics for the hyperparameters, $\gamma_k(c_{ij})$, μ_k and $\mathbf{\Lambda}_k$ of a ten component Gaussian mixture obtained after 700 iterations. Trained on 300,000 residue pairs per contact class and using *pseudo-likelihood* couplings to estimate the Hessian. X-axis represents the ten components numbered from 0 to 9. **Top** Component weights $\gamma_k(c_{ij})$ for residue pairs not in physical contact ($c_{ij} = 0$) and true contacts ($c_{ij} = 1$). **Middle** Distribution of the 400 elements in the mean vectors μ_k. **Bottom** Distribution of the 400 standard deviations corresponding to the square root of the diagonal of $\mathbf{\Lambda}_k^{-1}$.

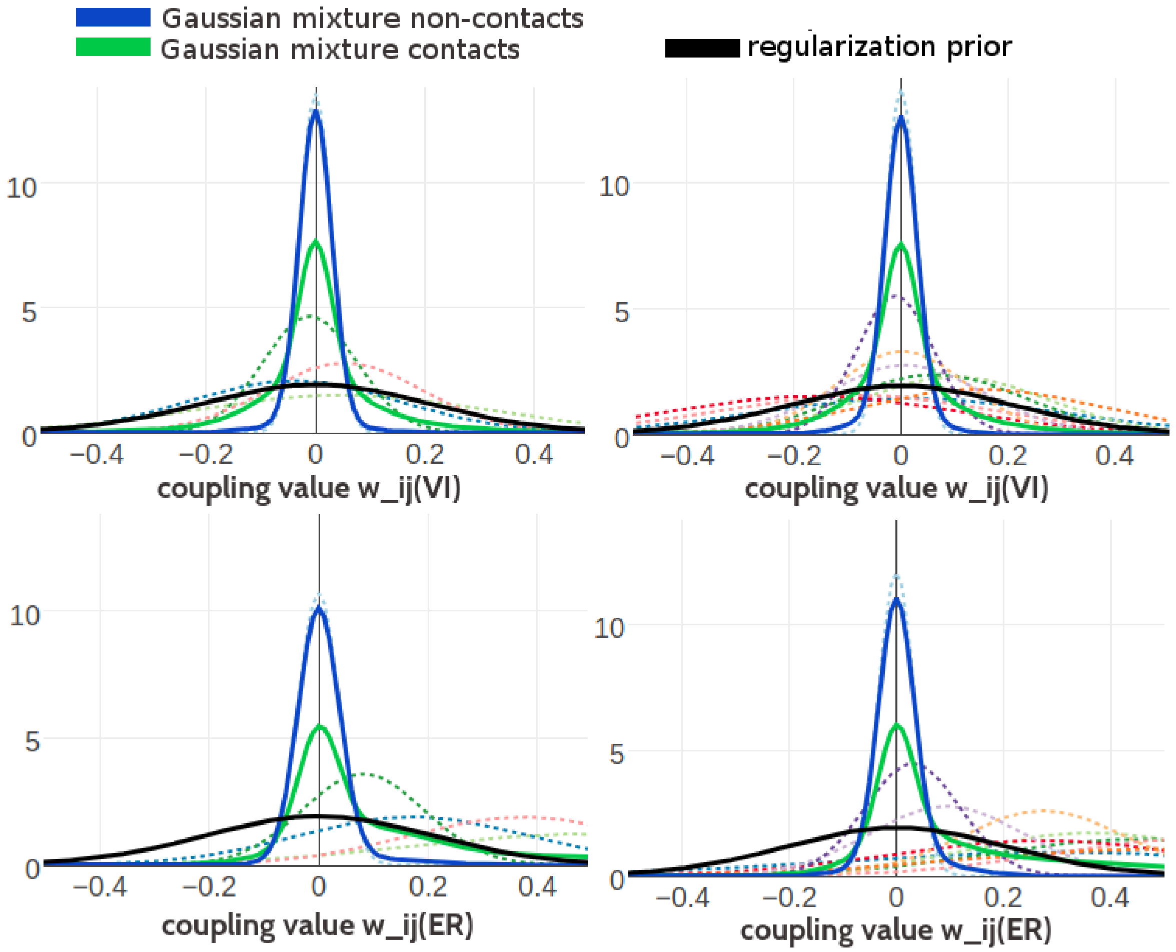


Figure 5.6: Visualization of one-dimensional projections of the five and ten component Gaussian mixture model for the contact-dependent coupling prior. Hyperparameters, $\gamma_k(c_{ij})$, μ_k and $\mathbf{\Lambda}_k$, have been trained on 300,000 residue pairs per contact class and using *pseudo-likelihood* couplings to estimate the Hessian. Green solid line: Gaussian mixture for contacts. Blue solid line: Gaussian mixture for non-contacts. Black solid line: regularization prior with $\lambda_1 = 0.2L$ with L being protein length and assumed $L = 150$. Dashed lines represent the unweighted Gaussian mixture components. **Top Left** One dimensional projection for pair (V,I) from the five component model. **Top Right** One dimensional projection for pair (V,I) from the ten component model. **Bottom Left** One dimensional projection for pair (E,R) from the five component model. **Bottom Right** One dimensional projection for pair (E,R) from the ten component model.

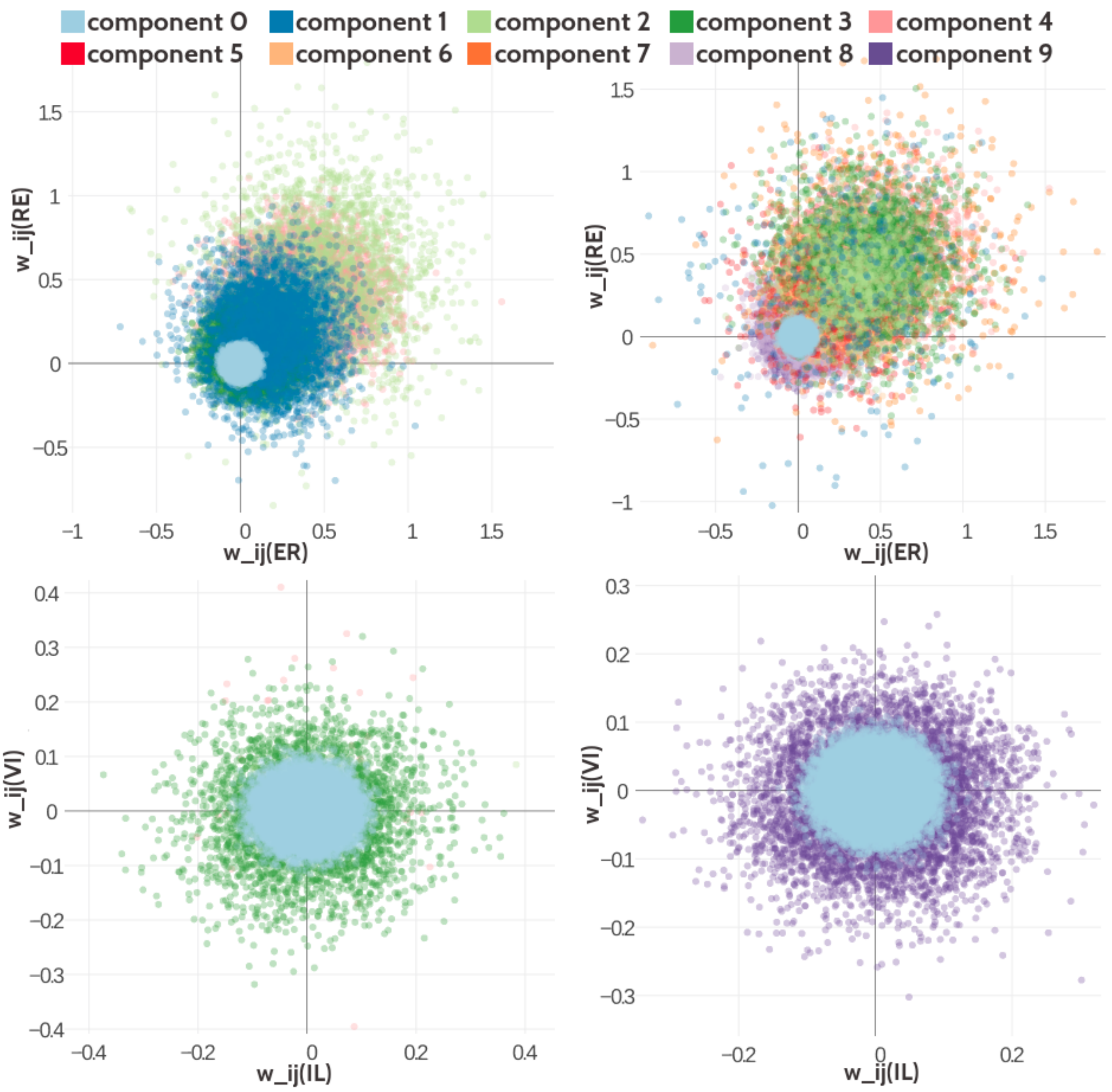


Figure 5.7: Visualization of two-dimensional projections of the five and ten component Gaussian mixture model for the contact-dependent coupling prior. Hyperparameters, $\gamma_k(c_{ij})$, μ_k and $\mathbf{\Lambda}_k$, have been trained on 300,000 residue pairs per contact class and using *pseudo-likelihood* couplings to estimate the Hessian. 10,000 paired couplings have been sampled from the Gaussian mixture model. The color of a sampled coupling pair represents the Gaussian mixture component that has generated this sample point. Color code is specified in the legend. **Top Left** Two-dimensional projection for pairs (E,R) and (R-E) for contacts (using component weight $g_k(1)$) from the five component Gaussian mixture model. **Top Right** Two-dimensional projection for pairs (E,R) and (R-E) for contacts (using component weight $g_k(1)$) from the ten component Gaussian mixture model. **Bottom Left** Two-dimensional projection for pair (I,L) and (V,I) for non-contacts (using component weight $g_k(0)$) from the five component Gaussian mixture model. **Bottom Right** Two-dimensional projection for pair (I,L) and (V,I) for non-contacts (using component weight $g_k(0)$) from the ten component Gaussian mixture model.

$$
\begin{aligned}
p(c_{ij}|\mathbf{X},\phi) &= \int d\mathbf{c}_{\backslash ij}\, p(\mathbf{c}|\mathbf{X},\phi) \\
&\propto \int d\mathbf{c}_{\backslash ij}\, p(\mathbf{X}|\mathbf{c})\, p(\mathbf{c}|\phi) \\
&\propto \int d\mathbf{c}_{\backslash ij} \prod_{i'<j'} \sum_{k=0}^{K} g_k(c_{i'j'}) \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} \prod_{i'<j'} p(c_{i'j'}|\phi_{i'j'})\,, \qquad (5.6)
\end{aligned}
$$

where $p(\mathbf{c}|\phi)$ represents a prior on contacts that is implemented by the random forest classifier trained on sequence derived features, ϕ, as described in chapter 4. By pulling the term depending only on the contact state c_{ij} out of the integral over $\mathbf{c}_{\backslash ij}$, one obtains the posterior distribution for c_{ij},

$$
\begin{aligned}
p(c_{ij}|\mathbf{X},\phi) &\propto p(c_{ij}|\phi_{ij}) \sum_{k=0}^{K} g_k(c_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} \\
&\times \prod_{i'<j',(i',j')\neq(i,j)} \int dc_{i'j'}\, p(c_{i'j'}|\phi_{i'j'}) \sum_{k=0}^{K} g_k(c_{i'j'}) \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})}\,. \qquad (5.7)
\end{aligned}
$$

Since the second factor involving the integrals over $c_{i'j'}$ is a constant with respect to c_{ij}, it can be written,

$$
p(c_{ij}|\mathbf{X},\phi) \propto p(c_{ij}|\phi_{ij}) \sum_{k=0}^{K} g_k(c_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})}\,. \qquad (5.8)
$$

A predicted contact map is obtained by using the posterior probability estimate for a contact, $p(c_{ij}=1|\mathbf{X},\phi)$, as an entry in the matrix for residue pair (i,j).

In the following I am going to assess the performance of the Bayesian models with hyperparameters learned using couplings from pseudo-likelihood maximization. The performance will be evaluated with respect to the precision of the top ranked contact predictions, whereby ranking of predictions now follows the posterior probability estimates for contacts.

Figure 5.8 shows a benchmark for the Bayesian models using a three component Gaussian mixture model for the coupling prior and with hyperparameters trained on different data set sizes (100,000, 300,000 and 500,000 residue pairs per contact class). The analysis of the Gaussian mixture models in the last sections has revealed that the statistics and resultant distributions are coherent regardless of data set size. And indeed, the precision over top ranked predictions is almost indistinguishable for the models learned on different data set sizes. The Gaussian mixture model with three components has 2004 parameters (see methods section 5.7.5.2) and it is reasonable to learn this many parameters given a data set of 2x 100,000 residue pairs even considering the unknown uncertainty of the couplings to be modelled.

Because the posterior probability of a contact utilizes additional information from the contact prior in form of the random forest classifier (see chapter 4), it is not fair to compare the posterior probabilities directly to the pseudo-likelihood derived contact scores. Instead, the predictions from the Bayesian model can be compared to the random forest model that has additionally been trained on the pseudo-likelihood derived contact scores (see section 4.4). As can be seen in Figure 5.8, the Bayesian model predicts contacts more accurately than the

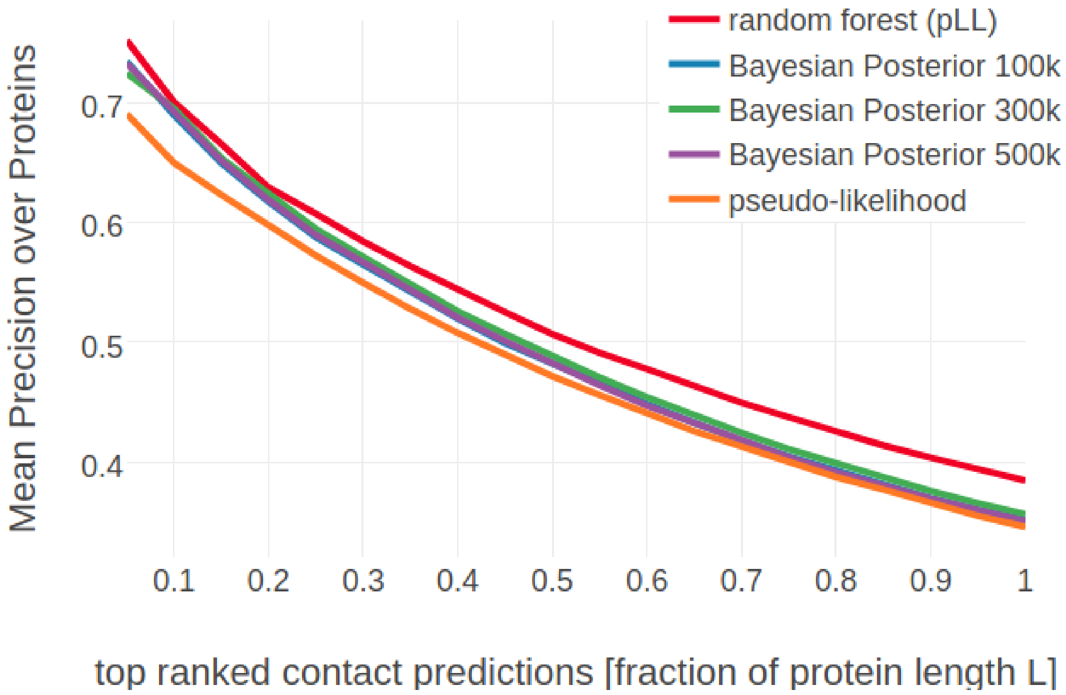


Figure 5.8: Mean precision for top ranked contact predictions over 500 proteins. The "Bayesian Posterior" methods compute the posterior probability of contacts with the Bayesian framework employing a three component Gaussian mixture coupling prior. Hyperparameters for the coupling prior have been trained on different data set sizes as specified in the legend. **random forest (pLL)** random forest model trained on sequence features and and additional pseudo-likelihood contact score feature. **Bayesian Posterior 100k**: Trained on 100,000 residue pairs per contact class. **Bayesian Posterior 300k**: Trained on 300,000 residue pairs per contact class. **Bayesian Posterior 500k**: Trained on 500,000 residue pairs per contact class. **pseudo-likelihood**: contact score is computed as APC corrected Frobenius norm of the couplings computed from pseudo-likelihood.

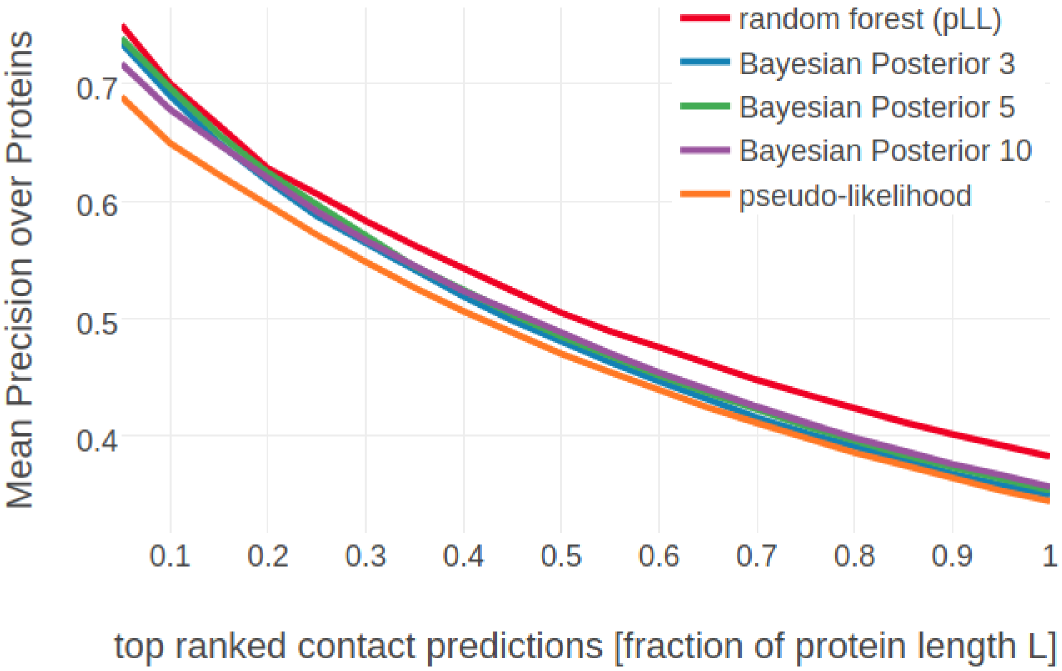


Figure 5.9: Mean precision for top ranked contact predictions over 500 proteins. The "Bayesian Posterior" methods compute the posterior probability of contacts with the Bayesian framework employing a Gaussian mixture coupling prior based on couplings computed with *pseudo-likelihood*. Hyperparameters for the coupling prior have been trained on 100,000 residue pairs per contact class. The number of Gaussian components in the Gaussian mixture model is specified in the legend. **random forest (pLL)** random forest model trained on sequence features and and additional pseudo-likelihood contact score feature. **Bayesian Posterior 3**: Bayesian model utilizing a three component Gaussian mixture. **Bayesian Posterior 5**: Bayesian model utilizing a five component Gaussian mixture. **Bayesian Posterior 10**: Bayesian model utilizing a ten component Gaussian mixture. **pseudo-likelihood**: contact score is computed as APC corrected Frobenius norm of the couplings computed from pseudo-likelihood.

heuristic contact score obtained from pseudo-likelihood couplings, but less accurately than the random forest model trained on sequence features and the pseudo-likelihood contact scores.

The likelihood function of contacts has been optimized with respect to the coupling prior hyperparameters using an equal number of residue pairs that are in physical contact and that are not in physical contact. The residue pairs that are not in physical contact have been defined on basis of a $25\AA$ C_β distance threshold. Choosing a different non-contact threshold, C_β distance $> 8\AA$, has a negligible impact on performance with the $25\AA$ C_β cutoff giving slightly better results (see Appendix Figure G.14). Furthermore, I checked whether a different ratio of contacts and non-contacts has an impact on performance. Appendix Figure G.14 also shows that choosing five times as many non-contacts as contacts gives slightly worse precision and has the disadvantage of longer run times.

Figure 5.9 compares the performance of Bayesian models with Gaussian mixtures having different number of components and trained on 100,000 residue pairs per contact class. The Bayesian model with a Gaussian mixture having five components shows minor improvements over the model with a three-component Gaussian mixture. Surprisingly, the Bayesian model with the ten component Gaussian mixture performs slightly worse than the other two models. This is unexpected, because the analysis in the last section indicated that both the five and the ten component Gaussian mixture models are able to precisely model the empirical

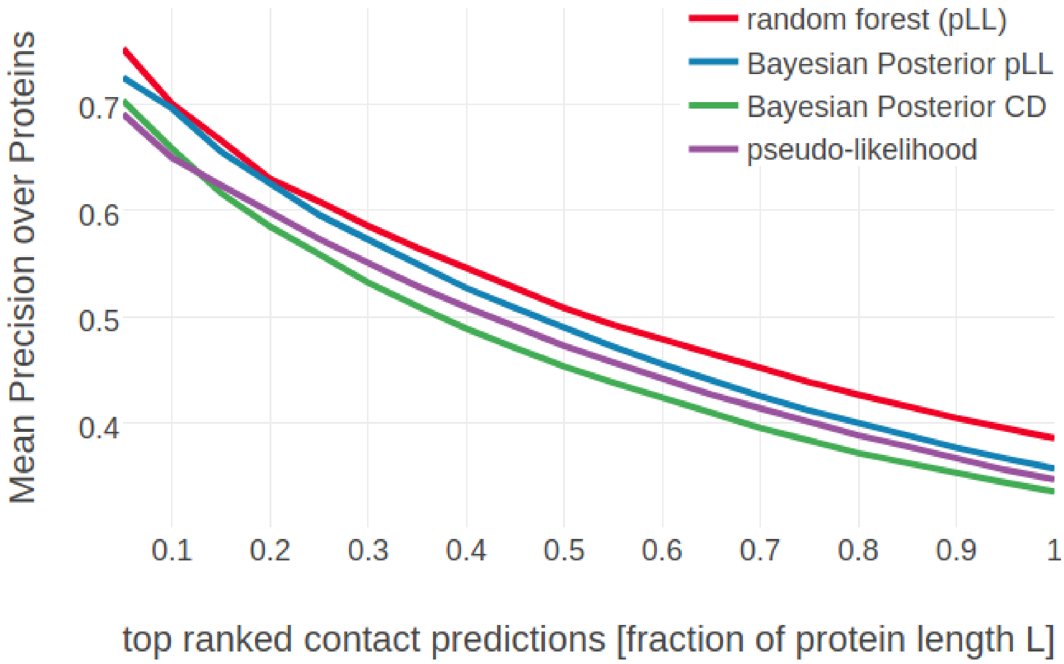


Figure 5.10: Mean precision for top ranked contact predictions over 500 proteins. The "Bayesian Posterior" methods compute the posterior probability of contacts with the Bayesian framework employing a three component Gaussian mixture coupling prior. Hyperparameters for the coupling prior have been trained on 100,000 residue pairs per contact class. **random forest (pLL)** random forest model trained on sequence features and and additional pseudo-likelihood contact score feature. **Bayesian Posterior pLL**: Bayesian model based on *pseudo-likelihood* couplings. **Bayesian Posterior CD**: Bayesian model based on *contrastive divergence* couplings. **pseudo-likelihood**: contact score is computed as APC corrected Frobenius norm of the couplings computed from pseudo-likelihood.

coupling distributions. However, it has also been pointed out before that training of the hyperparameters did not converge within several thousands of iterations and further training might be necessary for the five and ten component Gaussian mixture models.

The trends described for the Bayesian models based on pseudo-likelihood couplings also apply for the Bayesian models based on contrastive divergence couplings. In detail, the Bayesian models based on contrastive divergence couplings perform equally well, regardless of the size of the training set (see Appendix Figure G.15), the choice of the non-contact threshold or the number of Gaussian components (see Appendix Figure G.16). Rather surprising is the finding the Bayesian models based on contrastive divergence couplings perform worse than the ones based on pseudo-likelihood couplings (see Figure 5.10). In fact, they even have worse predictive power than the heuristic pseudo-likelihood contact score, though they involve prior information. This finding is unexpected given that a crucial approximation within the Bayesian framework employs the Hessian of the full likelihood (see method section 5.7.2) and not of the pseudo-likelihood. Therefore it is assumed that the approximation is more accurate when utilizing the couplings that have been obtained by maximizing the full likelihood with contrastive divergence. But apparently, the approximation works very well for pseudo-likelihood couplings.

It is interesting to note that the Bayesian models are mainly performing worse for proteins in the second Neff quartile which constitutes Neff values in the range $680 \leq \text{N}_{\text{eff}} < 2350$ (see Appendix Figure G.17). This finding applies to all Bayesian models, regardless of the method that was used to obtain the MAP estimate of couplings or the the number of Gaussian

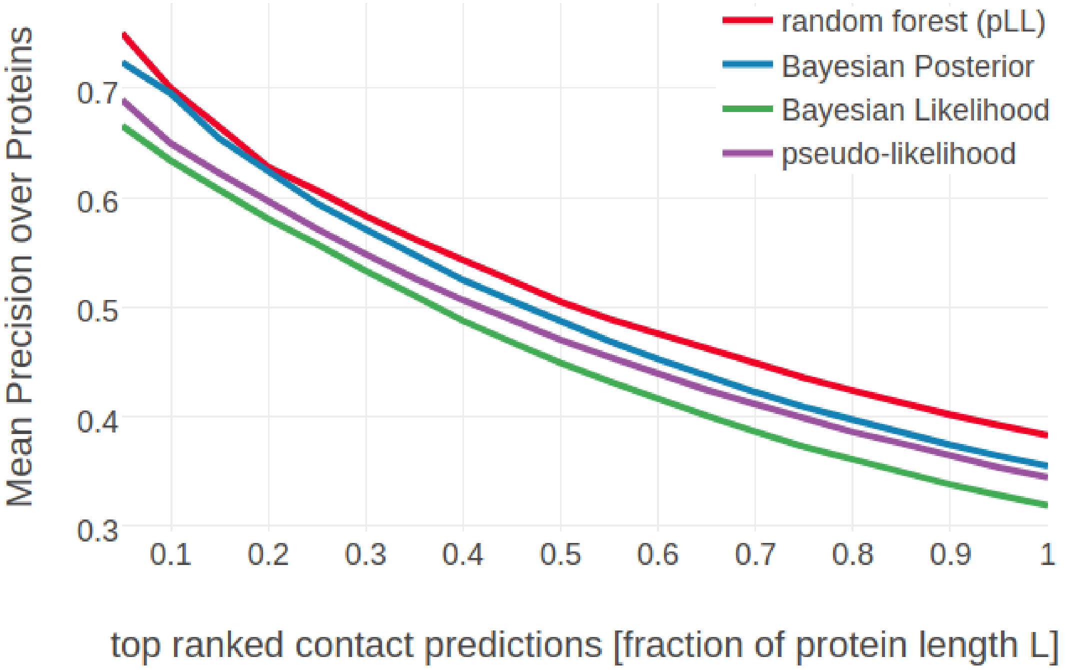


Figure 5.11: Mean precision for top ranked contact predictions over 500 proteins. **random forest (pLL)** random forest model trained on sequence features and and additional pseudo-likelihood contact score feature. **Bayesian Posterior**: Bayesian model computing the posterior probability of contacts with a three component Gaussian mixture coupling prior based on *pseudo-likelihood* couplings. Hyperparameters for the coupling prior have been trained on 300,000 residue pairs per contact class. **Bayesian Likelihood**: Log Likelihood of observing a contact as given in eq. (5.31). Coupling prior is modelled as three component Gaussian mixture based on *pseudo-likelihood* couplings. Hyperparameters for the coupling prior have been trained on 300,000 residue pairs per contact class. **pseudo-likelihood**: contact score is computed as APC corrected Frobenius norm of the couplings computed from pseudo-likelihood.

components used to model the coupling prior. A thorough inspection of proteins with Neff values within this particular range did not reveal any further insights.

5.5 Analysing Contact Maps Predicted With the Bayesian Model

In the following I will analyse the predictions from the Bayesian model utilizing a three component Gaussian mixture for the coupling prior and pseudo-likelihood couplings to approximate the regularized likelihood of sequences. While the posterior probabilities for contacts are used as predictions it is also worth having a look at the likelihood of contacts, given by eq. (5.31), to dissect the effect of likelihood and prior on the posterior. Figure 5.11 compares the precision of predictions given by the posterior probabilities and the log likelihoods for a contact. It can be found, that the log likelihood has decreased predictive performance compared to the heuristic contact score computed from pseudo-likelihood couplings.

Protein 1c75A00 has length L=71 and 28078 sequences in the alignment and is among the proteins with the highest number of effective sequences (Neff=16808 $>$ 95th percentile). The performance of the different methods for this protein reflects the ranking of methods in the overall benchmark. The Bayesian posterior probabilities achieve comparable performance as the heuristic pseudo-likelihood contact score computed as the APC corrected Frobenius

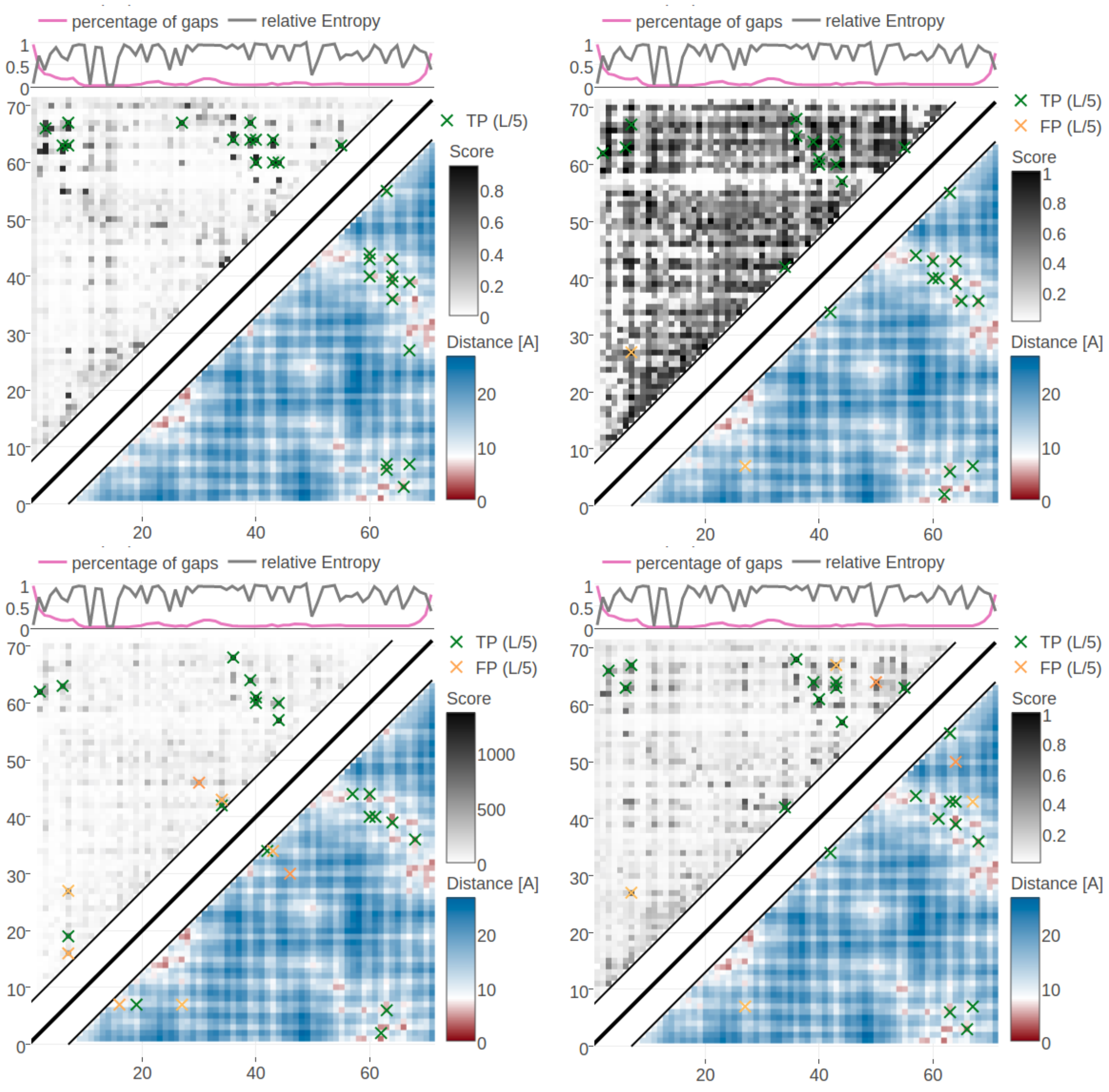


Figure 5.12: Contact maps predicted for protein 1c75A00. Upper left matrices show predicted contact maps and lower right matrices show the native distance maps. **Top Left** Contact map computed from probabilities of contacts as given by random forest model that has been trained on sequence features and pseudo-likelihood contact scores **Top Right** Contact map computed from posterior probability estimates given by Bayesian model utilizing a three component Gaussian mixture model and is based on pseudo-likelihood couplings. **Bottom Left** Contact map computed from log likelihood of contacts according to the Bayesian model utilizing a three component Gaussian mixture model and is based on pseudo-likelihood couplings. **Bottom Right** Contact map computed from probabilities of contacts as given by random forest model that has been trained on sequence features only.

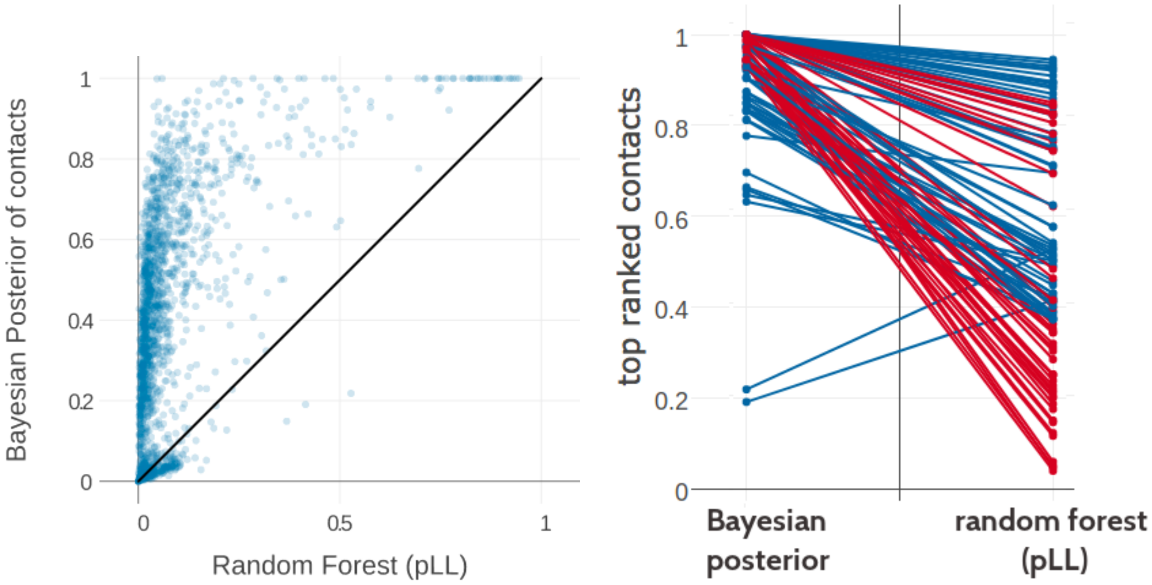


Figure 5.13: Comparing the predicted contact probabilities from the Bayesian model and the random forest model trained on sequence features and pseudo-likelihood couplings. **Left**: Probabilities for protein 1c75A00 predicted with Bayesian model using the posterior probability estimates for contacts and the random forest model trained on sequence features and pseudo-likelihood contact scores computed as APC corrected Frobenius norm of the couplings. **Right**: Comparing the ranking of top ranked contact predictions obtained from the Bayesian model and the random forest model trained on sequence features and pseudo-likelihood contact scores computed as APC corrected Frobenius norm of the couplings. Plot shows predictions for the top 71 (=L) predictions from either method. Identical residue pairs are connected with a line. Green indicates identical ranking of the residue pair for both methods. Blue indicates higher ranking of the residue pair for random forest model. Red indicates higher ranking of the residue pair for Bayesian model.

norm of the pseudo-likelihood couplings (see Appendix Figure G.18). The random forest model trained on both sequence features and the pseudo-likelihood contact score achieves the highest precision for the top L=71 contacts. It is remarkable to see that top 25 (=0.35L) predictions are correct. The predictions offered by the log likelihood of contacts give slightly worse results than the prediction given by the Bayesian posterior probabilities.

Figure 5.12 shows contact maps predicted from the posterior probabilities of contacts, the log likelihood of contacts and the random forest models trained only on sequence features and trained on both sequence features and the pseudo-likelihood contact score. The latter model predicts the 14 (=L/5) highest scoring contacts correctly which was already revealed in the benchmark plot for protein 1c75A00. The simple random forest model trained only on sequence features predicts many contacts in common with the other methods but also makes three false positive predictions. One of the incorrect predictions (i=27, j=7) also receives a high log likelihood and is consequently also wrongly predicted by the full Bayesian model. It can be observed that the Bayesian posterior probabilities for contacts are generally higher than the probabilities made by the random forest models. A more quantitative comparison of the probabilities for contacts obtained from the random forest model and from the Bayesian model is given in Figure 5.13.

The ranking of predicted contacts according to the probabilities is rather different for the random forest and the Bayesian model (see Figure 5.13). A straightforward possibility to improve overall contact prediction accuracy is to train another random forest model based on sequence features as well as the heuristic contact scores and the Bayesian posterior probabilities. However, the random forest model using several types of coevolution methods does not improve over the random forest model trained only on one of the scores (see Appendix Figure G.19). The same observation has been made for the random forest model involving both heuristic scores from pseudo-likelihood couplings and contrastive divergence couplings as described in section 4.4.

5.6 Discussion

The predicted contacts provided by the posterior contact probabilities of the Bayesian models proved to be less precise than those provided by the random forest trained on sequence features and pseudo-likelihood couplings. Even though the coupling prior modelled as a Gaussian mixture seems to reproduce the empirical distributions of high evident couplings very well, the current approach might comprise several weaknesses.

First of all, the Gaussian components are modelled with diagonal covariance matrices. Much more information can be learned by using full covariance matrices which would in turn require learning less components. However, using full covariance matrices would also increase computational complexity because the inverse of these matrices has to be computed. Furthermore, it is possible that the correlations between couplings are too sparse to exhibit a strong signal that can be efficiently learned. In that case it might be worth considering training on a data set that is more strictly filtered for evident couplings or on a reduced representation of the 400-dimensional coupling space. It is unlikely that overfitting is an issue during training. Not only has the neg log likelihood been monitored on a validation set during optimization, but also the consistent hyperparameter estimates regardless of the training set size speak against overfitting.

The assumption that the off-diagonal block matrices in the Hessian contain negligible information and therefore can be set to zero makes the Bayesian approach computational feasible (see method section 5.7.4). This assumption might represent an issue but currently we are not aware of how to quantitatively verify this assumption. These off-diagonal block matrices describe the interdependency between specific couplings in different pairs of columns. However, in our view the entries in these off-diagonal matrices should be negligible.

Another important point is that 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 (5.17).

The second point can be addressed quite effectively by learning a simple isotropic Gaussian prior with the same methodology that is used to infer the hyperparameters for the Gaussian mixture of the coupling prior. Since the new regularization prior will be very close to the mode of the integrand in the marginal likelihood, the Gaussian approximation to the regularized likelihood for the second iteration has improved in comparison to the first iteration. This procedure requires the generation of new coupling estimates by pseudo-likelihood maximization of by optimizing the full likelihood with contrastive divergence and thereby employing the new regularization prior.

A proof of concept that the full information in the coupling matrices can be used to improve the precision of contact predictions was given in the work of Golkov and colleagues

[239]. The developed a convolutional neural network for the prediction of protein residue-residue contacts that uses only coupling matrices as input features. In their benchmark the convolutional network predictor improved over Meta-PSICOV [85], which is a meta predictor combining several coevolution methods and sequence features. However, since their deep learning network resembles a black box machine learning approach further insights into the nature of interdependencies between couplings are barred. In contrast, the Bayesian statistical modeled presented here provides a principled approach that explicitly tries to model the underlying concepts and offers new realizations.

5.7 Methods

5.7.1 Modelling the Prior Over Couplings Depending on Contact States

The mixture weights $g_k(c_{ij})$ in eq. (5.5) are modelled as softmax:

$$g_k(c_{ij}) = \frac{\exp \gamma_k(c_{ij})}{\sum_{k'=0}^{K} \exp \gamma_{k'}(c_{ij})} \tag{5.9}$$

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 $\gamma_0(c_{ij}) = 1$.

5.7.2 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

$$p(\mathbf{w}|\mathbf{X}, \mathbf{v}^*) = p(\mathbf{X}|\mathbf{v}^*, \mathbf{w}) \mathcal{N}(\mathbf{w}|\mathbf{0}, \lambda_w^{-1}\mathbf{I}) \tag{5.10}$$

where the single potentials $\mathbf{v}$ are set to the target vector $\mathbf{v}^*$ as discussed in section 5.1. The posterior distribution can be approximated with a so called "Laplace Approximation"[95]: by performing a second order Taylor expansion around the mode $\mathbf{w}^*$ of the log posterior it can be written as

$$\begin{aligned} \log p(\mathbf{w}|\mathbf{X}, \mathbf{v}^*) \overset{!}{\approx} \; & \log p(\mathbf{w}^*|\mathbf{X}, \mathbf{v}^*) \\ & + \nabla_{\mathbf{w}} \log p(\mathbf{w}|\mathbf{X}, \mathbf{v}^*)|_{\mathbf{w}^*} (\mathbf{w} - \mathbf{w}^*) \\ & - \frac{1}{2} (\mathbf{w} - \mathbf{w}^*)^{\mathrm{T}} \mathbf{H} (\mathbf{w} - \mathbf{w}^*) \; . \end{aligned} \tag{5.11}$$

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

$$(\mathbf{H})_{klcd,ijab} = - \left. \frac{\partial^2 \log p(\mathbf{w}|\mathbf{X}, \mathbf{v}^*)}{\partial \mathbf{w}_{klcd} \, \partial w_{ijab}} \right|_{(\mathbf{w}^*)} \; . \tag{5.12}$$

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

$$\log p(\mathbf{w}|\mathbf{X}, \mathbf{v}^*) \approx \log p(\mathbf{w}^*|\mathbf{X}, \mathbf{v}^*) - \frac{1}{2} (\mathbf{w} - \mathbf{w}^*)^{\mathrm{T}} \, \mathbf{H} \, (\mathbf{w} - \mathbf{w}^*) \; . \tag{5.13}$$

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

$$
\begin{aligned}
p(\mathbf{w}|\mathbf{X},\mathbf{v}^*) &\approx p(\mathbf{w}^*|\mathbf{X},\mathbf{v}^*)\exp\left(-\frac{1}{2}(\mathbf{w}-\mathbf{w}^*)^{\mathrm{T}}\mathbf{H}(\mathbf{w}-\mathbf{w}^*)\right) \\
&= p(\mathbf{w}^*|\mathbf{X},\mathbf{v}^*)\frac{(2\pi)^{\frac{D}{2}}}{|\mathbf{H}|^{\frac{D}{2}}} \times \mathcal{N}(\mathbf{w}|\mathbf{w}^*,\mathbf{H}^{-1}) \\
&\propto \mathcal{N}(\mathbf{w}|\mathbf{w}^*,\mathbf{H}^{-1})\,, \qquad (5.14)
\end{aligned}
$$

with proportionality constant that depends only on the data and with a precision matrix equal to the negative Hessian matrix. The surprisingly easy computation of the Hessian can be found in Methods section 5.7.6.

5.7.3 Integrating out the Hidden Variables to Obtain the Likelihood Function of the Contact States

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

$$
p(\mathbf{X}|\mathbf{c}) = \int\int p(\mathbf{X}|\mathbf{v},\mathbf{w})\,p(\mathbf{v},\mathbf{w}|\mathbf{c})\,d\mathbf{v}\,d\mathbf{w}\;. \qquad (5.15)
$$

Inserting the prior over parameters $p(\mathbf{v},\mathbf{w}|\mathbf{c})$ from eq. (5.4) into the previous equation and performing the integral over $\mathbf{v}$, as discussed earlier in section 5.1, yields

$$
\begin{aligned}
p(\mathbf{X}|\mathbf{c}) &= \int\left(\int p(\mathbf{X}|\mathbf{v},\mathbf{w})\,\mathcal{N}(\mathbf{v}|\mathbf{v}^*,\lambda_v^{-1}\mathbf{I})\,d\mathbf{v}\right)\prod_{1\le i<j\le L} p(\mathbf{w}_{ij}|c_{ij})\,d\mathbf{w} \qquad (5.16) \\
p(\mathbf{X}|\mathbf{c}) &= \int p(\mathbf{X}|\mathbf{v}^*,\mathbf{w})\prod_{1\le i<j\le L} p(\mathbf{w}_{ij}|c_{ij})\,d\mathbf{w} \qquad (5.17)
\end{aligned}
$$

Next, the likelihood of sequences, $p(\mathbf{X}|\mathbf{v}^*,\mathbf{w})$, will be multiplied with the regularization 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 regularization prior again:

$$
p(\mathbf{X}|\mathbf{c}) = \int p(\mathbf{X}|\mathbf{v}^*,\mathbf{w})\,\mathcal{N}(\mathbf{w}|\mathbf{0},\lambda_w^{-1}\mathbf{I})\prod_{1\le i<j\le L}\frac{p(\mathbf{w}_{ij}|c_{ij})}{\mathcal{N}(\mathbf{w}_{ij}|\mathbf{0},\lambda_w^{-1}\mathbf{I})}\,d\mathbf{w}\,. \qquad (5.18)
$$

Now the crucial advantage of the likelihood regularization is borne out: the strength of the regularization prior, λ_w, can be chosen such that the mode $\mathbf{w}^*$ of the regularized likelihood is near to the mode of the integrand in the last integral. The regularization 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 regularized likelihood with a Gaussian distribution (eq. (5.14)), because this approximation will be fairly accurate in the region around its mode, which is near the region around the mode of the integrand and this again is in the region that contributes most to the integral:

$$
p(\mathbf{X}|\mathbf{c}) \propto \int \mathcal{N}(\mathbf{w}|\mathbf{w}^*,\mathbf{H}^{-1})\prod_{1\le i<j\le L}\frac{p(\mathbf{w}_{ij}|c_{ij})}{\mathcal{N}(\mathbf{w}_{ij}|\mathbf{0},\lambda_w^{-1}\mathbf{I})}d\mathbf{w}\,. \qquad (5.19)
$$

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 5.7.4 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. (5.19), it is therefore a good approximation to simply set the off-diagonal block matrices (case 3 in eq. (5.43)) to zero! The first term in the integrand of eq. (5.19) now factorizes over (i, j),

$$\mathcal{N}(\mathbf{w}|\mathbf{w}^*, \mathbf{H}^{-1}) \approx \prod_{1 \leq i < j \leq L} \mathcal{N}(\mathbf{w}_{ij}|\mathbf{w}^*_{ij}, \mathbf{H}^{-1}_{ij}), \tag{5.20}$$

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,

$$p(\mathbf{X}|\mathbf{c}) \propto \int \prod_{1 \leq i < j \leq L} \mathcal{N}(\mathbf{w}_{ij}|\mathbf{w}^*_{ij}, \mathbf{H}^{-1}_{ij}) \prod_{1 \leq i < j \leq L} \frac{p(\mathbf{w}_{ij}|c_{ij})}{\mathcal{N}(\mathbf{w}_{ij}|\mathbf{0}, \lambda_w^{-1}\mathbf{I})} d\mathbf{w} \tag{5.21}$$

$$p(\mathbf{X}|\mathbf{c}) \propto \int \prod_{1 \leq i < j \leq L} \left(\mathcal{N}(\mathbf{w}_{ij}|\mathbf{w}^*_{ij}, \mathbf{H}^{-1}_{ij}) \frac{p(\mathbf{w}_{ij}|c_{ij})}{\mathcal{N}(\mathbf{w}_{ij}|\mathbf{0}, \lambda_w^{-1}\mathbf{I})} \right) d\mathbf{w} \tag{5.22}$$

$$p(\mathbf{X}|\mathbf{c}) \propto \prod_{1 \leq i < j \leq L} \int \mathcal{N}(\mathbf{w}_{ij}|\mathbf{w}^*_{ij}, \mathbf{H}^{-1}_{ij}) \frac{p(\mathbf{w}_{ij}|c_{ij})}{\mathcal{N}(\mathbf{w}_{ij}|\mathbf{0}, \lambda_w^{-1}\mathbf{I})} d\mathbf{w}_{ij} \tag{5.23}$$

Inserting the coupling prior defined in eq. (5.5) yields

$$p(\mathbf{X}|\mathbf{c}) \propto \prod_{1 \leq i < j \leq L} \int \mathcal{N}(\mathbf{w}_{ij}|\mathbf{w}^*_{ij}, \mathbf{H}^{-1}_{ij}) \frac{\sum_{k=0}^{K} g_k(c_{ij}) \mathcal{N}(\mathbf{w}_{ij}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{w}_{ij}|\mathbf{0}, \lambda_w^{-1}\mathbf{I})} d\mathbf{w}_{ij} \tag{5.24}$$

$$p(\mathbf{X}|\mathbf{c}) \propto \prod_{1 \leq i < j \leq L} \sum_{k=0}^{K} g_k(c_{ij}) \int \frac{\mathcal{N}(\mathbf{w}_{ij}|\mathbf{w}^*_{ij}, \mathbf{H}^{-1}_{ij})}{\mathcal{N}(\mathbf{w}_{ij}|\mathbf{0}, \lambda_w^{-1}\mathbf{I})} \mathcal{N}(\mathbf{w}_{ij}|\mu_k, \mathbf{\Lambda}_k^{-1}) d\mathbf{w}_{ij} \,. \tag{5.25}$$

The integral can be carried out using the following formula:

$$\int d\mathbf{x} \frac{\mathcal{N}(\mathbf{x}|\mu_1, \mathbf{\Lambda}_1^{-1})}{\mathcal{N}(\mathbf{x}|\mathbf{0}, \mathbf{\Lambda}3^{-1})} \mathcal{N}(\mathbf{x}|\mu_2, \mathbf{\Lambda}_2^{-1}) = \frac{\mathcal{N}(\mathbf{0}|\mu_1, \mathbf{\Lambda}_1^{-1}) \mathcal{N}(\mathbf{0}|\mu_2, \mathbf{\Lambda}_2^{-1})}{\mathcal{N}(\mathbf{0}|\mathbf{0}, \mathbf{\Lambda}_3^{-1}) \mathcal{N}(\mathbf{0}|\mu_{12}, \mathbf{\Lambda}_{123}^{-1})} \tag{5.26}$$

with

$$\mathbf{\Lambda}_{123} := \mathbf{\Lambda}_1 - \mathbf{\Lambda}_3 + \mathbf{\Lambda}_2 \tag{5.27}$$

$$\mu_{12} := \mathbf{\Lambda}_{123}^{-1}(\mathbf{\Lambda}_1 \mu_1 + \mathbf{\Lambda}_2 \mu_2). \tag{5.28}$$

We define

$$\mathbf{\Lambda}_{ij,k} := \mathbf{H}_{ij} - \lambda_w \mathbf{I} + \mathbf{\Lambda}_k \tag{5.29}$$

$$\mu_{ij,k} := \mathbf{\Lambda}_{ij,k}^{-1}(\mathbf{H}_{ij} \mathbf{w}^*_{ij} + \mathbf{\Lambda}_k \mu_k) \,. \tag{5.30}$$

and obtain

$$p(\mathbf{X}|\mathbf{c}) \propto \prod_{1\leq i<j\leq L} \sum_{k=0}^{K} g_k(c_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} \,. \tag{5.31}$$

$\mathcal{N}(\mathbf{0}|\mathbf{0}, \lambda_w^{-1}\mathbf{I})$ 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.

5.7.4 The Hessian off-diagonal Elements Carry a Negligible Signal

Assume that $\lambda_w = 0$, i.e., no regularization 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 subtracting 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. (5.43), 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. (5.43)).

But when a realistic regularization 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. (5.43)) to zero. This corresponds to replacing the violet distribution in Figure 5.14 by the pink one. To see why, first note that the functions $g_k(c_{ij})$ and the component distributions $\mathcal{N}(\mathbf{w}_{ij}|\mu_k, \mathbf{\Lambda}_k^{-1})$ 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. (5.19) 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 5.14). For these reasons, the relative strengths of the overlaps with different mixture components labeled by k in eq. (5.5) should be little affected by setting the off-diagonal block matrix couplings to zero.

5.7.5 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 $R(\mu, \mathbf{\Lambda})$:

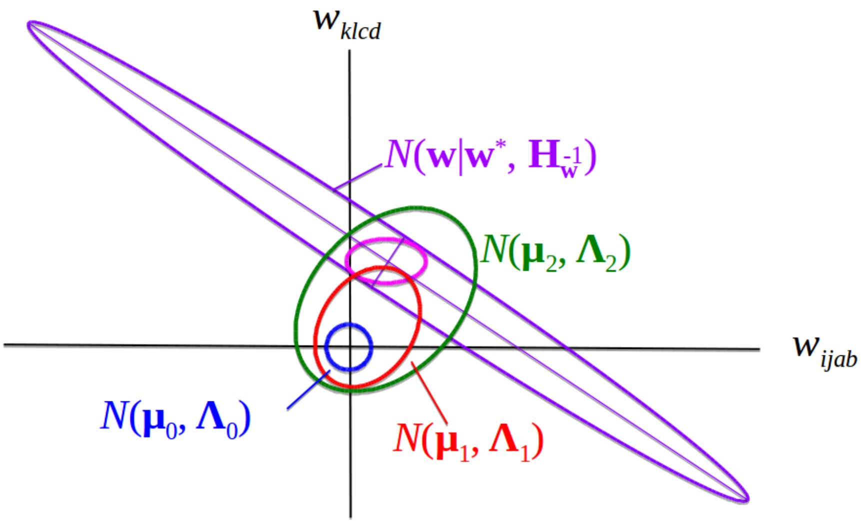

w_{klcd}
$N(\mathbf{w}|\mathbf{w}^*, \mathbf{H}_\mathbf{w}^{-1})$
$N(\boldsymbol{\mu}_2, \boldsymbol{\Lambda}_2)$
w_{ijab}
$N(\boldsymbol{\mu}_0, \boldsymbol{\Lambda}_0)$
$N(\boldsymbol{\mu}_1, \boldsymbol{\Lambda}_1)$

Figure 5.14: Setting the off-diagonal block matrices to zero in $\mathbf{H}$ corresponds to replacing the violet Gaussian distribution by the pink one. The ratios between the overlaps of $\mathcal{N}\left(\mathbf{w}\,|\mathbf{w}^*, \mathbf{H}^{-1}\right)$ with the distributions $\mathcal{N}(\mathbf{w}_{ij}|\mu_k, \mathbf{\Lambda}_k^{-1})$ for various choices of k is only weakly affected by this replacement.

$$LL(\mu, \mathbf{\Lambda}, \gamma) + R(\mu, \mathbf{\Lambda}) = \sum_{n=1}^{M} \log p(\mathbf{X}^m|\mathbf{c}^m, \mu, \mathbf{\Lambda}, \gamma) + R(\mu, \mathbf{\Lambda}) \rightarrow \max\ . \tag{5.32}$$

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

$$R(\mu, \mathbf{\Lambda}) = -\frac{1}{2\sigma_\mu^2}\sum_{k=1}^{K}\sum_{ab=1}^{400} \mu_{k,ab}^2 - \frac{1}{2\sigma_{\text{diag}}^2}\sum_{k=1}^{K}\sum_{ab=1}^{400} \Lambda_{k,ab,ab}^2 \tag{5.33}$$

Reasonable values are $\sigma_\mu = 0.1$, $\sigma_{\text{diag}} = 100$.
These values have been chosen empirically, so that regularization does not substantially impact the strength of hyperparameters but does prevent components with small weights from wandering off zero too far or from becoming too narrow. It has been found that this is necessary especially for mixtures with many components.

The log likelihood can be optimized using L-BFGS-B [240], 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. (5.31) the log likelihood for a single protein is

$$LL(\mu, \mathbf{\Lambda}, \gamma_k) = \sum_{1\leq i<j\leq L} \log \sum_{k=0}^{K} g_k(c_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} + R(\mu, \mathbf{\Lambda}) + \text{const.}\,. \tag{5.34}$$

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`.

The negative log likelihood will be monitored during optimization. Every ten iterations it will be evaluated on a validation set of 1000 residue pairs per contact class to ensure that the model is not overfitting the training data.

5.7.5.1 Dataset Specifications

An equal number of residue pairs that are in physical contact and are not in contact is selected according to the following criteria:

- contact: $\Delta C_\beta < 8\mathring{A}$
- non-contact: $\Delta C_\beta > 25\mathring{A}$ (also evaluated: $\Delta C_\beta > 8\mathring{A}$)
- 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 $(c_{ij}=0) \in \{10000, 100000, 30000, 500000\}$

Proteins from subsets 1-5 of the data set described in method section 2.6.1 have been used for training. Proteins are randomly selected and before residue pairs are selected from a protein, they are shuffled to avoid position bias. For validation of the models, 500 proteins are randomly selected from subsets 6-8 of the data set described in method section 2.6.1. The validation set used to monitor the value of the log likelihood function is generated according to the same criteria and constitutes 1000 residue pairs per contact class.

The MAP estimates of the coupling parameters $\mathbf{w}_{ij}^*$ that are needed to compute the Hessian $\mathbf{H}_{ij}$ as described in method section 5.7.6 are computed by maximizing the pseudo-likelihood and by maximizing the full likelihood with contrastive divergence. Stochastic gradient descent using the tuned hyperparameters presented in chapter 3 will be used to optimize the full likelihood with contrastive divergence. The *ADAM* optimizer is not used because its adaptive learning rates violate the condition $\sum_{a,b}^{20} w_{ijab} = 0$ which is described in section 3.5.1.

5.7.5.2 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. (5.9) 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 reparameterization.

The number of model parameters assembles as follows:

- $(K-1) \times 400$ parameters for μ_k with $k \in \{1, \dots, K\}$ $(\mu_0 = 0)$
- $K \times 400$ parameters for the diagonal $(\mathbf{\Lambda}_k)_{ab,ab}$ with $k \in \{0, \dots, K\}$
- $2 \times (K-1)$ 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 $K=3$ Gaussian components, 3608 parameters for $K=5$ and 7618 parameters for $K=10$ components.

5.7.6 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 $i, j, k, l \in \{1, \dots, L\}$ be columns in the MSA and let $a, b, c, d \in \{1, \dots, 20\}$ represent amino acids. The partial derivative $\partial / \partial \mathbf{w}_{klcd}$ of the second term in the gradient of the couplings in eq. (3.19) is

$$\begin{aligned}\frac{\partial^2 LL_{\text{reg}}(\mathbf{v}^*, \mathbf{w})}{\partial w_{klcd}\, \partial w_{ijab}} =& -\sum_{n=1}^{N} \sum_{\mathbf{y} \in S_n} \frac{\partial \left(\frac{\exp\left(\sum_{i=1}^{L} v_i(y_i) + \sum_{1 \le i < j \le L}^{L} w_{ij}(y_i, y_j)\right)}{Z_n(\mathbf{v}, \mathbf{w})} \right)}{\partial w_{klcd}} I(y_i = a, y_j = b) \\ & -\lambda_w \delta_{ijab,klcd}\,, \end{aligned} \tag{5.35}$$

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

$$\begin{aligned}\frac{\partial^2 LL_{\text{reg}}(\mathbf{v}^*, \mathbf{w})}{\partial w_{klcd}\, \partial w_{ijab}} =& -\sum_{n=1}^{N} \sum_{\mathbf{y} \in S_n} \frac{\exp\left(\sum_{i=1}^{L} v_i(y_i) + \sum_{1 \le i < j \le L}^{L} w_{ij}(y_i, y_j)\right)}{Z_n(\mathbf{v}, \mathbf{w})} I(y_i = a, y_j = b) \\ & \times \left[\frac{\partial}{\partial w_{klcd}} \left(\sum_{i=1}^{L} v_i(y_i) + \sum_{1 \le i < j \le L} w_{ij}(y_i, y_j) \right) - \frac{1}{Z_n(\mathbf{v}, \mathbf{w})} \frac{\partial Z_n(\mathbf{v}, \mathbf{w})}{\partial w_{klcd}} \right] \\ & -\lambda_w \delta_{ijab,klcd} \end{aligned} \tag{5.36}$$

$$\begin{aligned}\frac{\partial^2 LL_{\text{reg}}(\mathbf{v}^*, \mathbf{w})}{\partial w_{klcd}\, \partial w_{ijab}} =& -\sum_{n=1}^{N} \sum_{\mathbf{y} \in S_n} \frac{\exp\left(\sum_{i=1}^{L} v_i(y_i) + \sum_{1 \le i < j \le L}^{L} w_{ij}(y_i, y_j)\right)}{Z_n(\mathbf{v}, \mathbf{w})} I(y_i = a, y_j = b) \\ & \times \left[I(y_k = c, y_l = d) - \frac{\partial}{\partial w_{klcd}} \log Z_n(\mathbf{v}, \mathbf{w}) \right] \\ & -\lambda_w \delta_{ijab,klcd}\,. \end{aligned} \tag{5.37}$$

This expression can be simplified using

$$p(\mathbf{y} | \mathbf{v}, \mathbf{w}) = \frac{\exp\left(\sum_{i=1}^{L} v_i(y_i) + \sum_{1 \le i < j \le L} w_{ij}(y_i, y_j)\right)}{Z_n(\mathbf{v}, \mathbf{w})}, \tag{5.38}$$

yielding

$$\begin{aligned}\frac{\partial^2 LL_{\text{reg}}(\mathbf{v}^*,\mathbf{w})}{\partial w_{klcd}\,\partial w_{ijab}} &= -\sum_{n=1}^{N}\sum_{\mathbf{y}\in S_n} p(\mathbf{y}|\mathbf{v},\mathbf{w})\,I(y_i=a,y_j=b,y_k=c,y_l=d)\\ &\quad+\sum_{n=1}^{N}\sum_{\mathbf{y}\in S_n} p(\mathbf{y}|\mathbf{v},\mathbf{w})\,I(y_i=a,y_j=b)\sum_{\mathbf{y}\in S_n} p(\mathbf{y}|\mathbf{v},\mathbf{w})I(y_k=c,y_l=d)\\ &\quad-\lambda_w\delta_{ijab,klcd}\,. \end{aligned}\tag{5.39}$$

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

$$\begin{aligned}\frac{\partial^2 LL_{\text{reg}}(\mathbf{v}^*,\mathbf{w})}{\partial w_{klcd}\,\partial w_{ijab}} &= -N_{ijkl}\,p(x_i=a,x_j=b,x_k=c,x_l=d|\mathbf{v},\mathbf{w})\\ &\quad+N_{ijkl}\,p(x_i=a,x_j=b|\mathbf{v},\mathbf{w})\,p(x_k=c,x_l=d|\mathbf{v},\mathbf{w})-\lambda_w\delta_{ijab,klcd},\end{aligned}\tag{5.40}$$

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

$$\begin{aligned}\text{case 1}:(\mathbf{H})_{ijab,ijab} &= N_{ij}\,p(x_i=a,x_j=b|\mathbf{v}^*,\mathbf{w}^*)\,(1-p(x_i=a,x_j=b|\mathbf{v}^*,\mathbf{w}^*))\\ &\quad+\lambda_w &(5.41)\\ \text{case 2}:(\mathbf{H})_{ijcd,ijab} &= -N_{ij}\,p(x_i=a,x_j=b|\mathbf{v}^*,\mathbf{w}^*)\,p(x_i=c,x_j=d|\mathbf{v}^*,\mathbf{w}^*) &(5.42)\\ \text{case 3}:(\mathbf{H})_{klcd,ijab} &= N_{ijkl}\,p(x_i=a,x_j=b,x_k=c,x_l=d|\mathbf{v}^*,\mathbf{w}^*)\\ &\quad-N_{ijkl}\,p(x_i=a,x_j=b|\mathbf{v}^*,\mathbf{w}^*)\,p(x_k=c,x_l=d|\mathbf{v}^*,\mathbf{w}^*)\,. &(5.43)\end{aligned}$$

We know from eq. (3.21) that at the mode $\mathbf{w}^*$ the model probabilities match the empirical frequencies up to a small regularization term,

$$p(x_i=a,x_j=b|\mathbf{v}^*,\mathbf{w}^*)=q(x_i=a,x_j=b)-\frac{\lambda_w}{N_{ij}}w^*_{ijab}\,,\tag{5.44}$$

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

$$\begin{aligned}(\mathbf{H})_{ijab,ijab} &= N_{ij}\left(q(x_i=a,x_j=b)-\frac{\lambda_w}{N_{ij}}w^*_{ijab}\right)\left(1-q(x_i=a,x_j=b)+\frac{\lambda_w}{N_{ij}}w^*_{ijab}\right)\\ &\quad+\lambda_w &(5.45)\\ (\mathbf{H})_{ijcd,ijab} &= -N_{ij}\left(q(x_i=a,x_j=b)-\frac{\lambda_w}{N_{ij}}w^*_{ijab}\right)\left(q(x_i=c,x_j=d)-\frac{\lambda_w}{N_{ij}}w^*_{ijcd}\right) &(5.46)\end{aligned}$$

In order to write the previous eq. (5.46) in matrix form, the *regularised* empirical frequencies $\mathbf{q'}_{ij}$ will be defined as

$$(\mathbf{q\prime}_{ij})_{ab} = q'_{ijab} := q(x_i\!=\!a, x_j\!=\!b) - \lambda_w w^*_{ijab}/N_{ij}\,, \tag{5.47}$$

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

$$\mathbf{Q}_{ij} := \mathrm{diag}(\mathbf{q\prime}_{ij})\,. \tag{5.48}$$

Now eq. (5.46) can be written in matrix form

$$\mathbf{H}_{ij} = N_{ij}\left(\mathbf{Q}_{ij} - \mathbf{q\prime}_{ij}\mathbf{q\prime}^{\mathrm{T}}_{ij}\right) + \lambda_w \mathbf{I}\,. \tag{5.49}$$

5.7.7 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 5.7.3 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 (5.49) can be used to write $\mathbf{\Lambda}_{ij,k}$ in matrix form as

$$\mathbf{\Lambda}_{ij,k} = \mathbf{H}_{ij} - \lambda_w \mathbf{I} + \mathbf{\Lambda}_k = N_{ij}\mathbf{Q}_{ij} - N_{ij}\mathbf{q\prime}_{ij}\mathbf{q\prime}^{\mathrm{T}}_{ij} + \mathbf{\Lambda}_k\,. \tag{5.50}$$

Owing to eqs. (3.15) and (3.23), $\sum_{a,b=1}^{20} q'_{ijab} = 1$. The previous equation (5.50) facilitates the calculation of the inverse of this matrix using the *Woodbury identity* for matrices

$$(\mathbf{A} + \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{D} + \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1}\,. \tag{5.51}$$

by setting

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

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

$$\begin{aligned}
(\mathbf{H}_{ij} - \lambda_w \mathbf{I} + \mathbf{\Lambda}_k)^{-1} &= \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{q\prime}_{ij}\left(-N_{ij}^{-1} + \mathbf{q\prime}^{\mathrm{T}}_{ij}\mathbf{A}^{-1}\mathbf{q\prime}_{ij}\right)^{-1}\mathbf{q\prime}^{\mathrm{T}}_{ij}\mathbf{A}^{-1} \\
&= \mathbf{A}^{-1} + \frac{(\mathbf{A}^{-1}\mathbf{q\prime}_{ij})(\mathbf{A}^{-1}\mathbf{q\prime}_{ij})^{\mathrm{T}}}{N_{ij}^{-1} - \mathbf{q\prime}^{\mathrm{T}}_{ij}\mathbf{A}^{-1}\mathbf{q\prime}_{ij}}\,.
\end{aligned} \tag{5.52}$$

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

$$N_{ij}^{-1} - \sum_{a,b=1}^{20} \frac{q'^{2}_{ijab}}{N_{ij} q'_{ijab} + (\mathbf{\Lambda}_k)_{ab,ab}} > N_{ij}^{-1} - \sum_{a,b=1}^{20} \frac{q'^{2}_{ijab}}{N_{ij} q'_{ijab}} = 0 \tag{5.53}$$

and therefore the inverse of $\mathbf{\Lambda}_{ij,k}$ in eq. (5.52) is well defined. The log determinant of $\mathbf{\Lambda}_{ij,k}$ is necessary to compute the ratio of Gaussians (see equation (5.31)) and can be computed using the matrix determinant lemma:

$$\det(\mathbf{A} + \mathbf{u}\mathbf{v}^{\mathrm{T}}) = (1 + \mathbf{v}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{u}) \det(\mathbf{A}) \tag{5.54}$$

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

$$\det(\mathbf{\Lambda}_{ij,k}) = \det(\mathbf{H}_{ij} - \lambda_w \mathbf{I} + \mathbf{\Lambda}_k) = (1 - N_{ij}\mathbf{q'}^{\mathrm{T}}_{ij}\mathbf{A}^{-1}\mathbf{q'}_{ij}) \det(\mathbf{A})\,. \tag{5.55}$$

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

5.7.8 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. (5.34) (neglecting the regularization term) with respect to $\mu_{k,ab}$, one obtains

$$\frac{\partial}{\partial \mu_{k,ab}} LL(\mu, \mathbf{\Lambda}, \gamma_k) = \sum_{1 \le i < j \le L} \frac{g_k(c_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} \frac{\partial}{\partial \mu_{k,ab}} \log\left(\frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} \right)}{\sum_{k'=0}^{K} g_{k'}(c_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu'_k, \mathbf{\Lambda}'^{-1}_k)}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})}}. \tag{5.56}$$

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:

$$p(k|ij) = \frac{g_k(c_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})}}{\sum_{k'=0}^{K} g_{k'}(c_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu'_k, \mathbf{\Lambda}'^{-1}_k)}{\mathcal{N}(\mathbf{0}|\mu'_{ij,k}, \mathbf{\Lambda}'^{-1}_{ij,k})}}\,. \tag{5.57}$$

By substituting the definition for responsibility, (5.56) simplifies

$$\frac{\partial}{\partial \mu_{k,ab}} LL(\mu, \mathbf{\Lambda}, \gamma_k) = \sum_{1 \le i < j \le L} p(k|ij) \frac{\partial}{\partial \mu_{k,ab}} \log\left(\frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} \right), \tag{5.58}$$

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

$$\frac{\partial}{\partial \mu_{k,ab}} \log\left(\frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} \right) = \frac{1}{2} \frac{\partial}{\partial \mu_{k,ab}} \left(\log|\mathbf{\Lambda}_k| - \mu_k^{\mathrm{T}} \mathbf{\Lambda}_k \mu_k - \log|\mathbf{\Lambda}_{ij,k}| + \mu_{ij,k}^{\mathrm{T}} \mathbf{\Lambda}_{ij,k} \mu_{ij,k} \right)\,. \tag{5.59}$$

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

$$\frac{\partial}{\partial x}\left(\mathbf{y}^{\mathrm{T}}\mathbf{A}\mathbf{y}\right) = \frac{\partial \mathbf{y}^{\mathrm{T}}}{\partial x}\mathbf{A}\mathbf{y} + \mathbf{y}^{\mathrm{T}}\mathbf{A}\frac{\partial \mathbf{y}}{\partial x} = \mathbf{y}^{\mathrm{T}}(\mathbf{A}+\mathbf{A}^{\mathrm{T}})\frac{\partial \mathbf{y}}{\partial x} \tag{5.60}$$

the partial derivative therefore becomes

$$\begin{aligned}\frac{\partial}{\partial \mu_{k,ab}} \log\left(\frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})}\right) &= \left(-\mu_k^{\mathrm{T}}\mathbf{\Lambda}_k\mathbf{e}_{ab} \, + \mu_{ij,k}^{\mathrm{T}}\mathbf{\Lambda}_{ij,k}\mathbf{\Lambda}_{ij,k}^{-1}\mathbf{\Lambda}_k\mathbf{e}_{ab}\right) \\ &= \mathbf{e}_{ab}^{\mathrm{T}}\mathbf{\Lambda}_k(\mu_{ij,k} - \mu_k) \; . \end{aligned} \tag{5.61}$$

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

$$\nabla_{\mu_k} LL(\mu, \mathbf{\Lambda}, \gamma_k) = \sum_{1 \le i < j \le L} p(k|ij) \, \mathbf{\Lambda}_k \left(\mu_{ij,k} - \mu_k\right) \; . \tag{5.62}$$

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

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

Analogously to eq. (5.58) one first needs to solve

$$\begin{aligned}&\frac{\partial}{\partial \Lambda_{k,ab,cd}} \log \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} \\ &= \frac{1}{2}\frac{\partial}{\partial \Lambda_{k,ab,cd}} \left(\log|\mathbf{\Lambda}_k| - \mu_k^{\mathrm{T}}\mathbf{\Lambda}_k\mu_k - \log|\mathbf{\Lambda}_{ij,k}| + \mu_{ij,k}^{\mathrm{T}}\mathbf{\Lambda}_{ij,k}\mu_{ij,k}\right) \; , \end{aligned} \tag{5.63}$$

by applying eq. (5.60) as before as well as the formulas

$$\begin{aligned}\frac{\partial}{\partial x}\log|\mathbf{A}| &= \mathrm{Tr}\left(\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial x}\right) , \\ \frac{\partial \mathbf{A}^{-1}}{\partial x} &= -\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial x}\mathbf{A}^{-1} \, . \end{aligned} \tag{5.64}$$

This yields

$$\frac{\partial}{\partial \Lambda_{k,ab,cd}} \log |\mathbf{\Lambda}_k| = \mathrm{Tr}\left(\mathbf{\Lambda}_k^{-1} \frac{\partial \mathbf{\Lambda}_k}{\partial \Lambda_{k,ab,cd}}\right) = \mathrm{Tr}\left(\mathbf{\Lambda}_k^{-1}\mathbf{e}_{ab}\mathbf{e}_{cd}^{\mathrm{T}}\right) = \Lambda^{-1}_{k,cd,ab} \tag{5.65}$$

$$\frac{\partial}{\partial \Lambda_{k,ab,cd}} \log |\mathbf{\Lambda}_{ij,k}| = \mathrm{Tr}\left(\mathbf{\Lambda}_{ij,k}^{-1} \frac{\partial (\mathbf{H}_{ij} - \lambda_w \mathbf{I} + \mathbf{\Lambda}_k)}{\partial \Lambda_{k,ab,cd}}\right) = \Lambda^{-1}_{ij,k,cd,ab} \tag{5.66}$$

$$\frac{\partial (\mu_k^{\mathrm{T}} \mathbf{\Lambda}_k \mu_k)}{\partial \Lambda_{k,ab,cd}} = \mu_k^{\mathrm{T}} \mathbf{e}_{ab}\mathbf{e}_{cd}^{\mathrm{T}} \mu_k = \mathbf{e}_{ab}^{\mathrm{T}} \mu_k \mu_k^{\mathrm{T}} \mathbf{e}_{cd} = (\mu_k \mu_k^{\mathrm{T}})_{ab,cd} \tag{5.67}$$

$$\begin{aligned}
\frac{\partial (\mu_{ij,k}^{\mathrm{T}} \mathbf{\Lambda}_{ij,k} \mu_{ij,k})}{\partial \Lambda_{k,ab,cd}} =& \mu_{ij,k}^{\mathrm{T}} \frac{\partial \mathbf{\Lambda}_{ij,k}}{\partial \Lambda_{k,ab,cd}} \mu_{ij,k} + 2\mu_{ij,k}^{\mathrm{T}} \mathbf{\Lambda}_{ij,k} \frac{\partial \mathbf{\Lambda}_{ij,k}^{-1}}{\partial \Lambda_{k,ab,cd}} (\mathbf{H}_{ij}\mathbf{w}_{ij}^{*} + \mathbf{\Lambda}_k \mu_k) \\
& + 2\mu_{ij,k}^{\mathrm{T}} \frac{\partial \mathbf{\Lambda}_k}{\partial \Lambda_{k,ab,cd}} \mu_k \\
=& (\mu_{ij,k}\mu_{ij,k}^{\mathrm{T}} + 2\mu_{ij,k}\mu_k^{\mathrm{T}})_{ab,cd} \\
& - 2\mu_{ij,k}^{\mathrm{T}} \mathbf{\Lambda}_{ij,k} \mathbf{\Lambda}_{ij,k}^{-1} \frac{\partial \mathbf{\Lambda}_{ij,k}}{\partial \Lambda_{k,ab,cd}} \mathbf{\Lambda}_{ij,k}^{-1} (\mathbf{H}_{ij}\mathbf{w}_{ij}^{*} + \mathbf{\Lambda}_k \mu_k) \\
=& (\mu_{ij,k}\mu_{ij,k}^{\mathrm{T}} + 2\mu_{ij,k}\mu_k^{\mathrm{T}})_{ab,cd} - 2\mu_{ij,k}^{\mathrm{T}} \frac{\partial \mathbf{\Lambda}_{ij,k}}{\partial \Lambda_{k,ab,cd}} \mu_{ij,k} \\
=& (-\mu_{ij,k}\mu_{ij,k}^{\mathrm{T}} + 2\mu_{ij,k}\mu_k^{\mathrm{T}})_{ab,cd} \, .
\end{aligned} \tag{5.68}$$

Inserting these results into eq. (5.63) yields

$$\frac{\partial}{\partial \Lambda_{k,ab,cd}} \log \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} = \frac{1}{2} \left(\mathbf{\Lambda}_k^{-1} - \mathbf{\Lambda}_{ij,k}^{-1} - (\mu_{ij,k} - \mu_k)(\mu_{ij,k} - \mu_k)^{\mathrm{T}} \right)_{ab,cd} . \tag{5.69}$$

Substituting this expression into the equation (5.58) analogous to the derivation of gradient for $\mu_{k,ab}$ yields the equation

$$\nabla_{\mathbf{\Lambda}_k} LL(\mu, \mathbf{\Lambda}, \gamma_k) = \frac{1}{2} \sum_{1 \le i < j \le L} p(k|ij) \left(\mathbf{\Lambda}_k^{-1} - \mathbf{\Lambda}_{ij,k}^{-1} - (\mu_{ij,k} - \mu_k)(\mu_{ij,k} - \mu_k)^{\mathrm{T}} \right) . \tag{5.70}$$

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

5.7.10 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. (5.9). The derivative of the mixing weights $g_k(c_{ij})$ is:

$$\frac{\partial g_{k'}(c_{ij})}{\partial \gamma_k} = \begin{cases} g_k(c_{ij})(1 - g_k(c_{ij})) & : k' = k \\ g_{k'}(c_{ij}) - g_k(c_{ij}) & : k' \neq k \end{cases} \tag{5.71}$$

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

$$
\begin{aligned}
\frac{\partial}{\partial \gamma_k} LL(\mu, \mathbf{\Lambda}, \gamma_k) &= \sum_{1 \le i < j \le L} \frac{\sum_{k'=0}^{K} \frac{\partial}{\partial \gamma_k} g_{k'}(c_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(0|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})}}{\sum_{k'=0}^{K} g_{k'}(c_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})}} \\
&= \sum_{1 \le i < j \le L} \frac{\sum_{k'=0}^{K} g_{k'}(c_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})} \cdot \begin{cases} 1 - g_k(c_{ij}) & \text{if } k' = k \\ -g_k(c_{ij}) & \text{if } k' \neq k \end{cases}}{\sum_{k'=0}^{K} g_{k'}(c_{ij}) \frac{\mathcal{N}(\mathbf{0}|\mu_k, \mathbf{\Lambda}_k^{-1})}{\mathcal{N}(\mathbf{0}|\mu_{ij,k}, \mathbf{\Lambda}_{ij,k}^{-1})}} \\
&= \sum_{1 \le i < j \le L} \sum_{k'=0}^{K} p(k'|ij) \begin{cases} 1 - g_k(c_{ij}) & \text{if } k' = k \\ -g_k(c_{ij}) & \text{if } k' \neq k \end{cases} \\
&= \sum_{1 \le i < j \le L} p(k|ij) - g_k(c_{ij}) \sum_{k'=0}^{K} p(k'|ij) \\
&= \sum_{1 \le i < j \le L} p(k|ij) - g_k(c_{ij}) \qquad (5.72)
\end{aligned}
$$

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

It is straightforward to extend the Bayesian model for contact prediction presented in section 5.1 for distances. The prior over couplings will modelled using distance dependent mixture weights $g_k(c_{ij})$. Therefore eq. (5.5) is modified such that mixture weights $g_k(c_{ij})$ are modelled as softmax over linear functions $\gamma_k(c_{ij})$ (see Figure 5.15:

$$
g_k(c_{ij}) = \frac{\exp \gamma_k(c_{ij})}{\sum_{k'=0}^{K} \exp \gamma_{k'}(c_{ij})} , \qquad (5.73)
$$

$$
\gamma_k(c_{ij}) = - \sum_{k'=0}^{k} \alpha_{k'} (c_{ij} - \rho_{k'}). \qquad (5.74)
$$

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 $p(\mathbf{w}|\mathbf{X}) \to \mathcal{N}(0, \sigma_0^2 \mathbf{I})$.

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 behavior 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 optimization of these parameters more efficient.

5.7.11.1 The derivative of the log likelihood with respect to ρ_k

Analogous to the derivations of μ_k in section 5.7.8 and $\mathbf{\Lambda}_k$ in section 5.7.9, the partial derivative with respect to ρ_k is

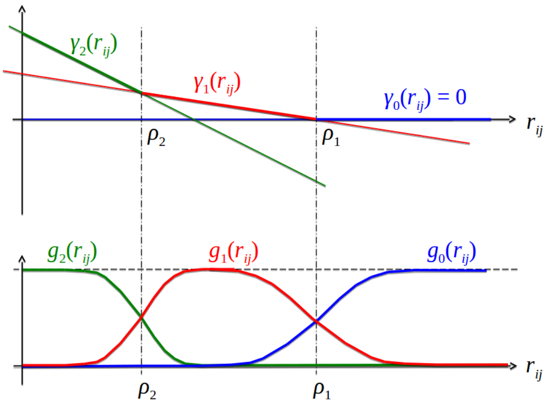


Figure 5.15: The Gaussian mixture coefficients $g_k(c_{ij})$ of $p(\mathbf{w}_{ij}|c_{ij})$ are modelled as softmax over linear functions $\gamma_k(c_{ij})$. ρ_k sets the transition point between neighbouring components $g_{k-1}(c_{ij})$ and $g_k(c_{ij})$, while α_k quantifies the abruptness of the transition between $g_{k-1}(c_{ij})$ and $g_k(c_{ij})$.

$$\frac{\partial}{\partial \rho_k} LL(\mu, \mathbf{\Lambda}, \rho, \alpha) = \sum_{1 \leq i < j \leq L} \sum_{k'=0}^{K} p(k'|ij) \frac{\partial}{\partial \rho_k} \log g_{k'}(c_{ij}) \,. \tag{5.75}$$

Using the definition of $g_k(c_{ij})$ in eq. (5.74), we find (remember that $\alpha_0 = 0$ as noted in the last section) that

$$\begin{aligned}
\frac{\partial}{\partial \rho_k} \log g_l(c_{ij}) &= \frac{\partial}{\partial \rho_k} \log \frac{\exp\left(-\sum_{k''=1}^{k'} \alpha_{k''}(c_{ij} - \rho_{k''})\right)}{\sum_{k'=0}^{K} \exp\left(-\sum_{k''=1}^{k'} \alpha_{k''}(c_{ij} - \rho_{k''})\right)} \\
&= -\frac{\partial}{\partial \rho_k} \sum_{k''=1}^{l} \alpha_{k''}(c_{ij} - \rho_{k''}) - \frac{\partial}{\partial \rho_k} \log \sum_{k'=0}^{K} \exp\left(-\sum_{k''=1}^{k'} \alpha_{k''}(c_{ij} - \rho_{k''})\right) \\
&= \alpha_k \, I(l \geq k) - \frac{\sum_{k'=0}^{K} \frac{\partial}{\partial \rho_k} \exp(-\sum_{k''=1}^{k'} \alpha_{k''}(c_{ij} - \rho_{k''}))}{\sum_{k'=0}^{K} \exp(-\sum_{k''=1}^{k'} \alpha_{k''}(c_{ij} - \rho_{k''}))} \\
&= \alpha_k \, I(l \geq k) - \frac{\sum_{k'=0}^{K} \exp(-\sum_{k''=1}^{k'} \alpha_{k''}(c_{ij} - \rho_{k''})) \, \alpha_k \, I(k' \geq k)}{\sum_{k'=0}^{K} \exp(-\sum_{k''=1}^{k'} \alpha_{k''}(c_{ij} - \rho_{k''}))} \\
&= \alpha_k \, I(l \geq k) - \frac{\sum_{k'=0}^{K} \exp(\gamma_{k'}(c_{ij})) \, \alpha_k \, I(k' \geq k)}{\sum_{k'=0}^{K} \exp(\gamma_{k'}(c_{ij}))} \\
&= \alpha_k \, I(l \geq k) - \sum_{k'=0}^{K} g_{k'}(c_{ij}) \, \alpha_k \, I(k' \geq k) \\
&= \alpha_k \left(I(l \geq k) - \sum_{k'=k}^{K} g_{k'}(c_{ij}) \right) .
\end{aligned} \tag{5.76}$$

Inserting this into eq. (5.75) yields

$$\frac{\partial}{\partial \rho_k} LL(\mu, \mathbf{\Lambda}, \rho, \alpha) = \sum_{1\le i<j\le L} \sum_{k'=0}^{K} p(k'|ij)\, \alpha_k \left(I(k' \ge k) - \sum_{k''=k}^{K} g_{k''}(c_{ij}) \right)$$
$$= \alpha_k \sum_{1\le i<j\le L} \left(\sum_{k'=k}^{K} p(k'|ij) - \sum_{k'=0}^{K} p(k'|ij) \sum_{k''=k}^{K} g_{k''}(c_{ij}) \right), \qquad (5.77)$$

and finally

$$\frac{\partial}{\partial \rho_k} LL(\mu, \mathbf{\Lambda}, \rho, \alpha) = \alpha_k \sum_{1\le i<j\le L} \sum_{k'=k}^{K} (p(k'|ij) - g_{k'}(c_{ij}))\,. \qquad (5.78)$$

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

5.7.11.2 The derivative of the log likelihood with respect to α_k

The partial derivative with respect to α_k is obtained similarly to the previous derivation,

$$\frac{\partial}{\partial \alpha_k} LL(\mu, \mathbf{\Lambda}, \rho, \alpha) = \sum_{1\le i<j\le L} \sum_{k'=0}^{K} p(k'|ij)\, \frac{\partial}{\partial \alpha_k} \log g_{k'}(c_{ij})\,. \qquad (5.79)$$

Similarly as before,

$$\frac{\partial}{\partial \alpha_k} \log g_l(c_{ij}) = \frac{\partial}{\partial \alpha_k} \log \frac{\exp(-\sum_{k''=1}^{l} \alpha_{k''}(c_{ij} - \rho_{k''})}{\sum_{k'=0}^{K} \exp(-\sum_{k''=1}^{k'} \alpha_{k''}(c_{ij} - \rho_{k''}))}$$
$$= -\frac{\partial}{\partial \alpha_k} \sum_{k''=1}^{l} \alpha_{k''}(c_{ij} - \rho_{k''}) - \frac{\partial}{\partial \alpha_k} \log \sum_{k'=0}^{K} \exp\left(- \sum_{k''=1}^{k'} \alpha_{k''}(c_{ij} - \rho_{k''}) \right)$$
$$= -(c_{ij} - \rho_k)\, I(l \ge k) - \frac{\sum_{k'=0}^{K} \frac{\partial}{\partial \alpha_k} \exp(-\sum_{k''=1}^{k'} \alpha_{k''}(c_{ij} - \rho_{k''}))}{\sum_{k'=0}^{K} \exp(-\sum_{k''=1}^{k'} \alpha_{k''}(c_{ij} - \rho_{k''}))}$$
$$= -(c_{ij} - \rho_k) \left(I(l \ge k) - \sum_{k''=k}^{K} g_{k''}(c_{ij})) \right). \qquad (5.80)$$

Inserting this into eq. (5.79) yields

$$\frac{\partial}{\partial \alpha_k} LL(\mu, \mathbf{\Lambda}, \rho, \alpha) = - \sum_{1\le i<j\le L} \sum_{k'=0}^{K} p(k'|ij)\, (c_{ij} - \rho_k) \left(I(k' \ge k) - \sum_{k''=k}^{K} g_{k''}(c_{ij})) \right)$$
$$= - \sum_{1\le i<j\le L} (c_{ij} - \rho_k) \left(\sum_{k'=k}^{K} p(k'|ij) - \sum_{k'=0}^{K} p(k'|ij) \sum_{k''=k}^{K} g_{k''}(c_{ij})) \right), \qquad (5.81)$$

and finally

$$\frac{\partial}{\partial \alpha_k} LL(\mu, \mathbf{\Lambda}, \rho, \alpha) = \sum_{1 \le i < j \le L} (\rho_k - c_{ij}) \sum_{k'=k}^{K} (p(k'|ij) - g_{k'}(c_{ij}))\,. \tag{5.82}$$

6 Conclusion and Outlook

With the work presented here, I was the first to formulate DCA as a principled statistical approach, providing true probability estimates and promoting biological insights. The transparency of the modelling process and the flexibility of the Bayesian framework lay the foundation to further improvements of DCA for small protein families in a mechanistic way.

In chapter 2, I presented a thorough analysis of coupling matrices that are inferred from a multiple sequence alignment (MSA) and reflect the tendencies of amino acids to co-occur at paired positions in the MSA. I showed that coupling matrices contain valuable information with a meaningful biological interpretation. For example, the distributions of coupling values reflect biophysical interaction preferences between amino acids and the signal weakens with increasing residue-residue distances. Furthermore, interdependencies between different couplings are coherent and induce characteristic patterns in coupling matrices, often indicating the structural constraint for the residue pair. The majority of this information is lost by the the way current methods apply heuristics to compute a prediction for a residue-residue contact. However, in my Bayesian framework presented in chapter 5, this information is explicitly modelled.

Chapter 3 presented an alternative approach to infer the *Potts* model parameters. Due to the complexity of the normalization constant it is infeasible to maximize the full likelihood to derive the model parameters. The most popular DCA approaches optimize the pseudo-likelihood instead but it is unknown how well the pseudo-likelihood solution approximates the full likelihood solution in case protein families have only few members. In my work, I optimized the full likelihood by using an approximate gradient provided by an algorithm called *contrastive divergence*, which is a novel method in contact prediction. I systematically tuned the stochastic gradient descent algorithm for the use with *contrastive divergence* and also examined various modifications to the estimation of the gradient. My approach achieved comparable precision as pseudo-likelihood methods with minor improvements for small protein families, which could be traced back to amplified signals between strongly conserved residue pairs.

A random forest classifier for contact prediction which was trained on sequence features is discussed in chapter 4. This model yields a robust estimator that outperforms coevolution methods for small protein families where they suffer from the low signal-to-noise ratio. In line with the most successful contact predictors, which exploit information from various sources and multiple DCA methods, I integrated the predictions of the pseudo-likelihood and the constrastive divergence method as additional features for training. The individual methods greatly contribute and improve the predictive performance of the random forest classifier.

The Bayesian framework proposed in chapter 5 represents a principled statistical approach that eradicates the use of heuristics by explicitly modelling the full information contained in the coupling signatures while at the same time accounting for the uncertainty of the data. Based on the observations and biological interpretations of coupling signals in chapter 2, the prior on couplings was modelled as a Gaussian mixture. The hyperparameters were trained on inferred couplings and structures from many proteins and the Gaussian mixture model convincingly reproduced empirical coupling distributions. Posterior probability estimates of residue-residue contacts are obtained by combining the likelihood of contacts with prior information in form of the random forest contact class probabilities. They posterior probabilities are less precise than the heuristic predictions obtained from the pseudo-likelihood approach combined with prior information. A possible explanation is that the Gaussian mixture model of the coupling prior does not yet capture enough information in order to efficiently discriminate between contacts and non-contacts. Even though reproducing the one- and two-dimensional empirical distributions, it is plausible that the precise interdependencies between couplings require a more complex model, e.g. by using more Gaussian components or full instead of diagonal covariance matrices. Furthermore, the approximation to the regularized likelihood of the sequences with a multivariate Gaussian can and perhaps must be iteratively improved by another round of training employing an improved regularization prior. Finally, the reason could be that certain inherent modelling assumptions are not met or are too inaccurate but work to verify these assumptions is still ongoing.

Especially with the limited knowledge and uncertainty in the data, the Bayesian statistical approach developed here provides a solid theoretical and statistically sound formulation for DCA with enhanced explanatory power compared to the uninformative heuristics. Through the formulation in the language of Bayesian statistics, the framework naturally allows the integration of additional prior knowledge and it facilitates its further usage in even more complex Bayesian hierarchies. It is also straightforward to extend the model towards the estimation of posterior probabilities of residue-residue distances (see section 5.7.11). The analysis in chapter 2 has demonstrated that the coupling signal weakens with increasing inter-residue distances which represents additional information that awaits full utilization. The information gain of residue-residue distance estimates over binary contact prediction is substantial and is a promising way to greatly improve *de novo* structure prediction [241].

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

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 [242]

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	Asparagi**N**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

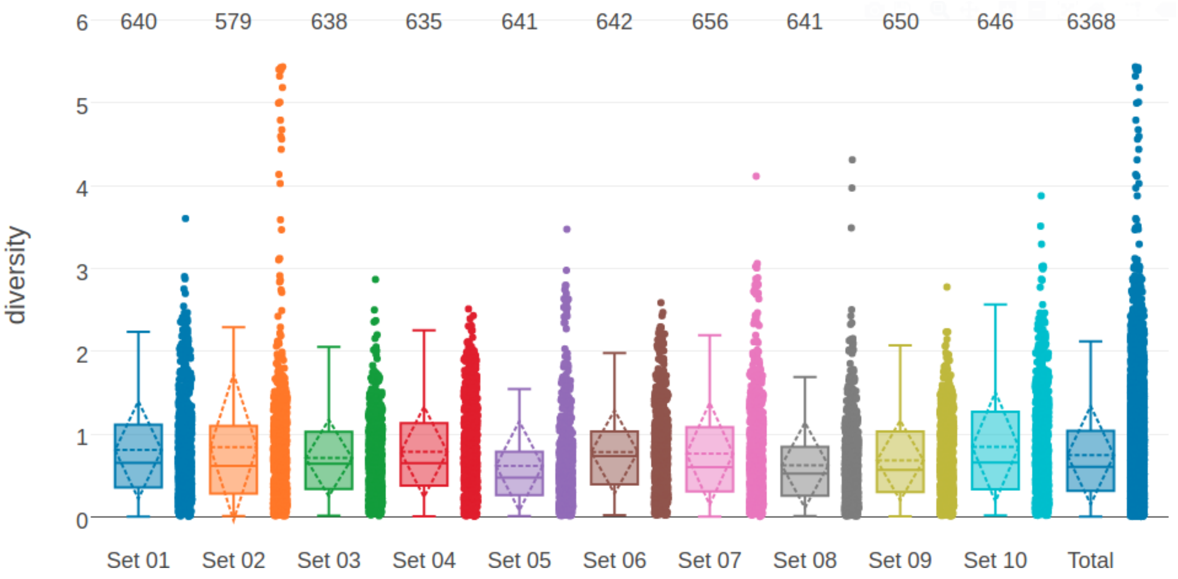


Figure C.1: Distribution of alignment diversity (= $\sqrt{(\frac{N}{L})}$) in the dataset and its ten subsets.

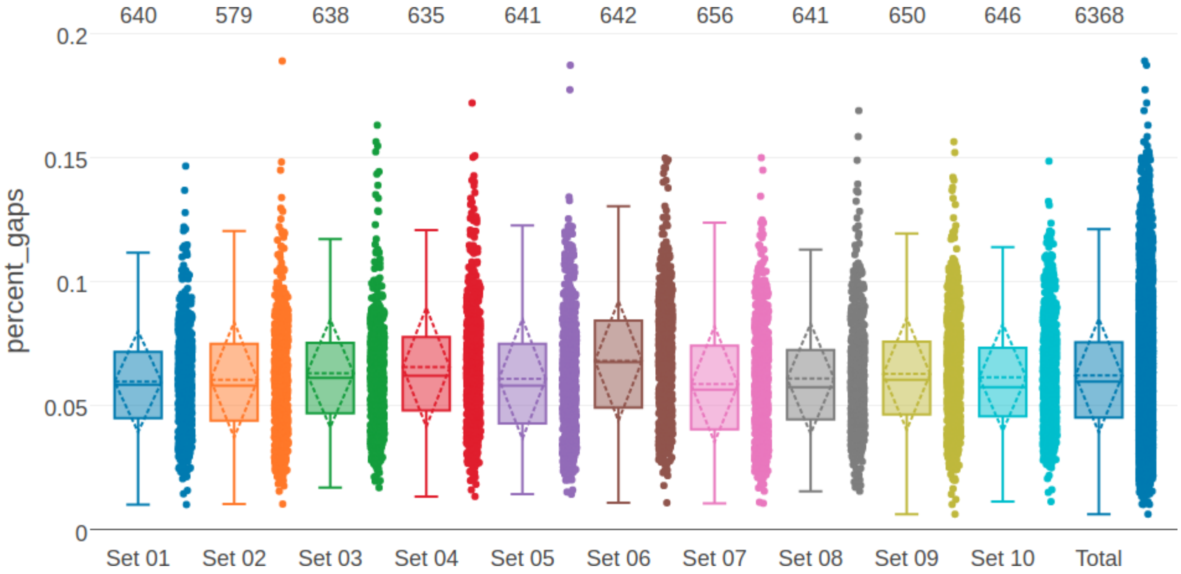


Figure C.2: Distribution of gap percentage of alignments in the dataset and its ten subsets.

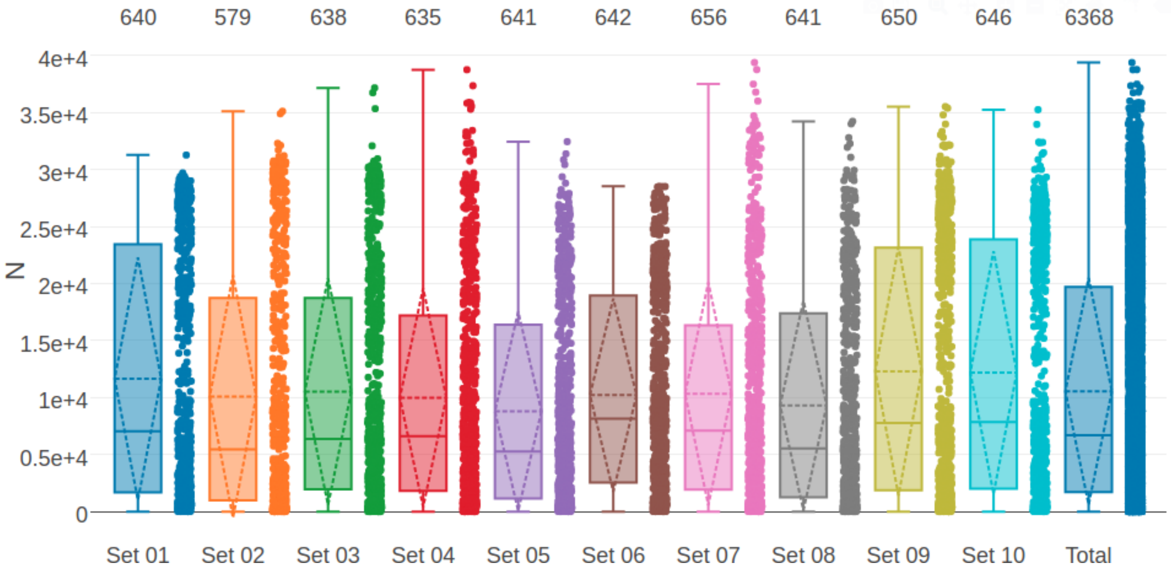


Figure C.3: Distribution of alignment size (number of sequences N) in the dataset and its ten subsets.

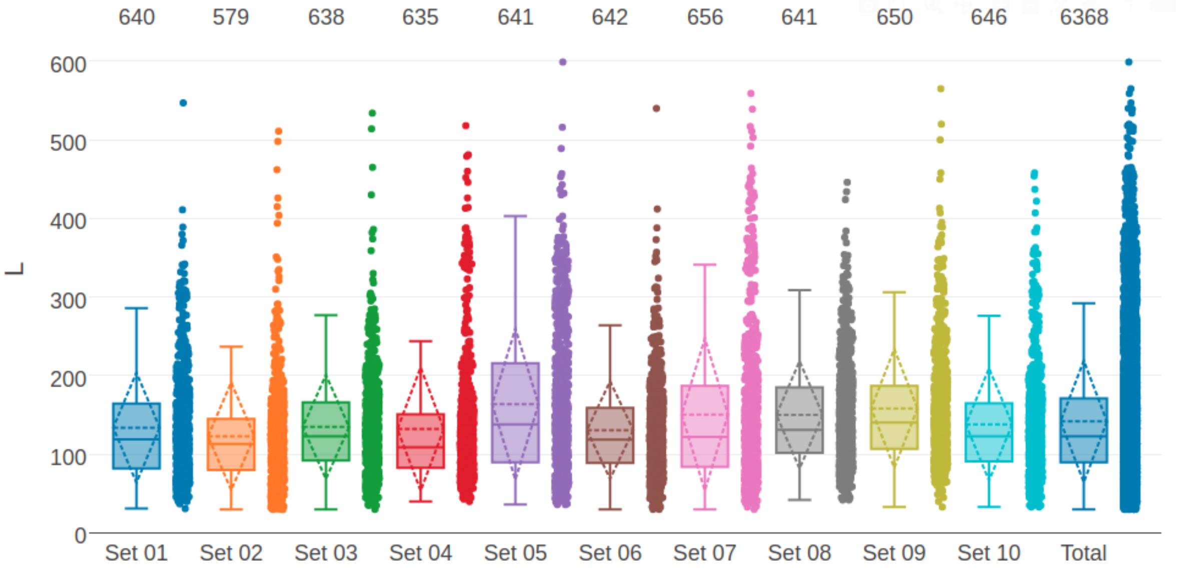


Figure C.4: Distribution of protein length L in the dataset and its ten subsets.

D

Interpretation of Coupling Matrices

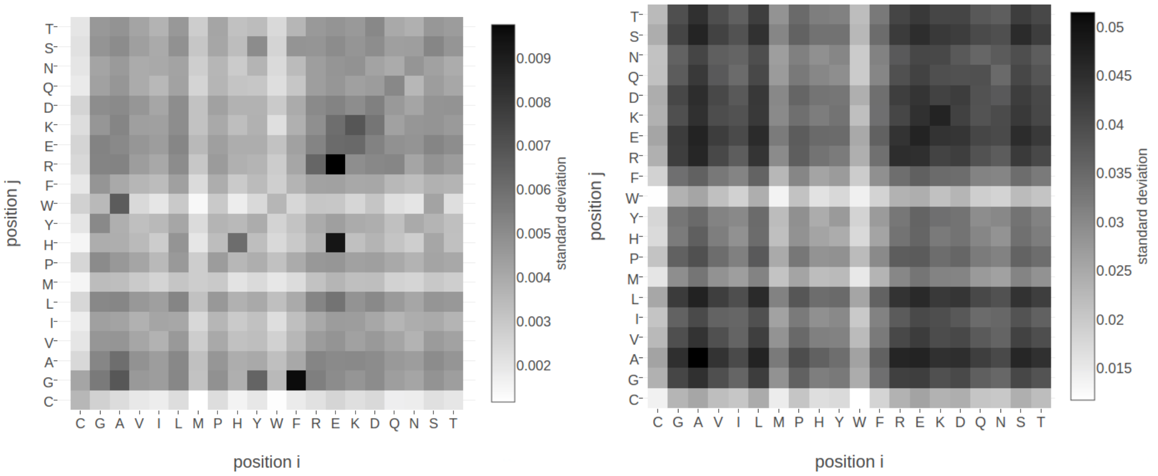


Figure D.1: Standard deviation of squared coupling values ${w_{ijab}}^2$ and of coupling values w_{ijab} for residue pairs not in physical contact ($\Delta C_\beta > 25\mathring{A}$). Dataset contains 100.000 residue pairs per class (for details see methods section 2.6.6). Amino acids are abbreviated with one-letter code and they are broadly grouped with respect to physico-chemical properties listed in Appendix B **Left** Standard deviation of squared coupling values ${w_{ijab}}^2$. **Right** Standard deviation of coupling values w_{ijab}.

E

Optimizing Full Likelihood with Gradient Descent

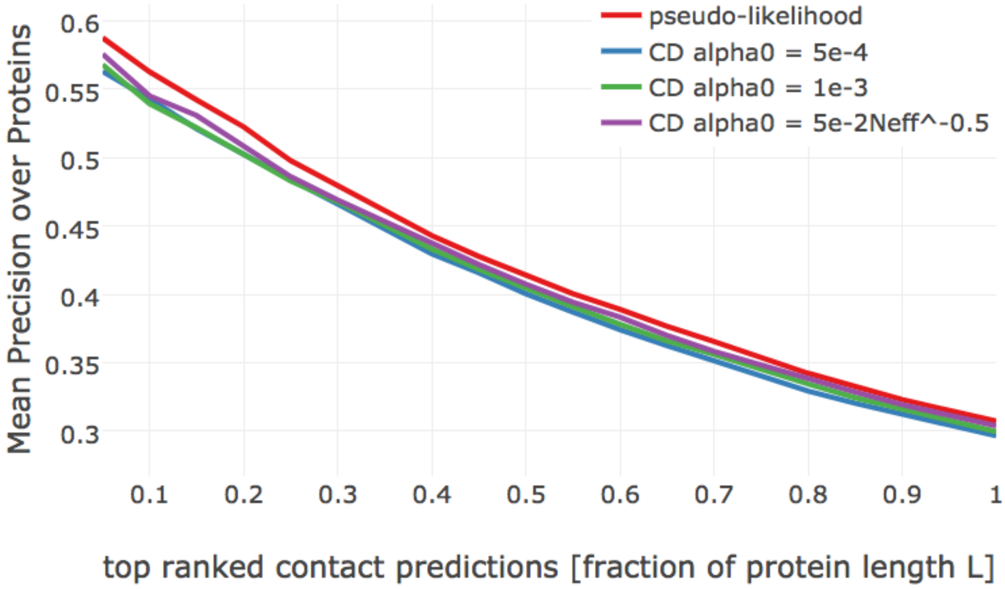


Figure E.1: Mean precision for top ranked contact predictions over 300 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. **pseudo-likelihood**: couplings computed with pseudo-likelihood. **CD alpha0 = 5e-4**: couplings computed with CD using stochastic gradient descent with initial learning rate, $\alpha_0 = 5e-4$. **CD alpha0 = 1e-3**: couplings computed with CD using stochastic gradient descent with initial learning rate, $\alpha_0 = 1e-3$. **CD alpha0 = 5e-2Neffˆ-0.5**: couplings computed with CD using stochastic gradient descent with initial learning rate defined as a function of Neff, $\alpha_0 = \frac{5e-2}{\sqrt{N_{\text{eff}}}}$.

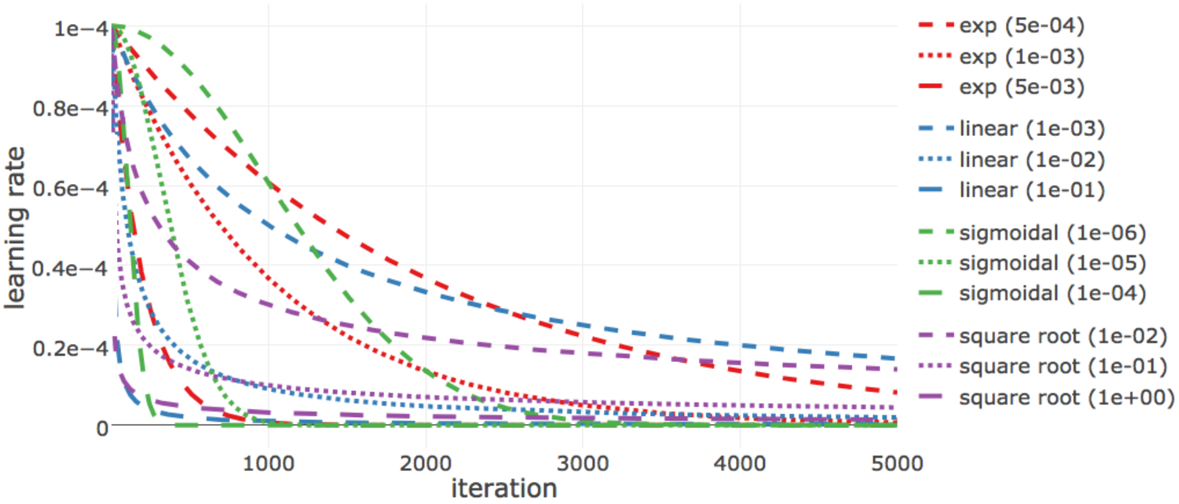


Figure E.2: Value of learning rate against the number of iterations for different learning rate schedules. Red legend group represents the **exponential** learning rate schedule $\alpha_{t+1} = \alpha_0 \cdot \exp(-\gamma t)$. Blue legend group represents the **linear** learning rate schedule $\alpha = \alpha_0 / (1 + \gamma \cdot t)$. Green legend group represents the **sigmoidal** learning rate schedule $\alpha_{t+1} = \alpha_t / (1 + \gamma \cdot t)$. Purple legend group represents the **square root** learning rate schedule $\alpha = \alpha_0 / \sqrt{1 + \gamma \cdot t}$. The initial learning rate α_0 is set to 1e-4, the iteration number is given by t and γ is the decay rate and its value is given in brackets in the legend.

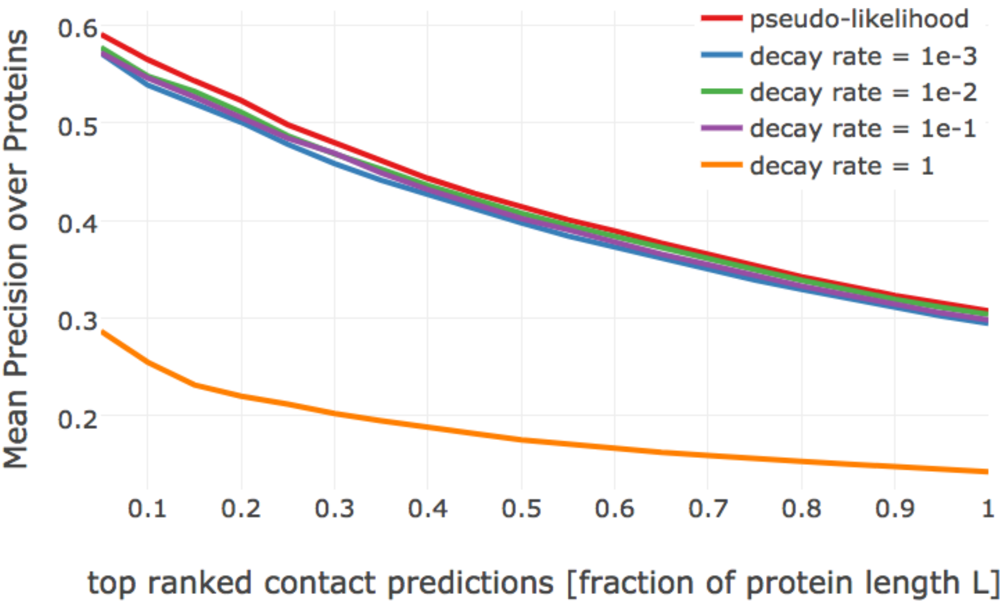


Figure E.3: Mean precision for top ranked contact predictions over 300 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. pseudo-likelihood: contact scores computed from pseudo-likelihood. The other methods derive contact scores from couplings 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.

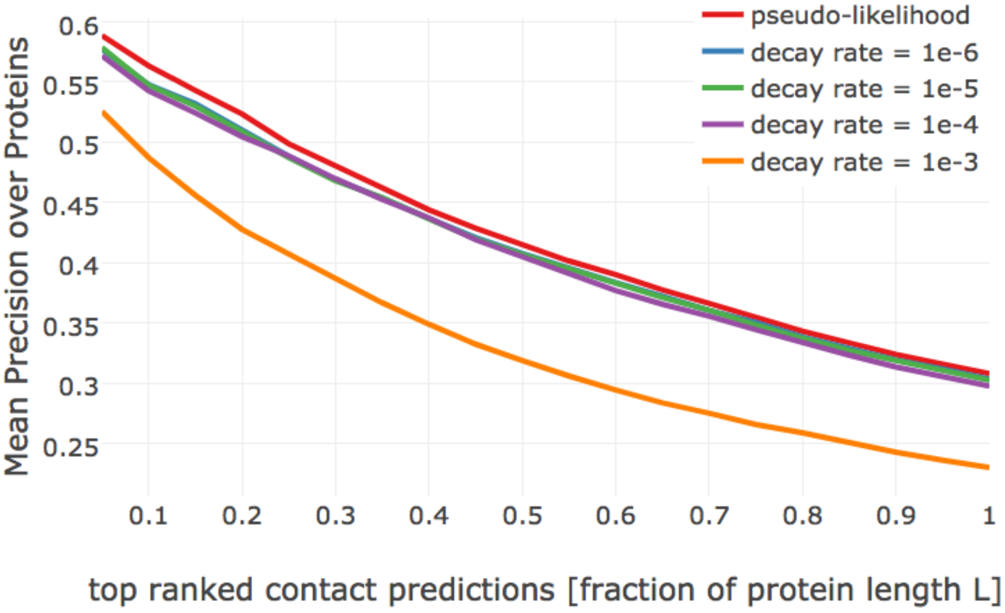


Figure E.4: Mean precision for top ranked contact predictions over 300 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. pseudo-likelihood: contact scores computed from pseudo-likelihood. The other methods derive contact scores from couplings 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.

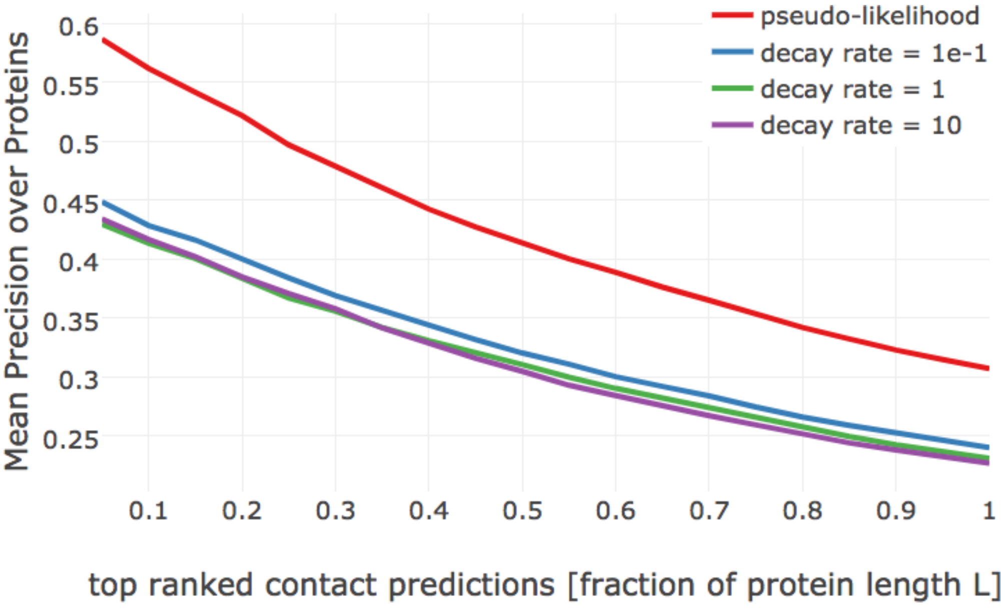


Figure E.5: Mean precision for top ranked contact predictions over 300 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. pseudo-likelihood: contact scores computed from pseudo-likelihood. The other methods derive contact scores from couplings 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.

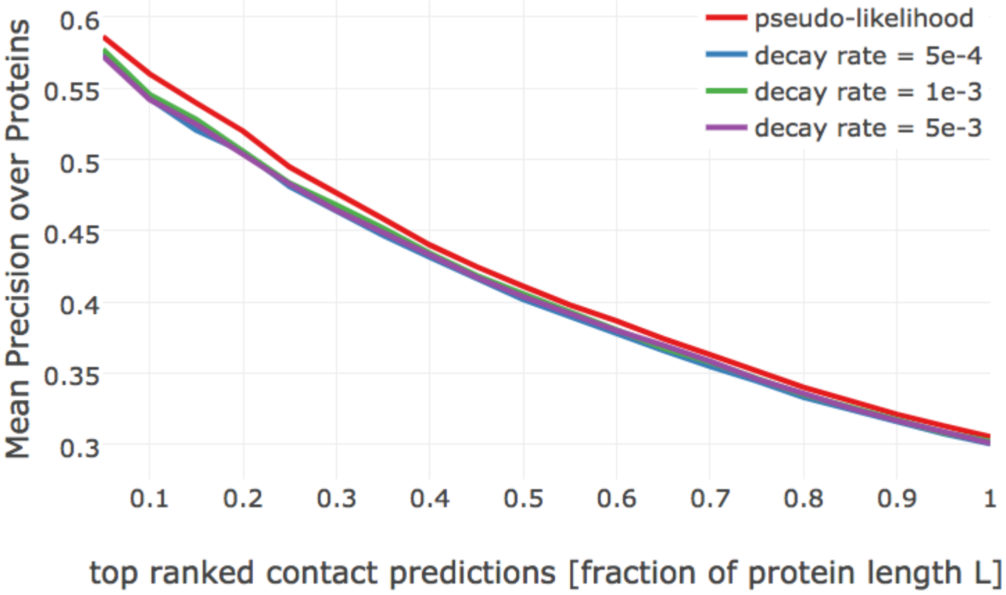


Figure E.6: Mean precision for top ranked contact predictions over 300 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. pseudo-likelihood: contact scores computed from pseudo-likelihood. The other methods derive contact scores from couplings 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.

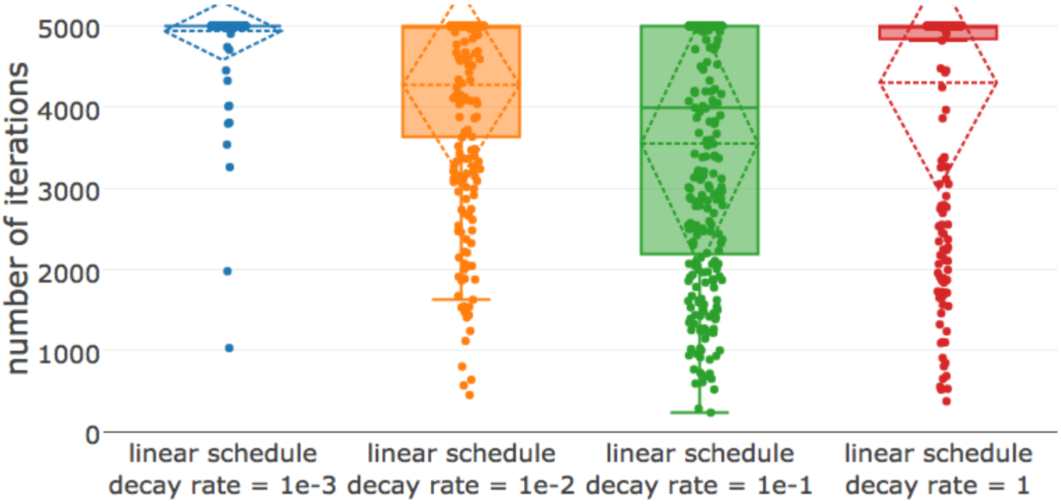


Figure E.7: Distribution of the number of iterations until convergence for stochastic gradient descent optimizations of the full likelihood using different decay rates with a **linear** learning rate schedule $\alpha = \alpha_0/(1 + \gamma t)$ with t being the iteration number and the decay rate γ is specified in the legend. Initial learning rate α_0 defined with respect to Neff and maximum number of iterations is set to 5000.

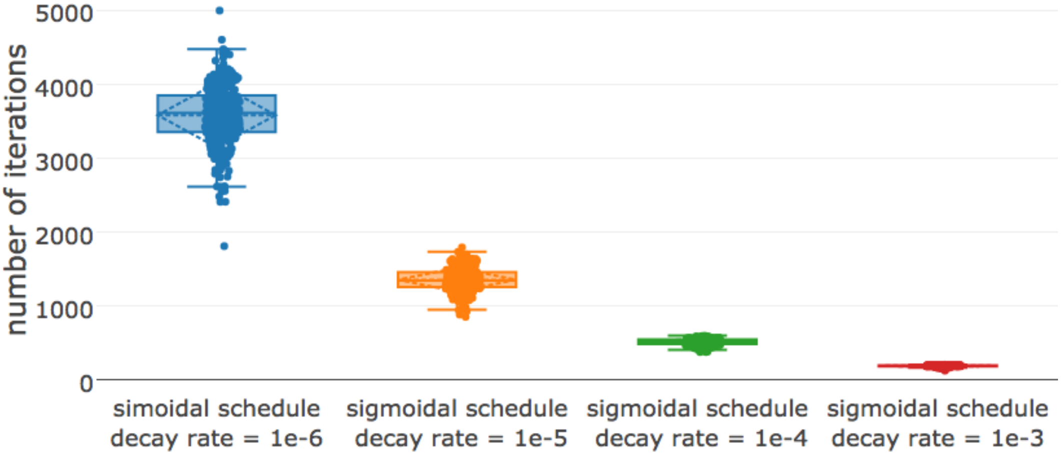


Figure E.8: Distribution of the number of iterations until convergence for stochastic gradient descent optimizations of the full likelihood using different decay rates with a **sigmoidal** learning rate schedule $\alpha_{t+1} = \alpha_t/(1 + \gamma t)$ with t being the iteration number and the decay rate γ is specified in the legend. Initial learning rate α_0 defined with respect to Neff and maximum number of iterations is set to 5000.

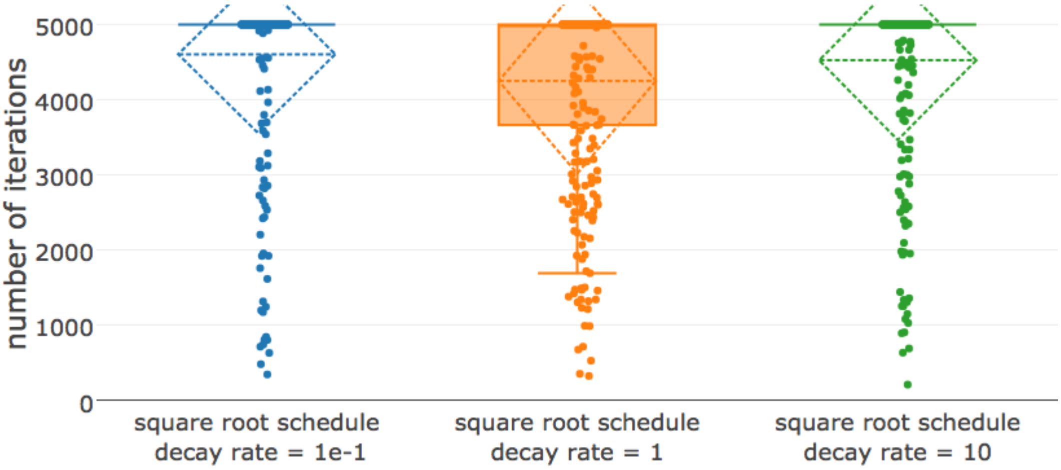


Figure E.9: Distribution of the number of iterations until convergence for stochastic gradient descent optimizations of the full likelihood using different decay rates with a **square root** learning rate schedule $\alpha = \alpha_0/\sqrt{1 + \gamma t}$ with t being the iteration number and the decay rate γ is specified in the legend. Initial learning rate α_0 defined with respect to Neff and maximum number of iterations is set to 5000.

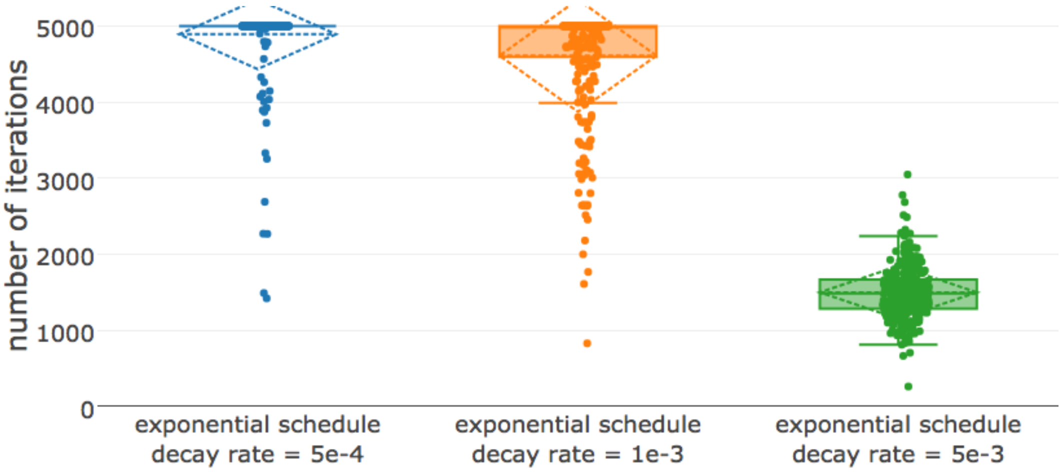


Figure E.10: Distribution of the number of iterations until convergence for stochastic gradient descent optimizations of the full likelihood using different decay rates with an **exponential** learning rate schedule $\alpha = \alpha_0 \cdot \exp(-\gamma t)$ with t being the iteration number and the decay rate γ is specified in the legend. Initial learning rate α_0 defined with respect to Neff and maximum number of iterations is set to 5000.

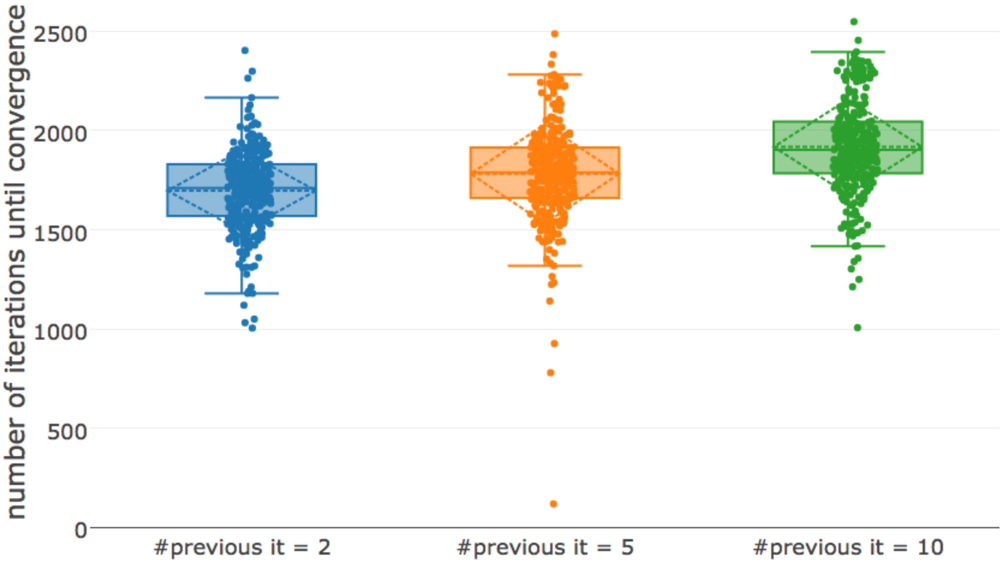


Figure E.11: Distribution of the number of iterations until convergence for gradient descent optimizations of the full likelihood. The relative change of the L2 norm over coupling parameters, $||\mathbf{w}||_2$, is evaluated over a defined number of previous iterations and is specified in the legend. Convergence is assumed when the relative change falls below a small value $\epsilon = 1e-8$. The optimal hyperparameters settings for SGD as described in section 3.2.2 have been used.

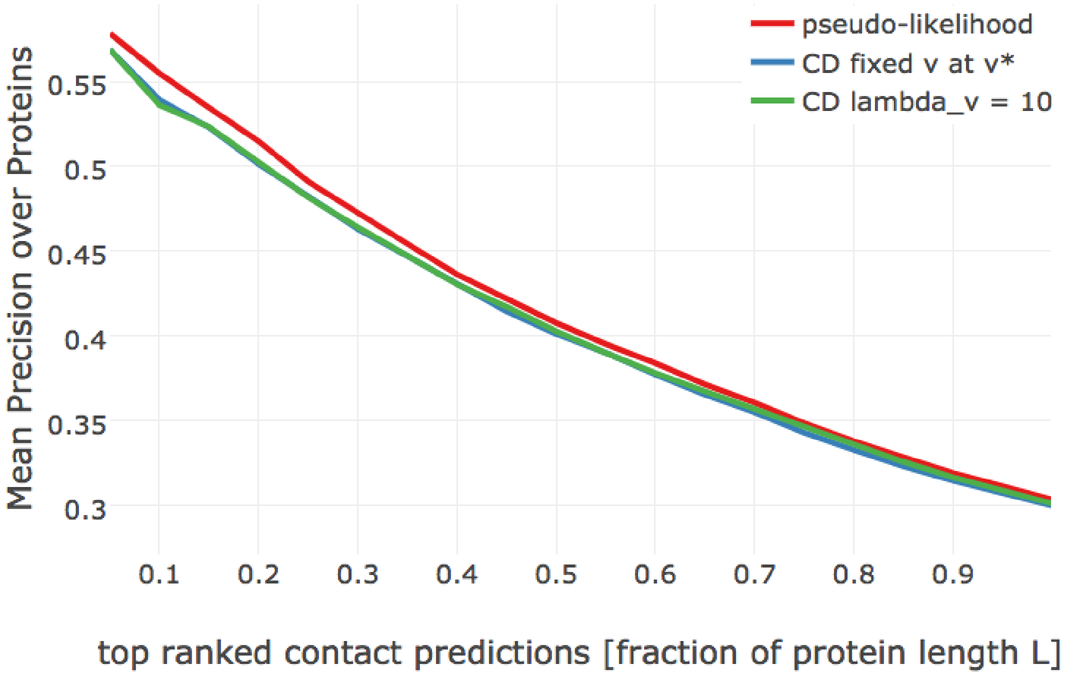


Figure E.12: Mean precision for top ranked contact predictions over 300 proteins. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. SGD settings for CD optimization are as follows: sigmoidal learning rate schedule with decay rate $\gamma = 5\text{e} - 6$ and initial learning rate $\alpha_0 = 5\text{e} - 2/N_{\text{eff}}$. **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. (3.27). **CD lambda_v = 10**: contact scores computed from CD with SGD and single potentials $\mathbf{v}$ are subject to optimization using L2-reglarization with $\lambda_v = 10$.

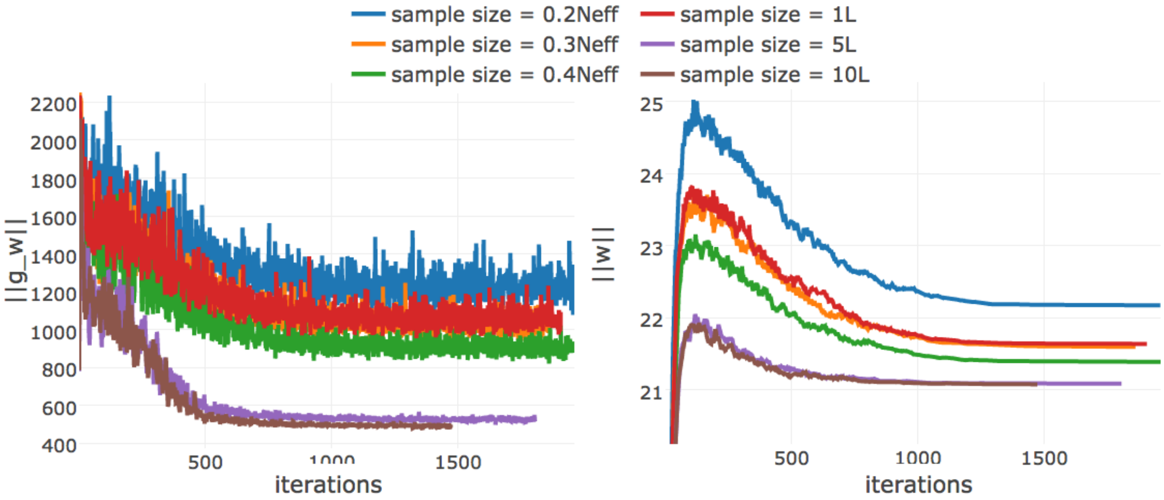


Figure E.13: Monitoring parameter norm and gradient norm for protein 1ahoA00 during SGD optimization of CD using different sample sizes. Protein 1ahoA00 has length L=64 and 378 sequences in the alignment (Neff=229). The number of sequences, that is used for Gibbs sampling to approximate the gradient, is given in the legend with 1L = 64 sequences, 5L = 320 sequences, 10L = min(10L, N) = 378 sequences, 0.2Neff = 46 sequences, 0.3Neff = 69 sequences, 0.4Neff = 92 sequences. **Left** L2-norm of the gradients for coupling parameters, $||\nabla_{\mathbf{w}} LL(\mathbf{v}^*, \mathbf{w})||_2$ (without contribution of regularizer). **Right** L2-norm of the coupling parameters $||\mathbf{w}||_2$.

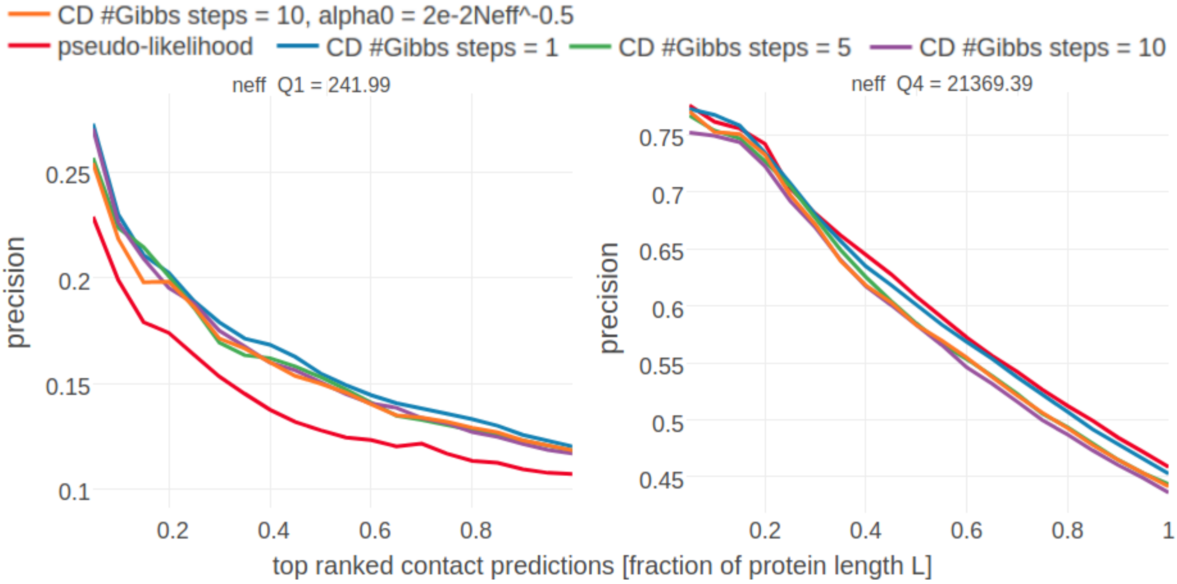


Figure E.14: Mean precision for top ranked contact predictions over 300 proteins splitted into four equally sized subsets with respect to Neff. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. Subsets are defined according to quantiles of Neff values. 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.

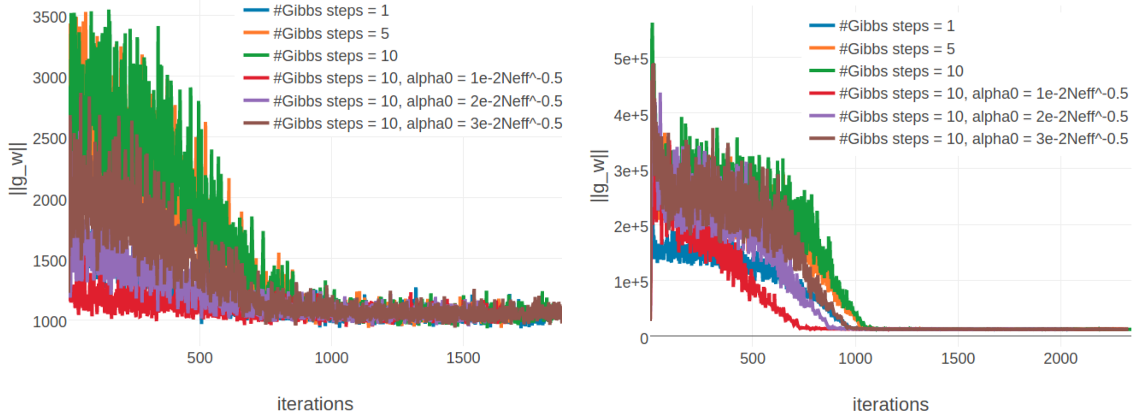


Figure E.15: Monitoring L2 norm of the gradient, $||\nabla_{\mathbf{w}} LL(\mathbf{v}^*, \mathbf{w})||_2$, for protein 1aho_A_00 and 1c75_A_00 during SGD optimization using different number of Gibbs steps and initial learning rates, α_0. Number of Gibbs steps is given in the legend, as well as particular 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).

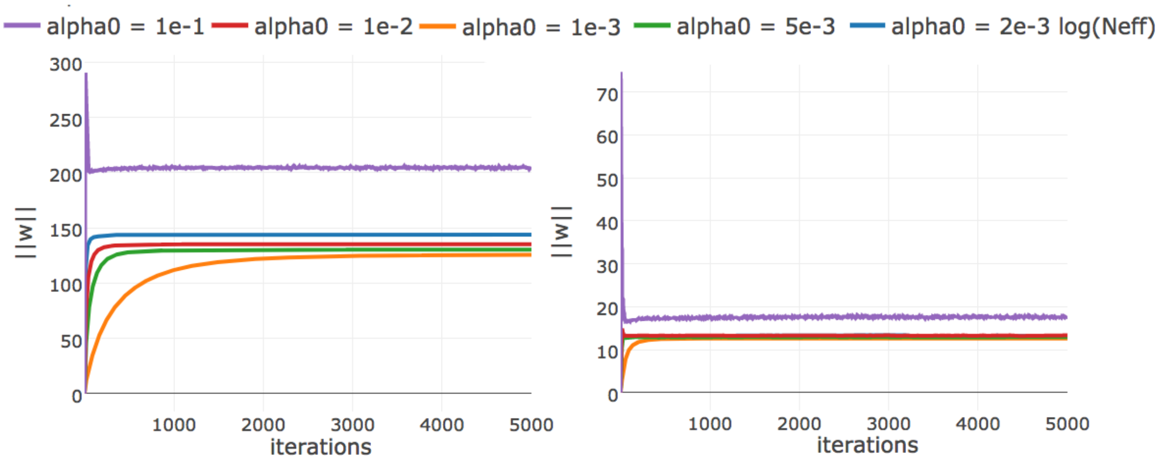


Figure E.16: L2-norm of the coupling parameters, $||\mathbf{w}||_2$, during CD optimization with *ADAM* with different fixed learning rates (no decay). The learning rate α_0 is specified in the legend. **Left** Protein 1c75A00 has length L=71 and 28078 sequences in the alignment (Neff=16808) **Right** Protein 1mkcA00 has length L=43 and 142 sequences in the alignment (Neff=96).

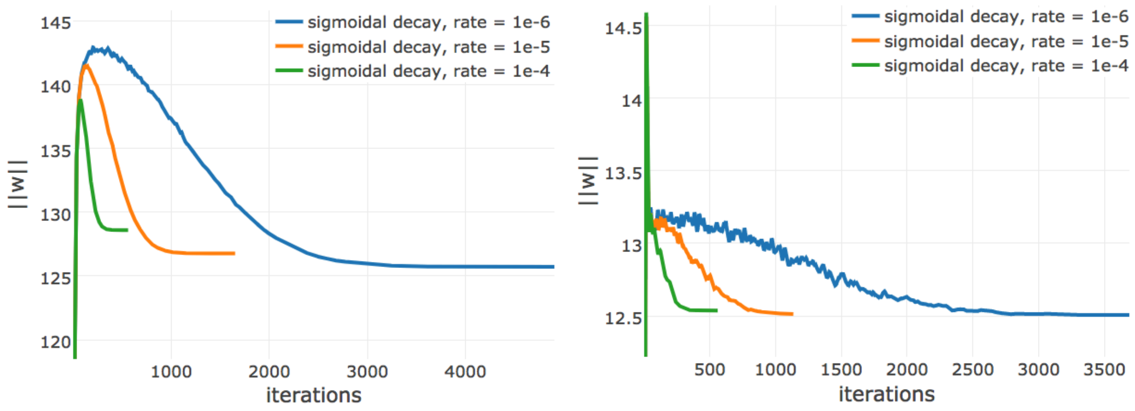


Figure E.17: L2-norm of the coupling parameters, $||\mathbf{w}||_2$, during CD optimization with *ADAM* and different learning rate annealing schedules. The learning rate α is specified with respect to Neff as $\alpha = 2\mathrm{e}{-3}\log(\mathrm{N}_{\mathrm{eff}})$. The learning rate annealing schedule 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). **Left** Protein 1c75A00 has length L=71 and 28078 sequences in the alignment (Neff=16808) **Right** Protein 1mkcA00 has length L=43 and 142 sequences in the alignment (Neff=96).

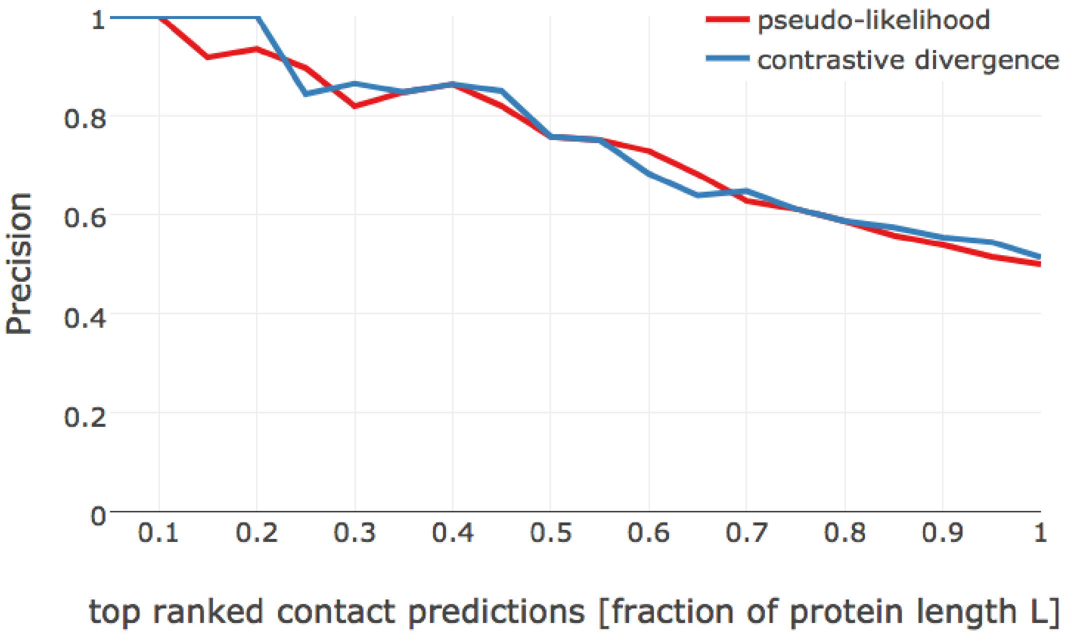


Figure E.18: Precision of top ranked contact predictions for protein 1c75A00. Contact scores are computed as the APC corrected Frobenius norm of the couplings $\mathbf{w}_{ij}$. **pseudo-likelihood**: contact scores computed from pseudo-likelihood. **contrastive divergence**: contact scores computed from CD optimized with SGD.

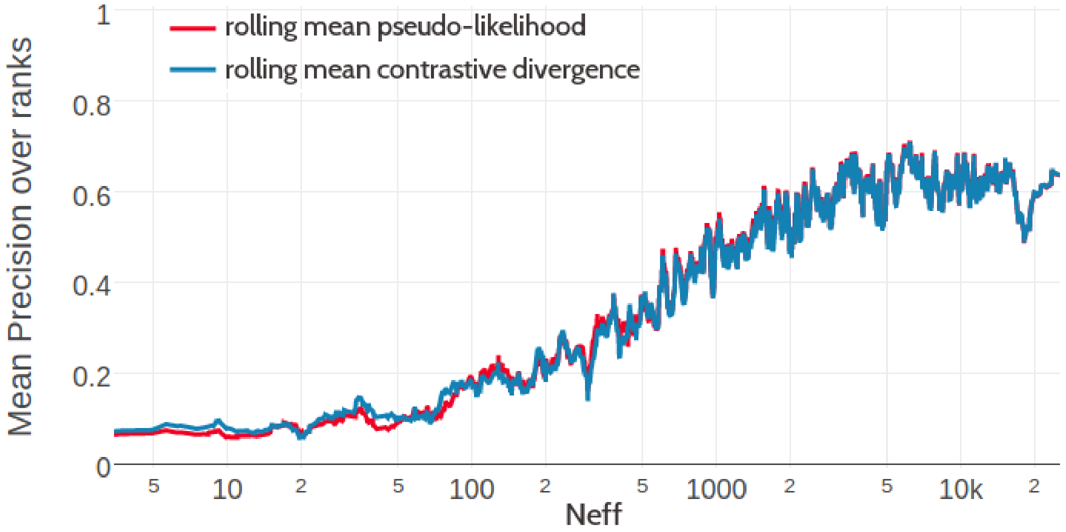


Figure E.19: Rolling mean over the mean precision of the L/10 to L top ranked predictions per protein for testset with 2300 proteins. Contact scores computed as APC corrected Frobenius norm over pseudo-likelihood and contrastive divergence couplings. The rolling mean has been computed for the central protein within a window of 20 proteins. Window is shrunk for the proteins at the borders of Neff distribution.

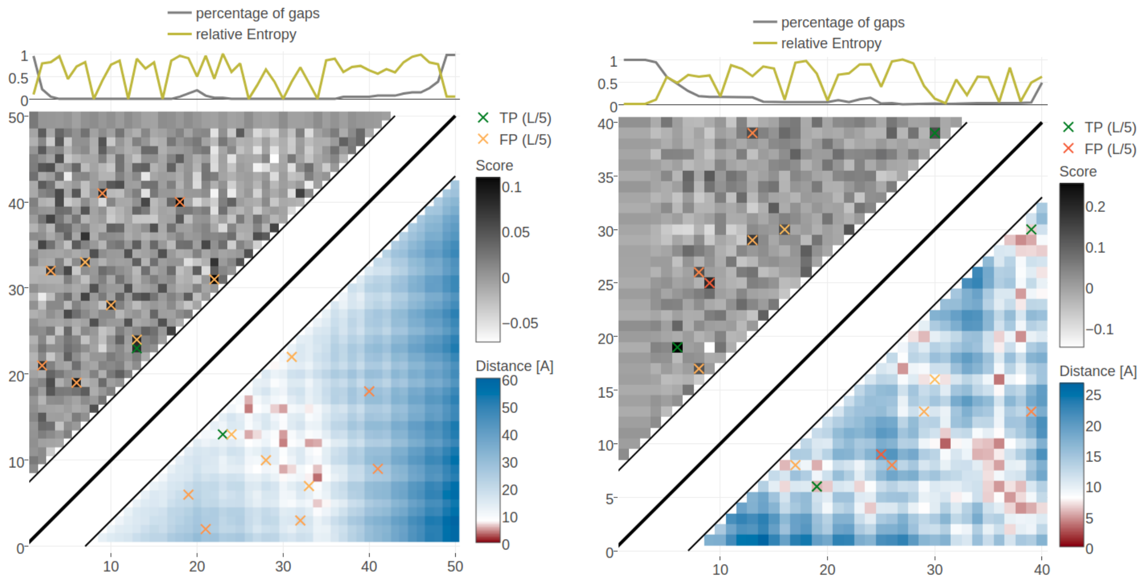


Figure E.20: Contact maps for protein 1ss3A00 and 1c55A00 computed as APC corrected Frobenius norm of the pseudo-likelihood couplings. Contacts are defined according to a $8\mathring{A}$ C_β distance cutoff. **Left**: predicted contact map and native distance map for protein 1ss3A00 (protein length=50, N=42, Neff=36). **Right** predicted contact map and native distance map for protein 1c55A00 (protein length = 40, N=115, Neff = 78).

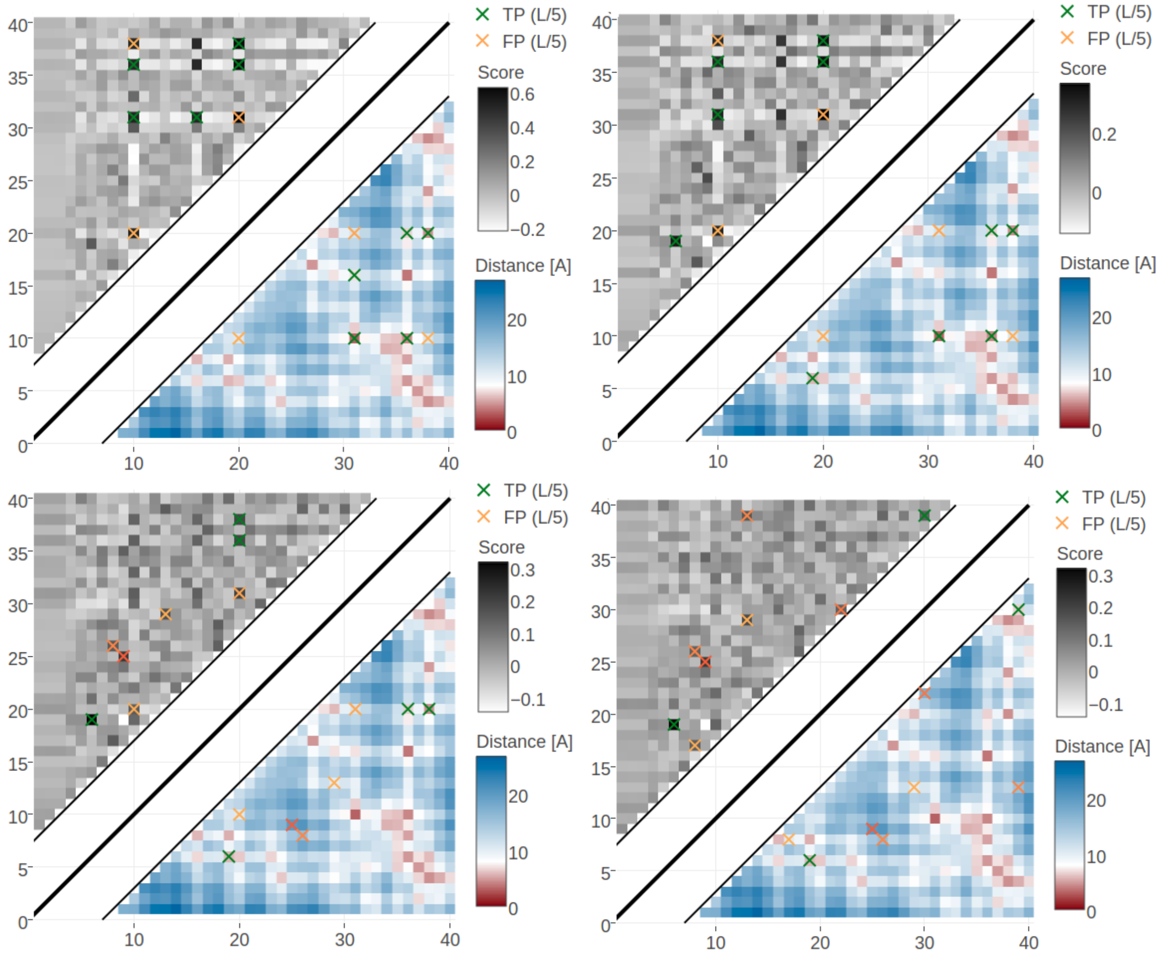


Figure E.21: Contact maps for protein 1c55A00 (protein length = 40, N=115, Neff = 88) computed as APC corrected Frobenius norm of the contrastive-divergence couplings computed with different sample size choices. Contacts are defined according to a 8$\mathring{A}$ C_β distance cutoff. **Top Left**: sample size=0.3neff ≈ 23 sequences. **Top Right** sample size=0.5neff ≈ 39 sequences. **Bottom Left**: sample size=0.8neff ≈ 62 sequences. **Bottom Right** sample size=max(10L,N)=>115 sequences.

F

Training of the Random Forest Contact Prior

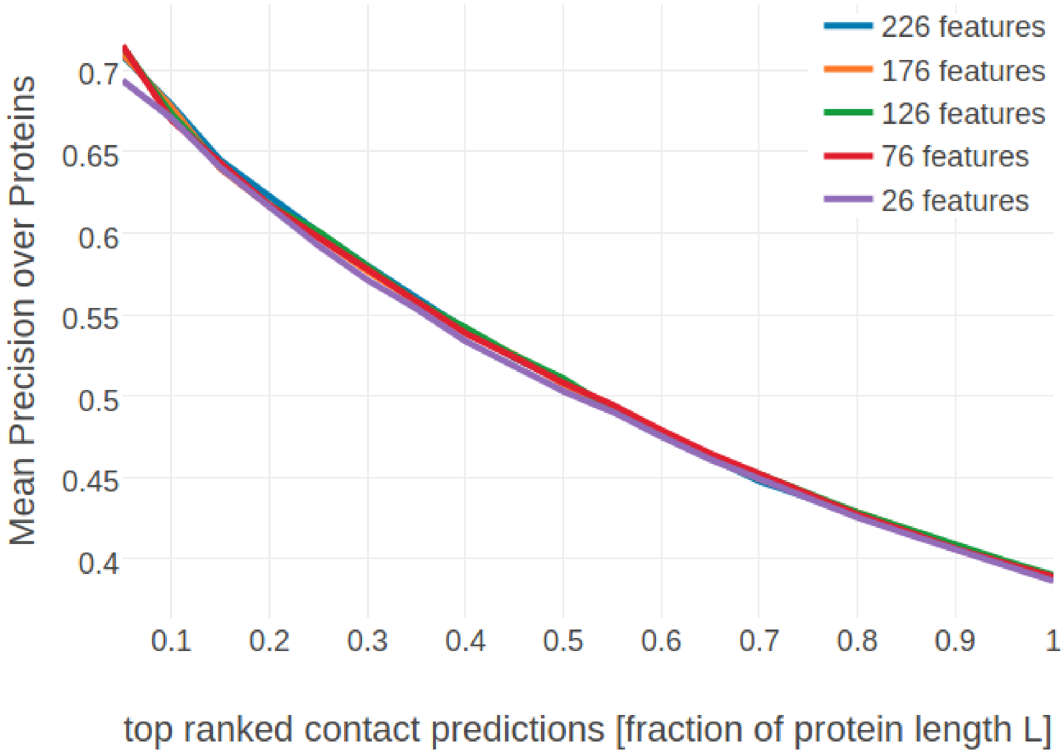


Figure F.1: Mean precision for top ranked contacts over 200 proteins for variaous random forest models trained on subsets of features. Subsets of features have been selected as described in section 4.6.4.

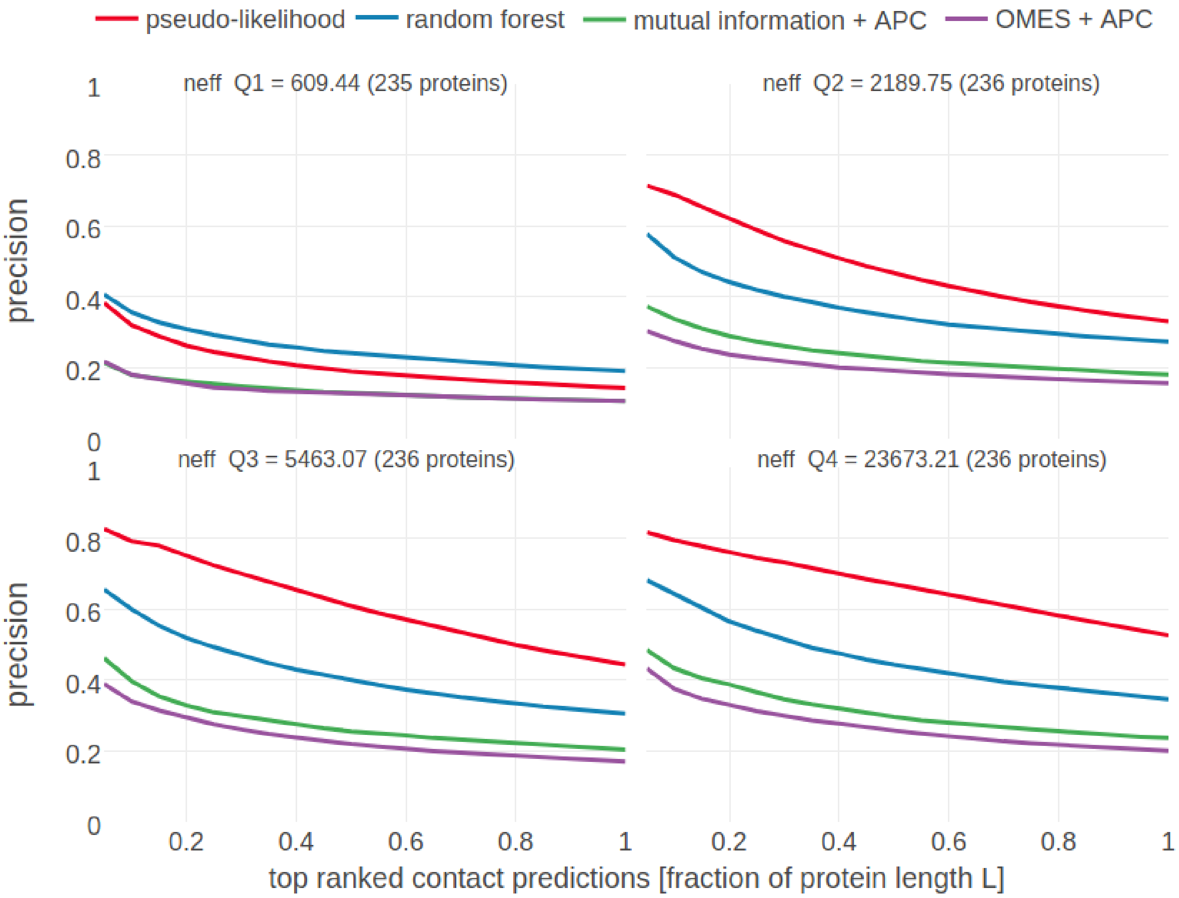


Figure F.2: Mean precision for top ranked contacts predicted with random forest on a test set of 1000 proteins splitted into four equally sized subsets with respect to Neff. Subsets are defined according to quantiles of Neff values. Upper left: Subset of proteins with Neff < Q1. Upper right: Subset of 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 [224]. **mutual information** = APC corrected mutual information between amino acid counts (using pseudo-counts).

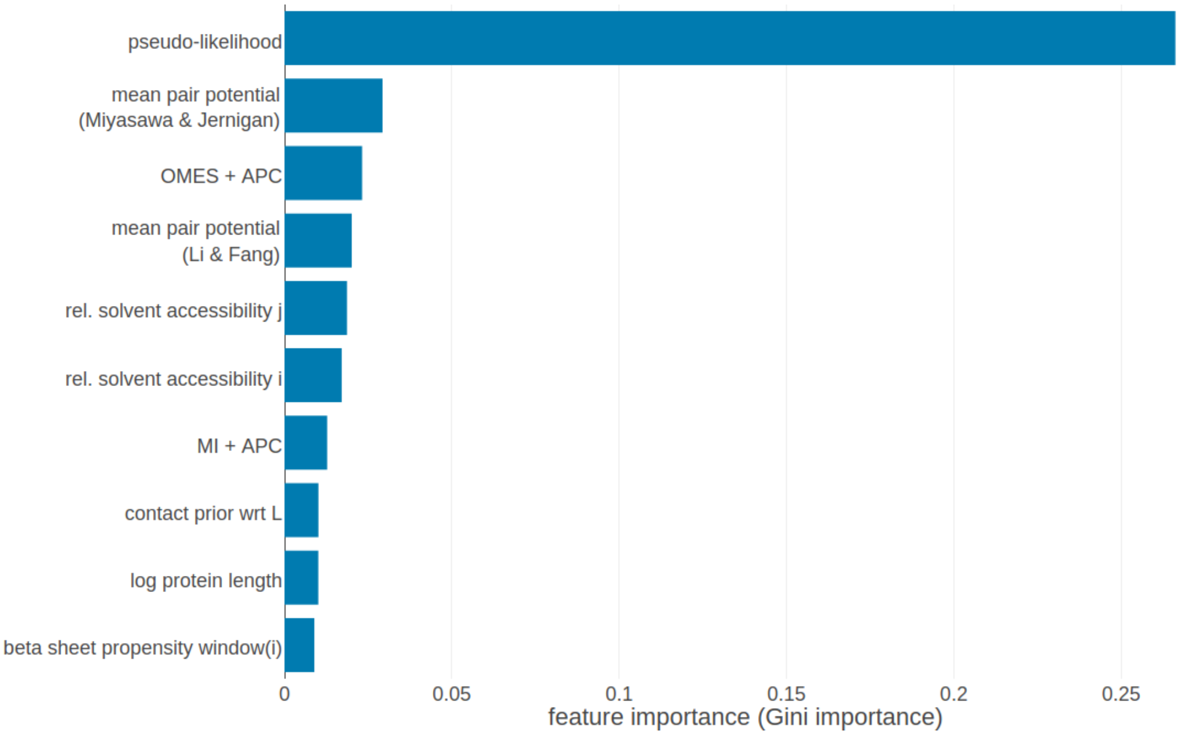


Figure F.3: Top ten features for Random Forest trained with additional pseudo-likelihood contact score feature. Features ranked according to *Gini importance*. **pseudo-likelihood**: APC corrected Frobenius norm of couplings computed with pseudo-likelihood. **mean pair potential (Miyasawa & Jernigan)**: average quasi-chemical energy of transfer of amino acids from water to the protein environment [225]. **OMES+APC**: APC corrected OMES score according to Fodor&Aldrich [224]. **mean pair potential (Li&Fang)**: average general contact potential by Li & Fang [70]. **rel. solvent accessibilty i(j)**: RSA score computed with Netsurfp (v1.0) [226] 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 4.6.2). **log protein length**: logarithm of protein length. **beta sheet propensity window(i)**: beta-sheet propensity according to Psipred [227] computed within a window of five positions around i. Features are described in detail in methods section 4.6.1.

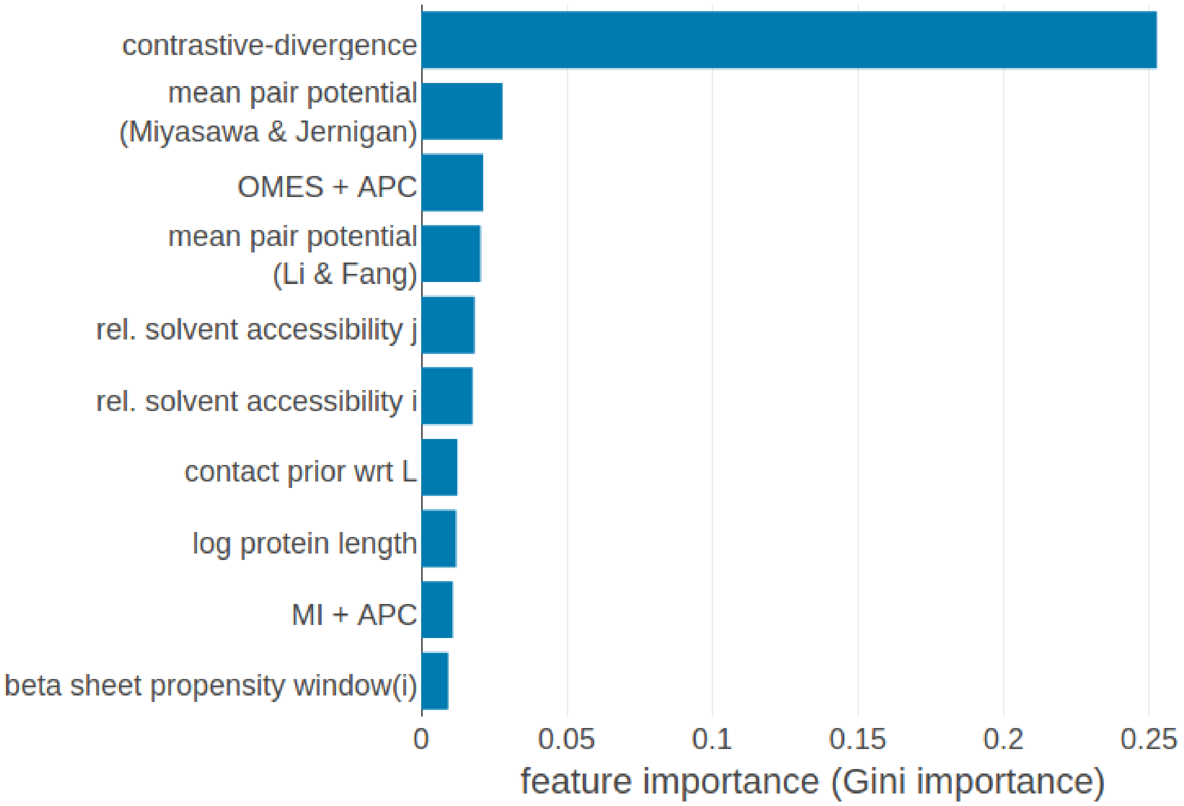


Figure F.4: Top ten features for Random Forest trained with additional contrastive divergence contact score feature. Features ranked according to *Gini importance*. Features are the same as in Figure F.3 plus the following additional features: **contrastive divergence**: APC corrected Frobenius norm of couplings computed with contrastive divergence. Features are described in detail in methods section 4.6.1.

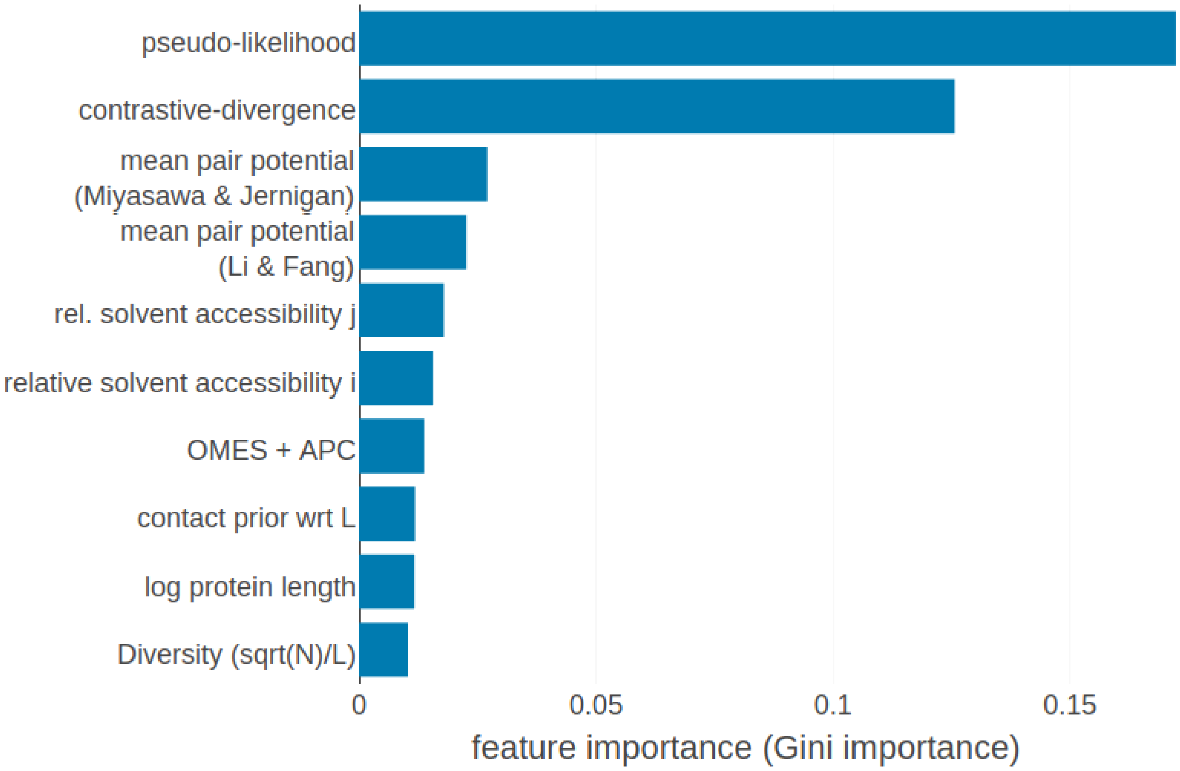


Figure F.5: Top ten features for Random Forest trained with additional pseudo-likleihood and contrastive divergence contact score feature. Features ranked according to *Gini importance*. Features are the same as in Figure F.3 plus the following additional features: **contrastive divergence**: APC corrected Frobenius norm of couplings computed with contrastive divergence. **Diversity (sqrt(N)/L)**: diversity of the alignment. Features are described in detail in methods section 4.6.1.

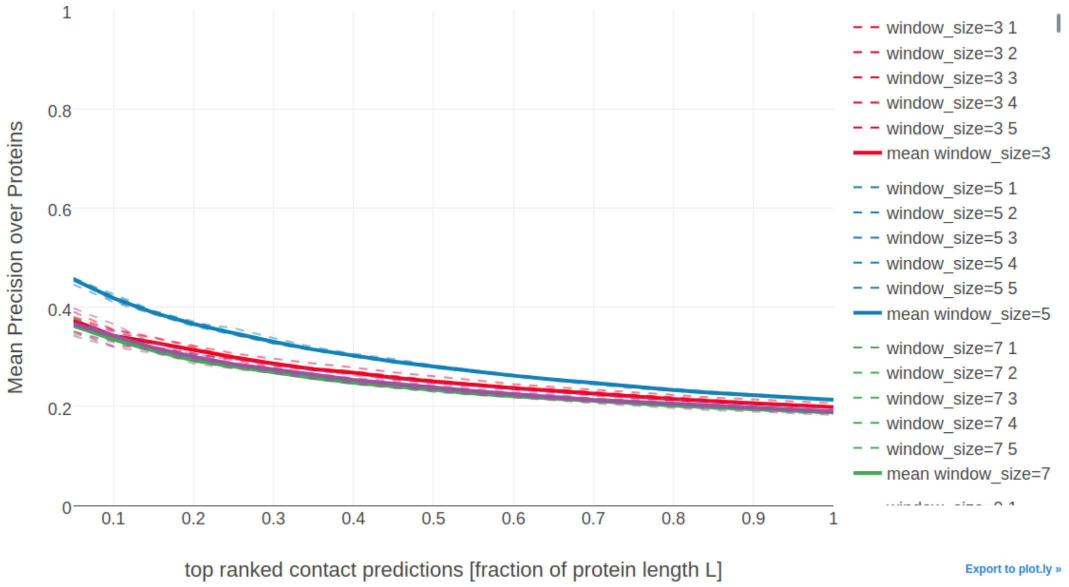


Figure F.6: Mean precision over validation set of 200 proteins for top ranked contact predictions for different choices of window size for single position features. Dashed lines represent the models trained on four subsets of the training data according to the 5-fold cross-validation scheme. Solid lines represent the mean over the five cross-validation models.

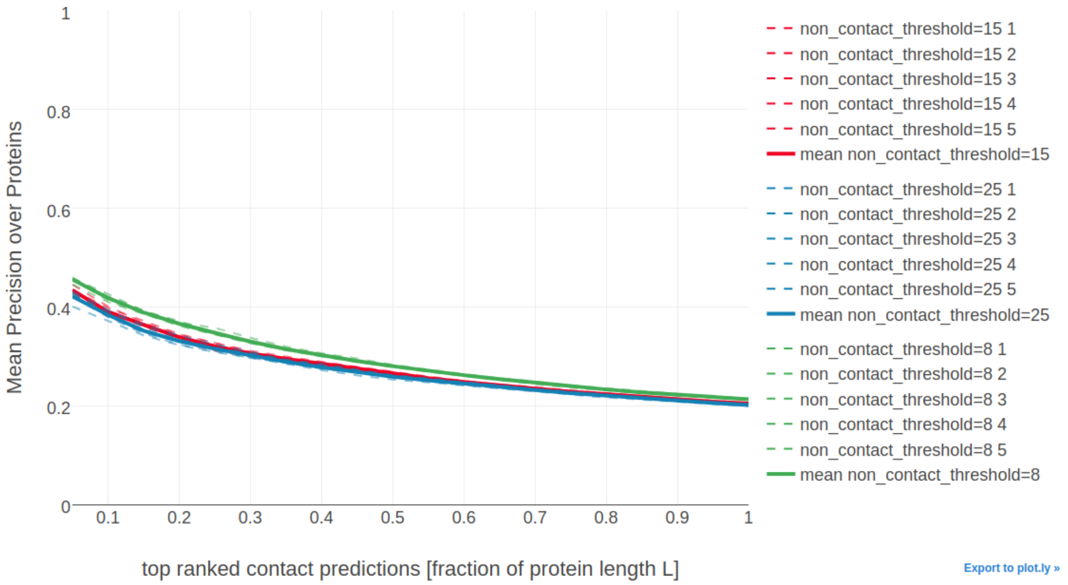


Figure F.7: Mean precision over validation set of 200 proteins for top ranked contact predictions for different choices of the non-contact threshold to define non-contacts. Dashed lines represent the models trained on four subsets of the training data according to the 5-fold cross-validation scheme. Solid lines represent the mean over the five cross-validation models.

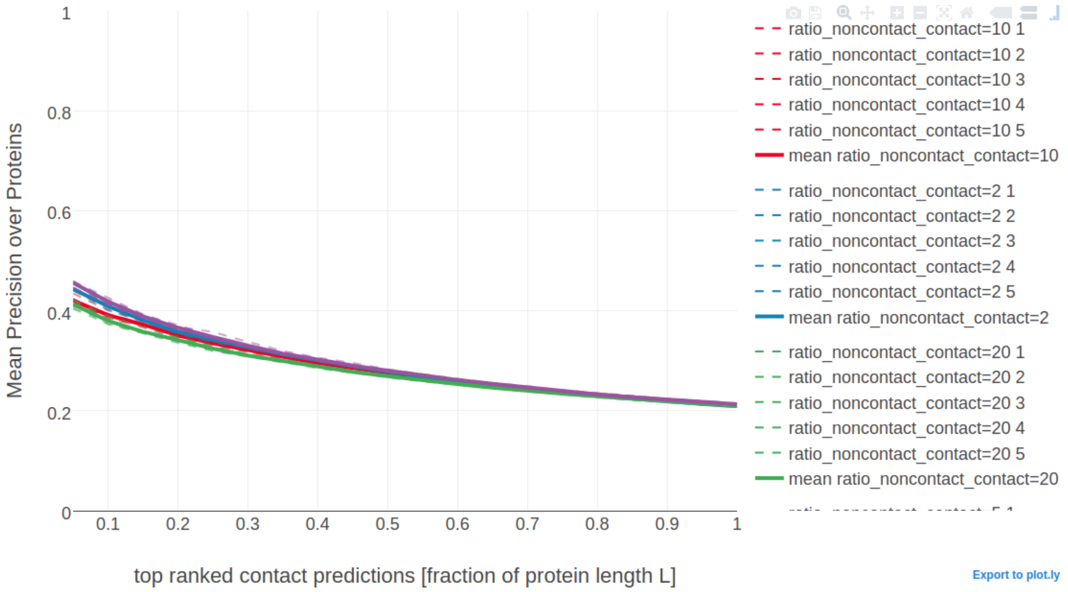


Figure F.8: Mean precision over validation set of 200 proteins for top ranked contact predictions for different choices of dataset composition with respect to the ratio of contacts and non-contacts. Dashed lines represent the models trained on four subsets of the training data according to the 5-fold cross-validation scheme. Solid lines represent the mean over the five cross-validation models.

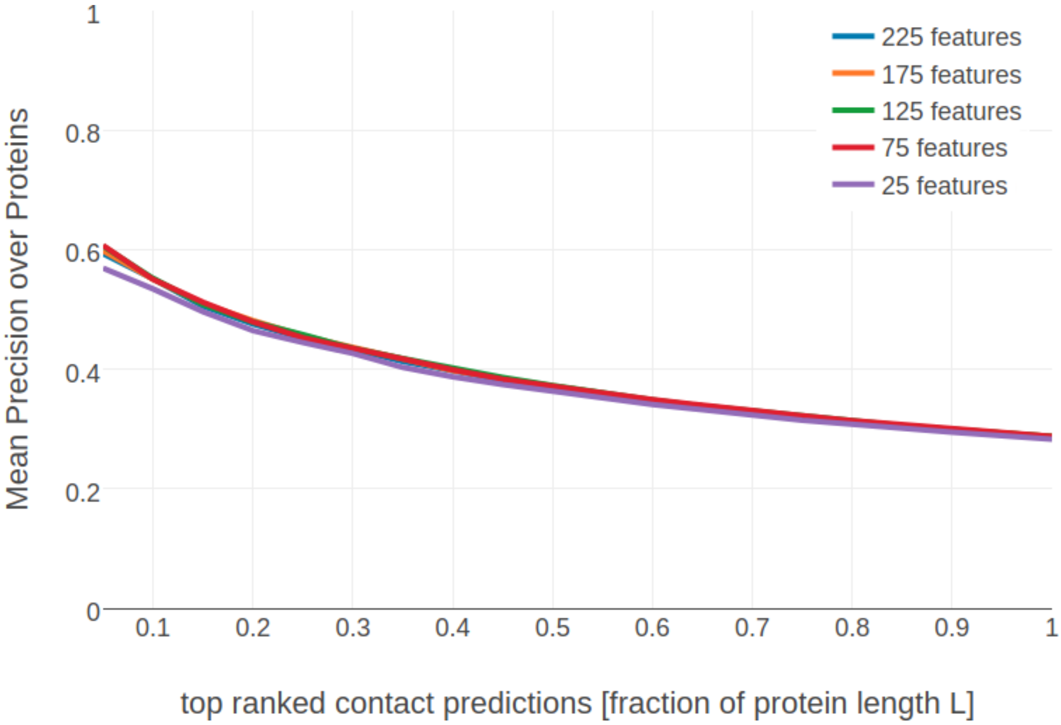


Figure F.9: Mean precision of top ranked predictions over 200 proteins for random forest models trained on subsets of features of decreasing importance. Subsets of features have been selected as described in methods section 4.6.4.

G

Bayesian statistical model for contact prediction

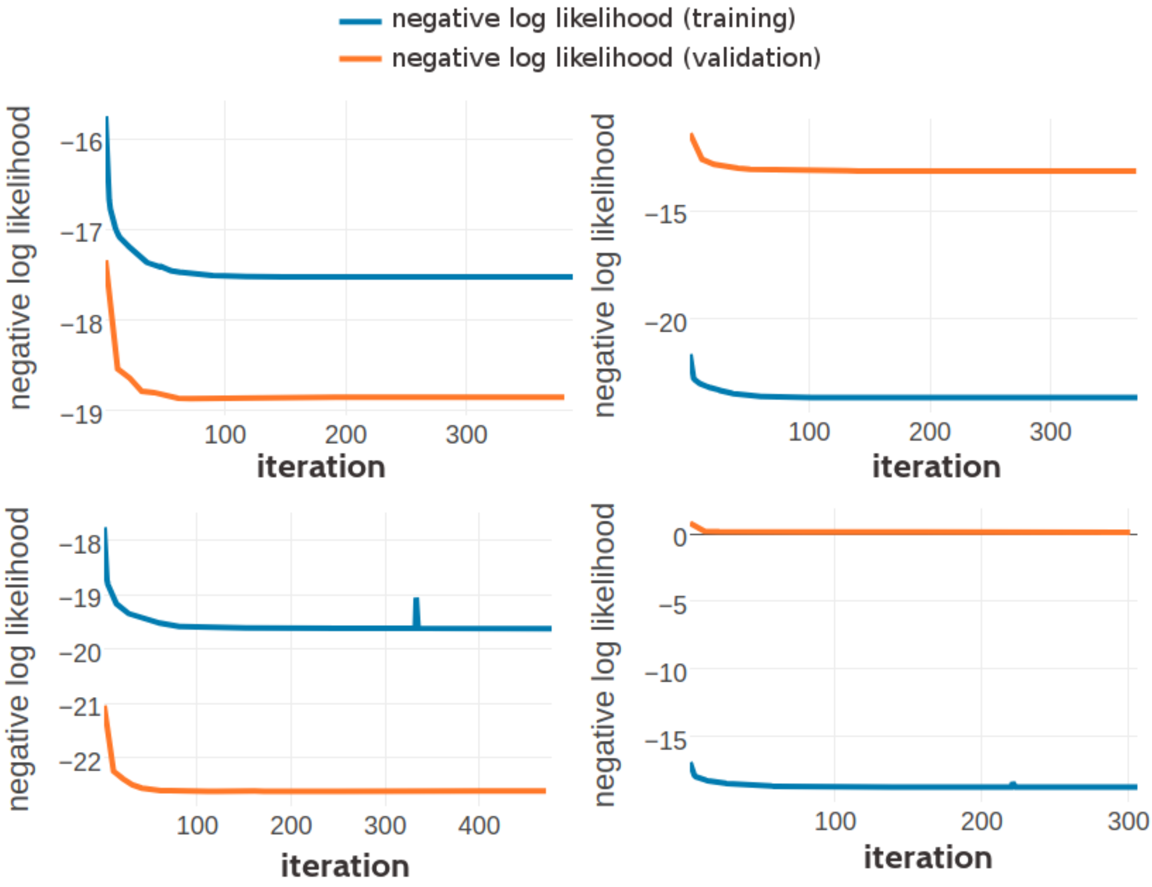


Figure G.1: Monitoring the negative log likelihood during optimization of three component Gaussian mixture using *pseudo-likelihood* couplings to estimate the Hessian. **Top Left**: Training set contains 10,000 residue pairs per contact class. Converged after 388 iterations. **Top Right**: Training set contains 100,000 residue pairs per contact class. Converged after 371 iterations. **Bottom Left**: Training set contains 300,000 residue pairs per contact class. **Bottom Right**: Training set contains 500,000 residue pairs per contact class.

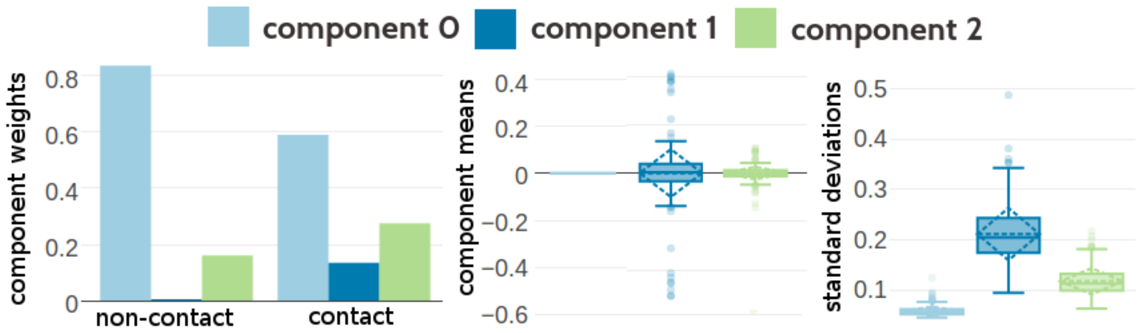


Figure G.2: Statistics for the hyperparameters, $\gamma_k(c_{ij})$, μ_k and $\mathbf{\Lambda}_k$ obtained after 388 iterations. Trained on 10,000 residue pairs per contact class for a three component Gaussian mixture and using *pseudo-likelihood* couplings to estimate the Hessian. **Left** Component weights $\gamma_k(c_{ij})$ for residue pairs not in physical contact ($c_{ij}=0$) and true contacts ($c_{ij}=1$). **Center** Distribution of the 400 elements in the mean vectors μ_k. **Right** Distribution of the 400 standard deviations corresponding to the square root of the diagonal of $\mathbf{\Lambda}_k^{-1}$.

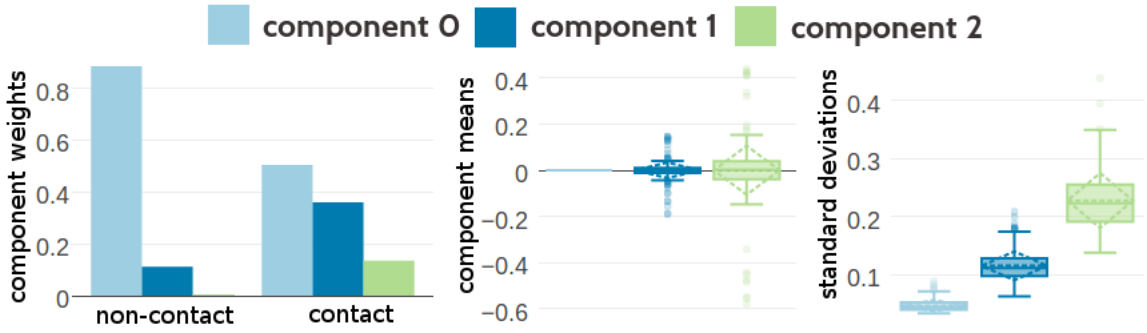


Figure G.3: Statistics for the hyperparameters, $\gamma_k(c_{ij})$, μ_k and $\mathbf{\Lambda}_k$ obtained after 371 iterations. Trained on 100,000 residue pairs per contact class for a three component Gaussian mixture and using *pseudo-likelihood* couplings to estimate the Hessian. **Left** Component weights $\gamma_k(c_{ij})$ for residue pairs not in physical contact ($c_{ij}=0$) and true contacts ($c_{ij}=1$). **Center** Distribution of the 400 elements in the mean vectors μ_k. **Right** Distribution of the 400 standard deviations corresponding to the square root of the diagonal of $\mathbf{\Lambda}_k^{-1}$.

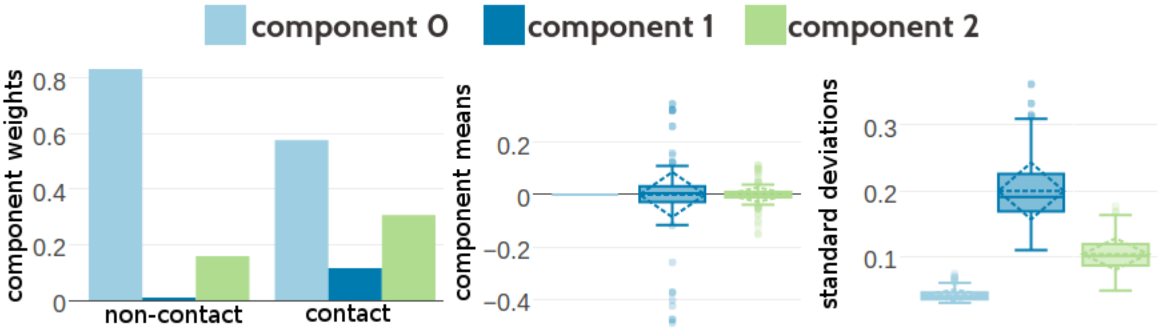


Figure G.4: Statistics for the hyperparameters, $\gamma_k(c_{ij})$, μ_k and $\mathbf{\Lambda}_k$ obtained after optimization of the likelihood function of contact states for 336 iterations. Trained on 10,000 residue pairs per contact class for a three component Gaussian mixture and using *contrastive divergence* couplings to estimate the Hessian. **Left** Component weights $\gamma_k(c_{ij})$ for residue pairs not in physical contact ($c_{ij}=0$) and true contacts ($c_{ij}=1$). **Center** Distribution of the 400 elements in the mean vectors μ_k. **Right** Distribution of the 400 standard deviations corresponding to the square root of the diagonal of $\mathbf{\Lambda}_k^{-1}$.

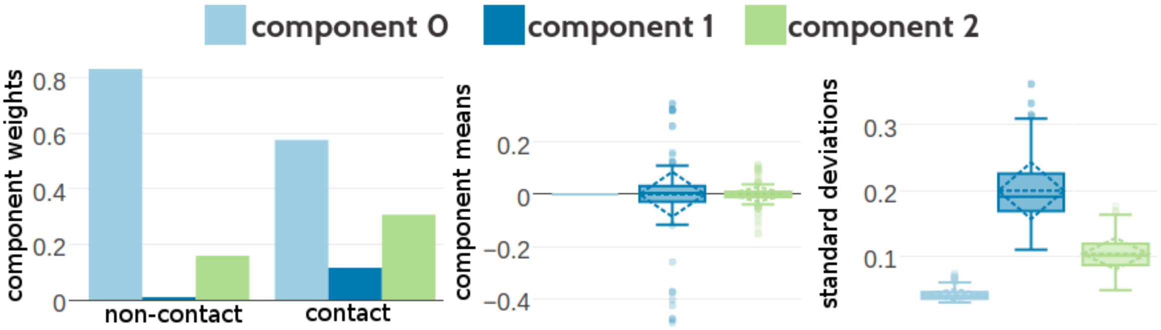


Figure G.5: Statistics for the hyperparameters, $\gamma_k(c_{ij})$, μ_k and $\mathbf{\Lambda}_k$ obtained after optimization of the likelihood function of contact states for 377 iterations. Trained on 100,000 residue pairs per contact class for a three component Gaussian mixture and using *contrastive divergence* couplings to estimate the Hessian. **Left** Component weights $\gamma_k(c_{ij})$ for residue pairs not in physical contact ($c_{ij}=0$) and true contacts ($c_{ij}=1$). **Center** Distribution of the 400 elements in the mean vectors μ_k. **Right** Distribution of the 400 standard deviations corresponding to the square root of the diagonal of $\mathbf{\Lambda}_k^{-1}$.

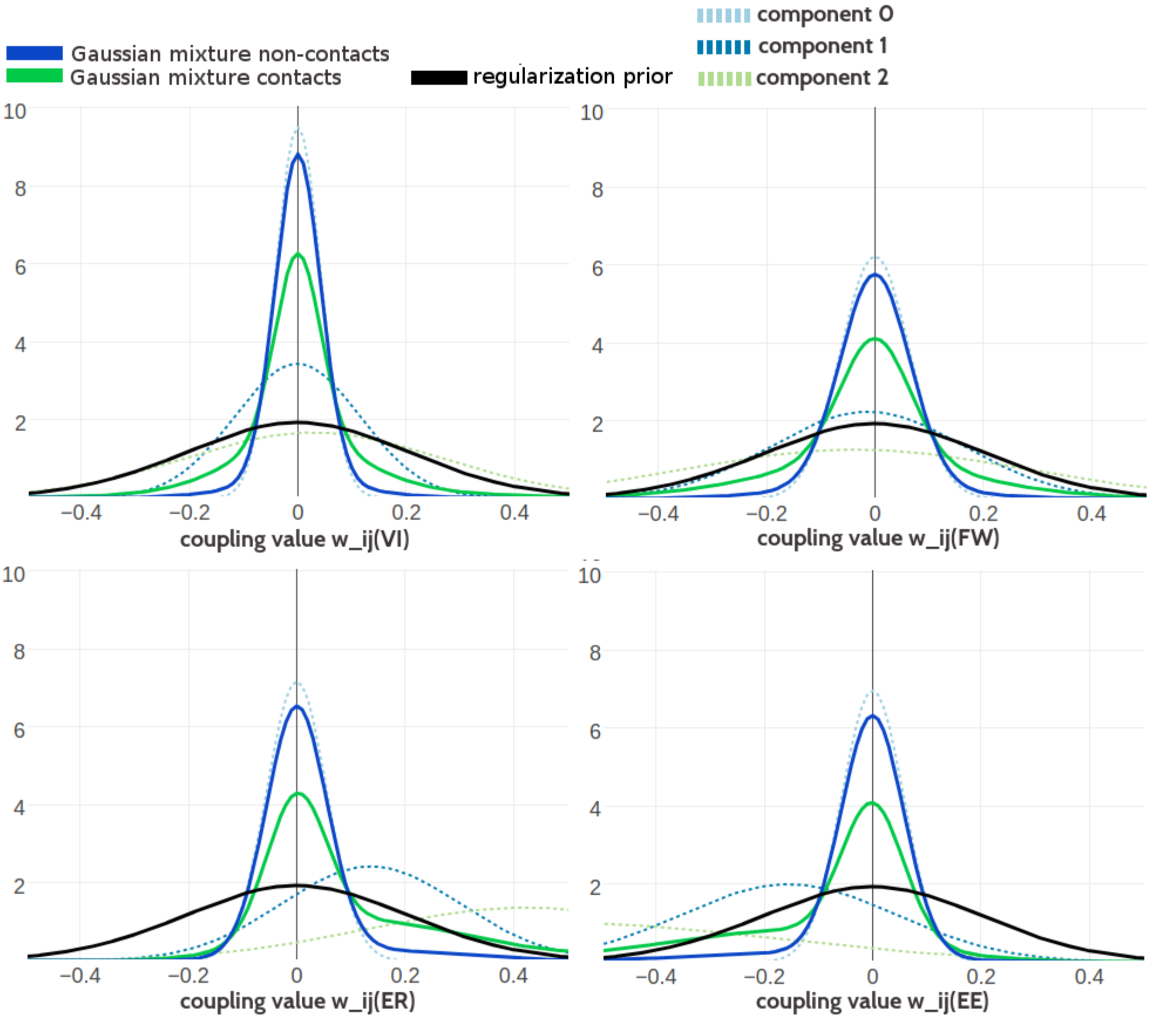


Figure G.6: Visualisation of one-dimensional projections of the three-component Gaussian mixture model for the contact-dependent coupling prior. Hyperparameters, $\gamma_k(c_{ij})$, μ_k and $\mathbf{\Lambda}_k$, have been trained on 100,000 residue pairs per contact class and using *pseudo-likelihood* couplings to estimate the Hessian. Green solid line: Gaussian mixture for contacts. Blue solid line: Gaussian mixture for non-contacts. Black solid line: regularization prior with $\lambda_1 = 0.2L$ with L being protein length and assumed $L = 150$. Light blue dashed line: Gaussian component 0. Dark blue dashed line: Gaussian component 1. Light green dashed line: Gaussian component 2. **Top Left** One dimensional projection for pair (V,I). **Top Right** One dimensional projection for pair (F,W). **Bottom Left** One dimensional projection for pair (E,R). **Bottom Right** One dimensional projection for pair (E,E).

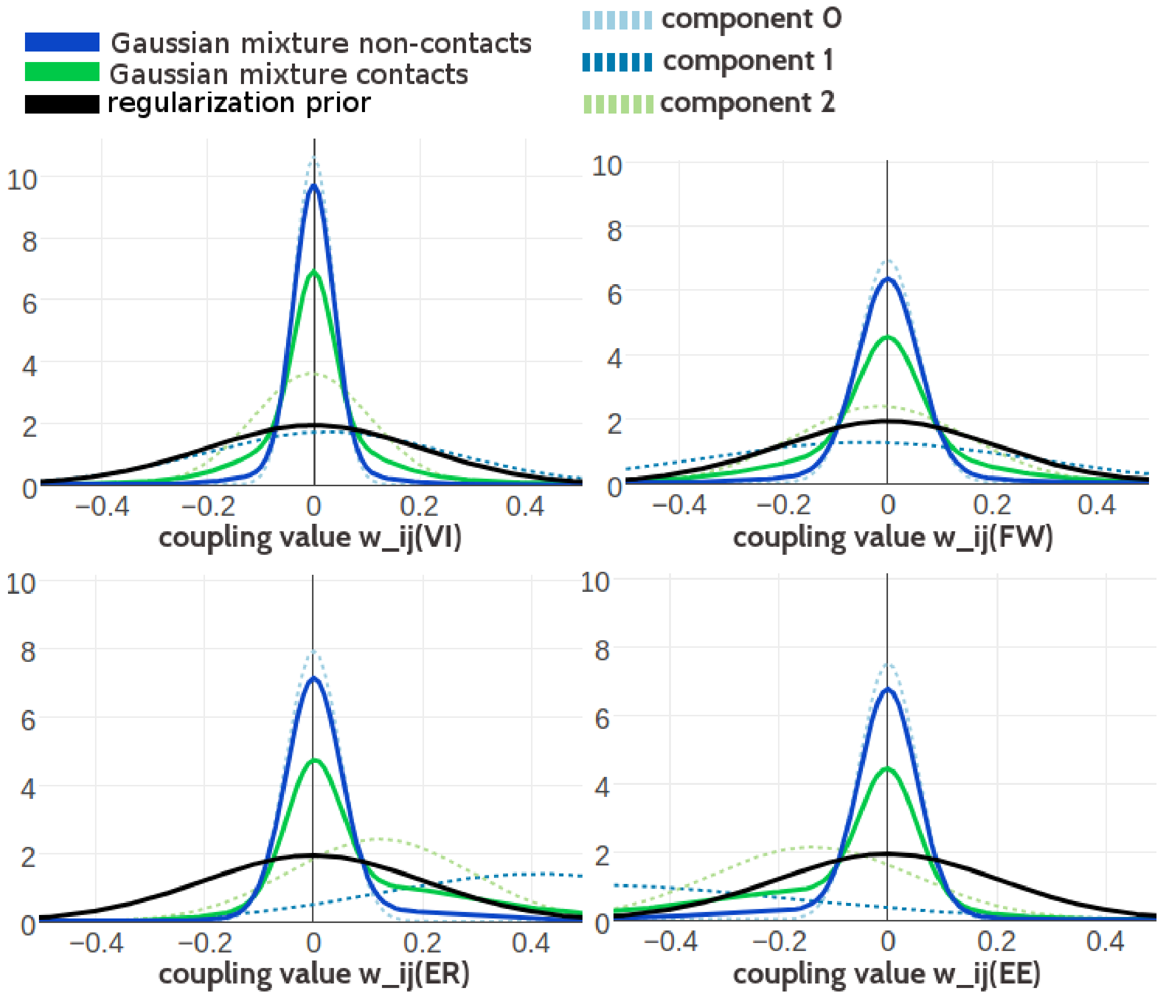


Figure G.7: Visualisation of one-dimensional projections of the three-component Gaussian mixture model for the contact-dependent coupling prior. Hyperparameters, $\gamma_k(c_{ij})$, μ_k and $\mathbf{\Lambda}_k$, have been trained on 500,000 residue pairs per contact class and using *pseudo-likelihood* couplings to estimate the Hessian. Green solid line: Gaussian mixture for contacts. Blue solid line: Gaussian mixture for non-contacts. Black solid line: regularization prior with $\lambda_1 = 0.2L$ with L being protein length and assumed $L = 150$. Light blue dashed line: Gaussian component 0. Dark blue dashed line: Gaussian component 1. Light Green dashed line: Gaussian component 2. **Top Left** One dimensional projection for pair (V,I). **Top Right** One dimensional projection for pair (F,W). **Bottom Left** One dimensional projection for pair (E,R). **Bottom Right** One dimensional projection for pair (E,E).

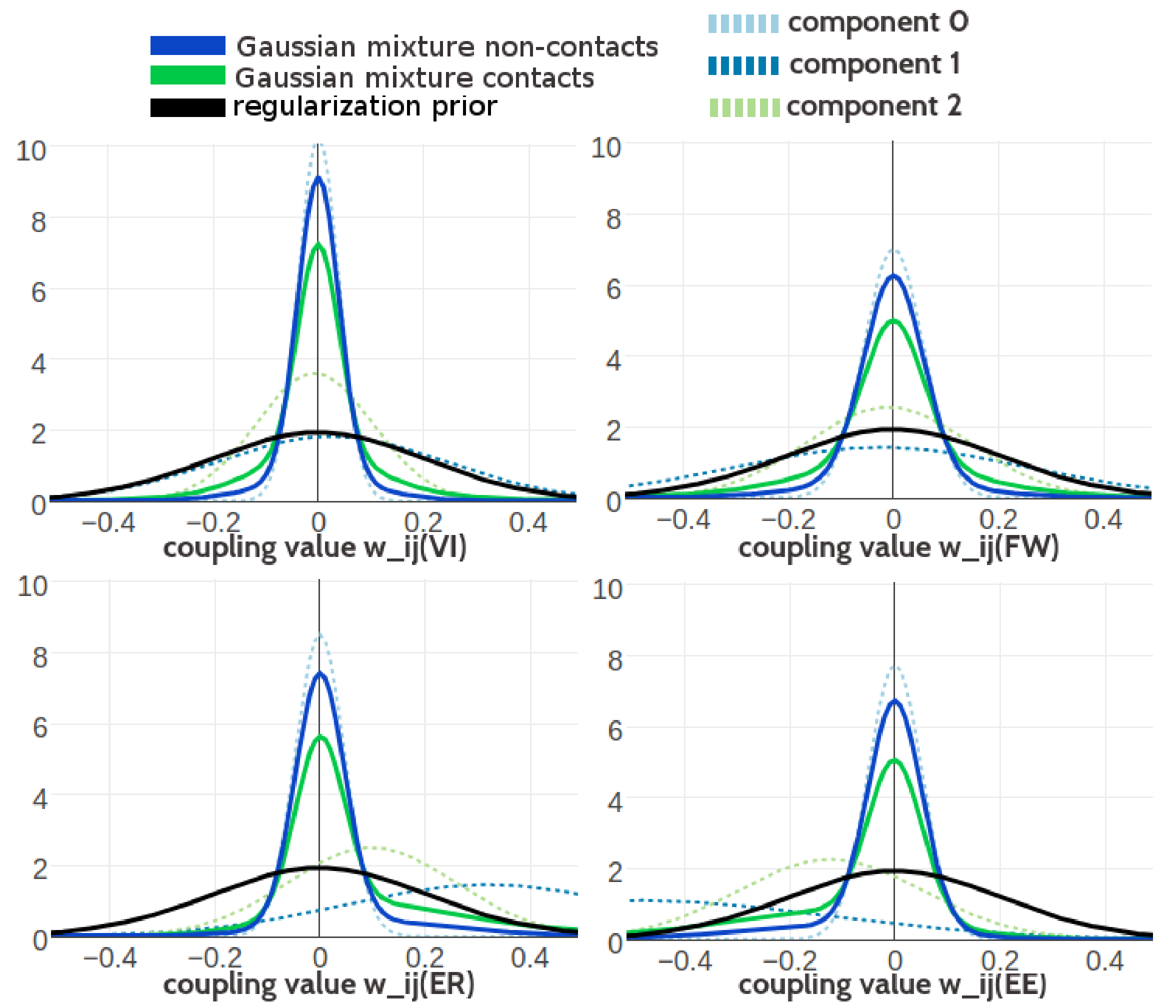


Figure G.8: Visualisation of one-dimensional projections of the three-component Gaussian mixture model for the contact-dependent coupling prior. Hyperparameters, $\gamma_k(c_{ij})$, μ_k and $\mathbf{\Lambda}_k$, have been trained on 100,000 residue pairs per contact class and using *contrastive divergence* couplings to estimate the Hessian. Green solid line: Gaussian mixture for contacts. Blue solid line: Gaussian mixture for non-contacts. Black solid line: regularization prior with $\lambda_1 = 0.2L$ with L being protein length and assumed $L = 150$. Light blue dashed line: Gaussian component 0. Dark blue dashed line: Gaussian component 1. Light green dashed line: Gaussian component 2. **Top Left** One dimensional projection for pair (V,I). **Top Right** One dimensional projection for pair (F,W). **Bottom Left** One dimensional projection for pair (E,R). **Bottom Right** One dimensional projection for pair (E,E).

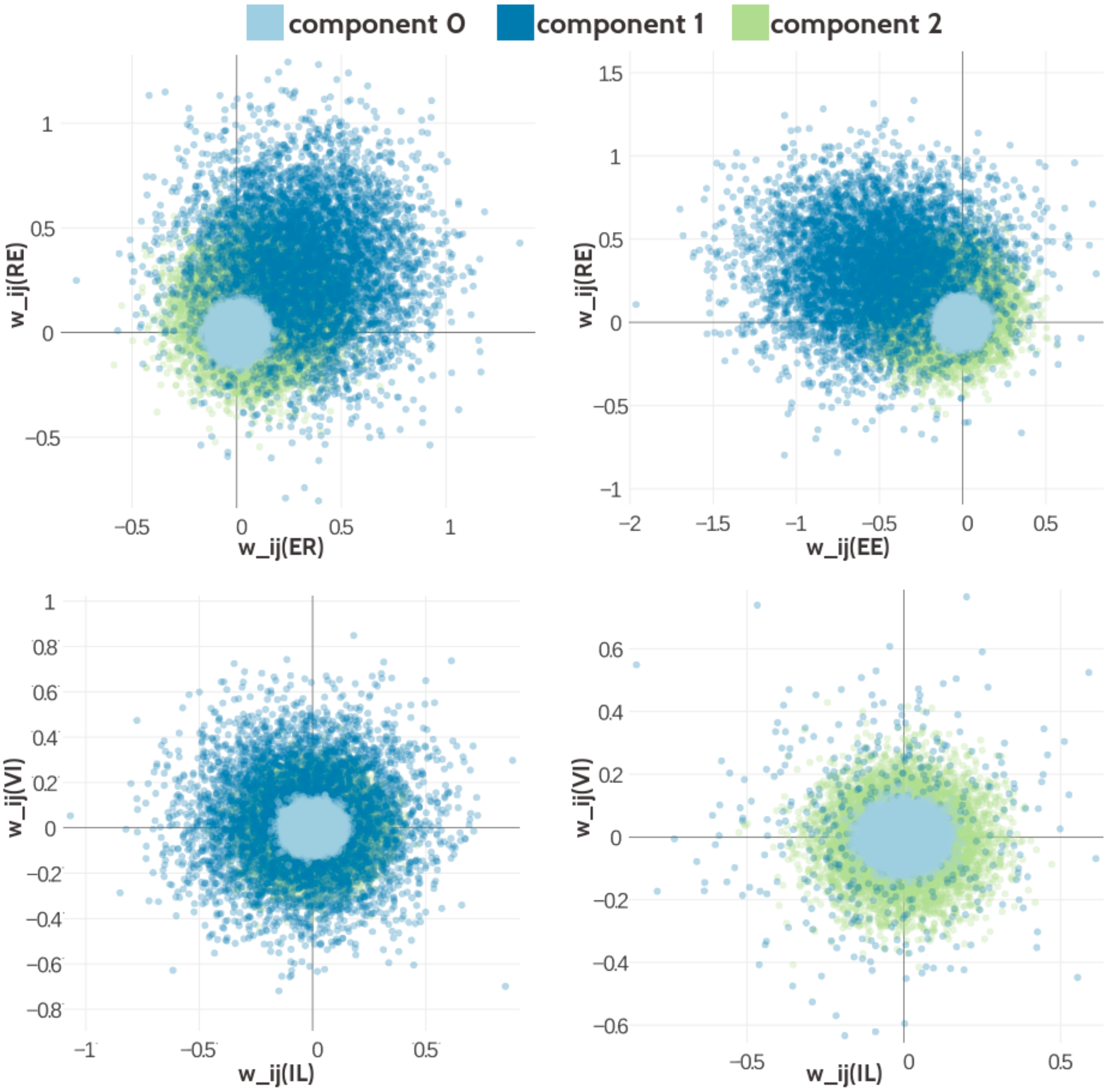


Figure G.9: Visualisation of two-dimensional projections of the three-component Gaussian mixture model for the contact-dependent coupling prior.Hyperparameters, $\gamma_k(c_{ij})$, μ_k and $\mathbf{\Lambda}_k$, have been trained on 300,000 residue pairs per contact class and using *contrastive divergence* couplings to estimate the Hessian. 10,000 paired couplings have been sampled from the Gaussian mixture model. The different colors represent the generating component and color code is specified in the legend. **Top Left** Two-dimensional projection for pairs (E,R) and (R-E) for contacts (using component weight $g_k(1)$). **Top Right** Two- dimensional projection for pairs (E,E) and (R,E) for contacts (using component weight $g_k(1)$). **Bottom Left** Two-dimensional projection for pairs (I,L) and (V,I) for contacts (using component weight $g_k(1)$). **Bottom Right** Two-dimensional projection for pair (I,L) and (V,I) for non-contacts (using component weight $g_k(0)$).

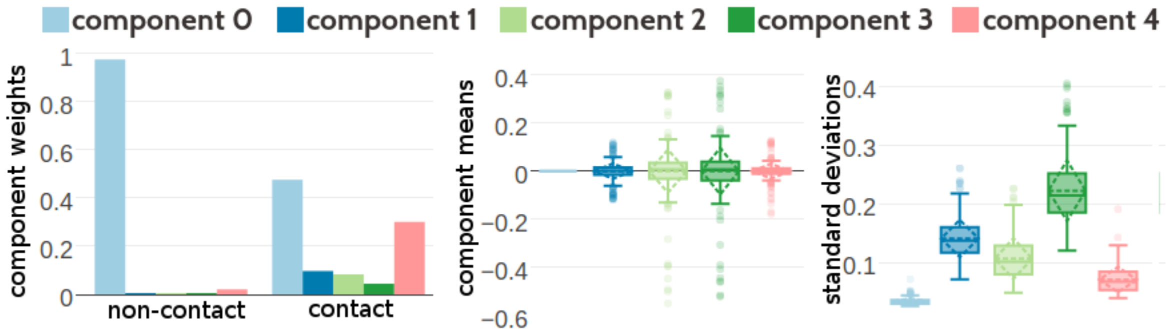


Figure G.10: Statistics for the hyperparameters, $\gamma_k(c_{ij})$, μ_k and $\mathbf{\Lambda}_k$ obtained after 2605 iterations. Trained on 100,000 residue pairs per contact class for a five component Gaussian mixture and using *contrastive divergence* couplings to estimate the Hessian. **Left** Component weights $\gamma_k(c_{ij})$ for residue pairs not in physical contact ($c_{ij} = 0$) and true contacts ($c_{ij} = 1$). **Center** Distribution of the 400 elements in the mean vectors μ_k. **Right** Distribution of the 400 standard deviations corresponding to the square root of the diagonal of $\mathbf{\Lambda}_k^{-1}$.

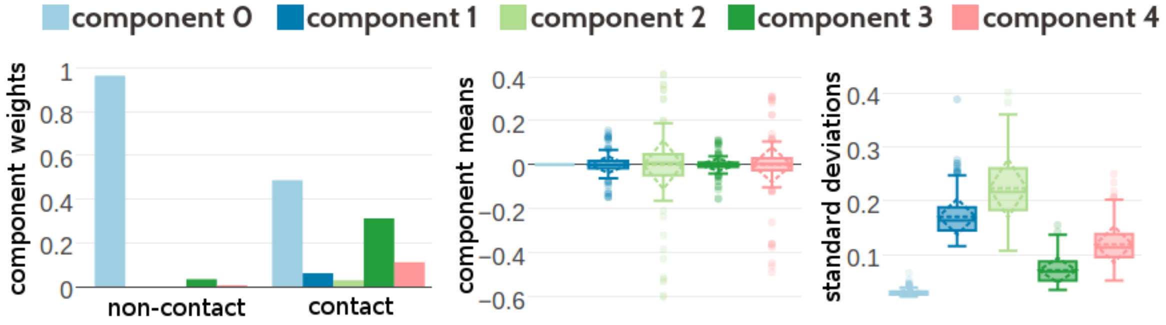


Figure G.11: Statistics for the hyperparameters, $\gamma_k(c_{ij})$, μ_k and $\mathbf{\Lambda}_k$ obtained after 1229 iterations. Trained on 300,000 residue pairs per contact class for a five component Gaussian mixture and using *contrastive divergence* couplings to estimate the Hessian. **Left** Component weights $\gamma_k(c_{ij})$ for residue pairs not in physical contact ($c_{ij} = 0$) and true contacts ($c_{ij} = 1$). **Center** Distribution of the 400 elements in the mean vectors μ_k. **Right** Distribution of the 400 standard deviations corresponding to the square root of the diagonal of $\mathbf{\Lambda}_k^{-1}$.

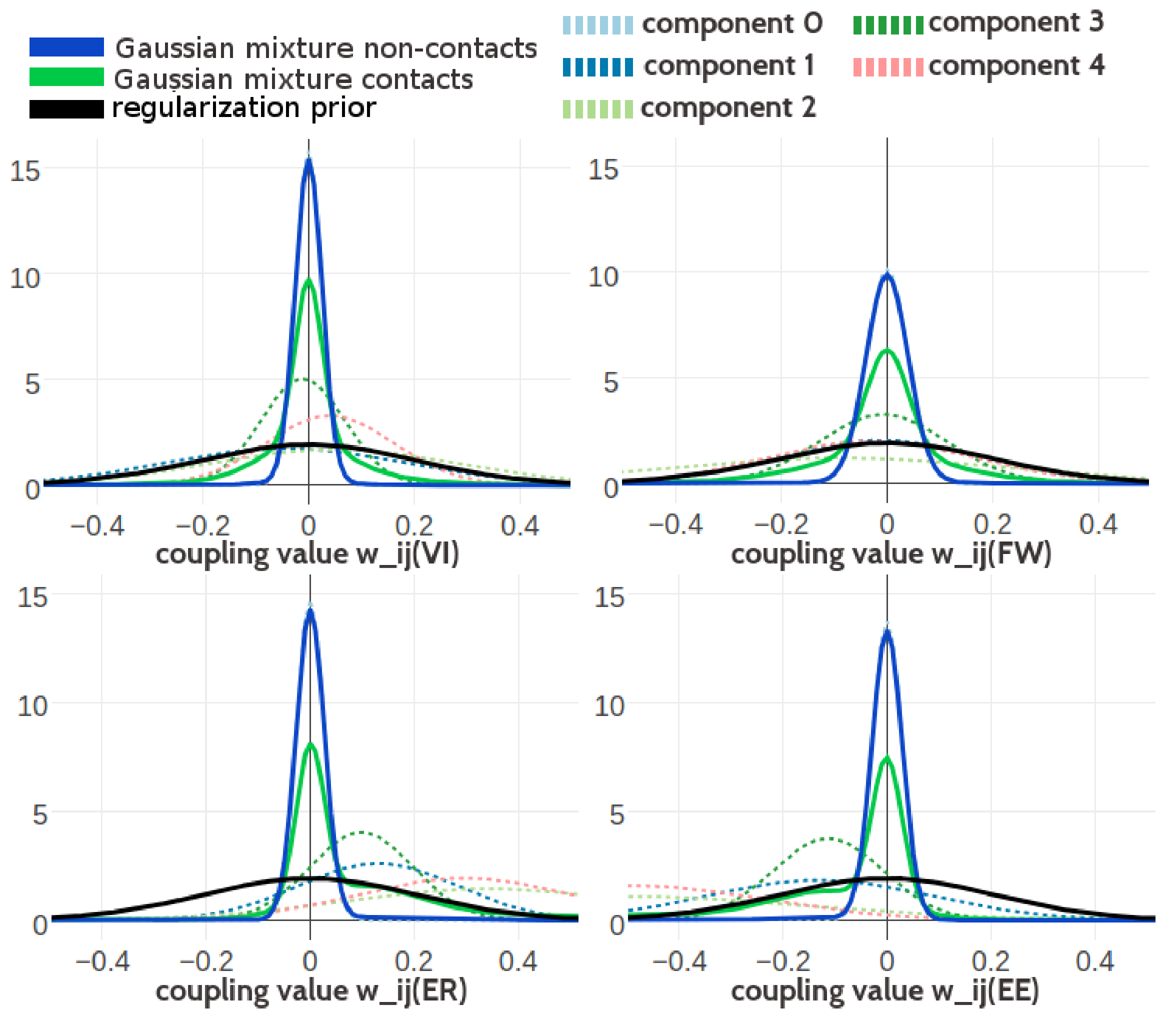


Figure G.12: Visualisation of one-dimensional projections of the five-component Gaussian mixture model for the contact-dependent coupling prior. Hyperparameters, $\gamma_k(c_{ij})$, μ_k and $\mathbf{\Lambda}_k$, have been trained on 300,000 residue pairs per contact class and using *contrastive divergence* couplings to estimate the Hessian. Green solid line: Gaussian mixture for contacts. Blue solid line: Gaussian mixture for non-contacts. Black solid line: regularization prior with $\lambda_1 = 0.2L$ with L being protein length and assumed $L = 150$. Light blue dashed line: Gaussian component 0. Dark blue dashed line: Gaussian component 1. Light green dashed line: Gaussian component 2. Dark green dashed line: Gaussian component 3. Light pink dashed line: Gaussian component 4. **Top Left** One dimensional projection for pair (V,I). **Top Right** One dimensional projection for pair (F,W). **Bottom Left** One dimensional projection for pair (E,R). **Bottom Right** One dimensional projection for pair (E,E).

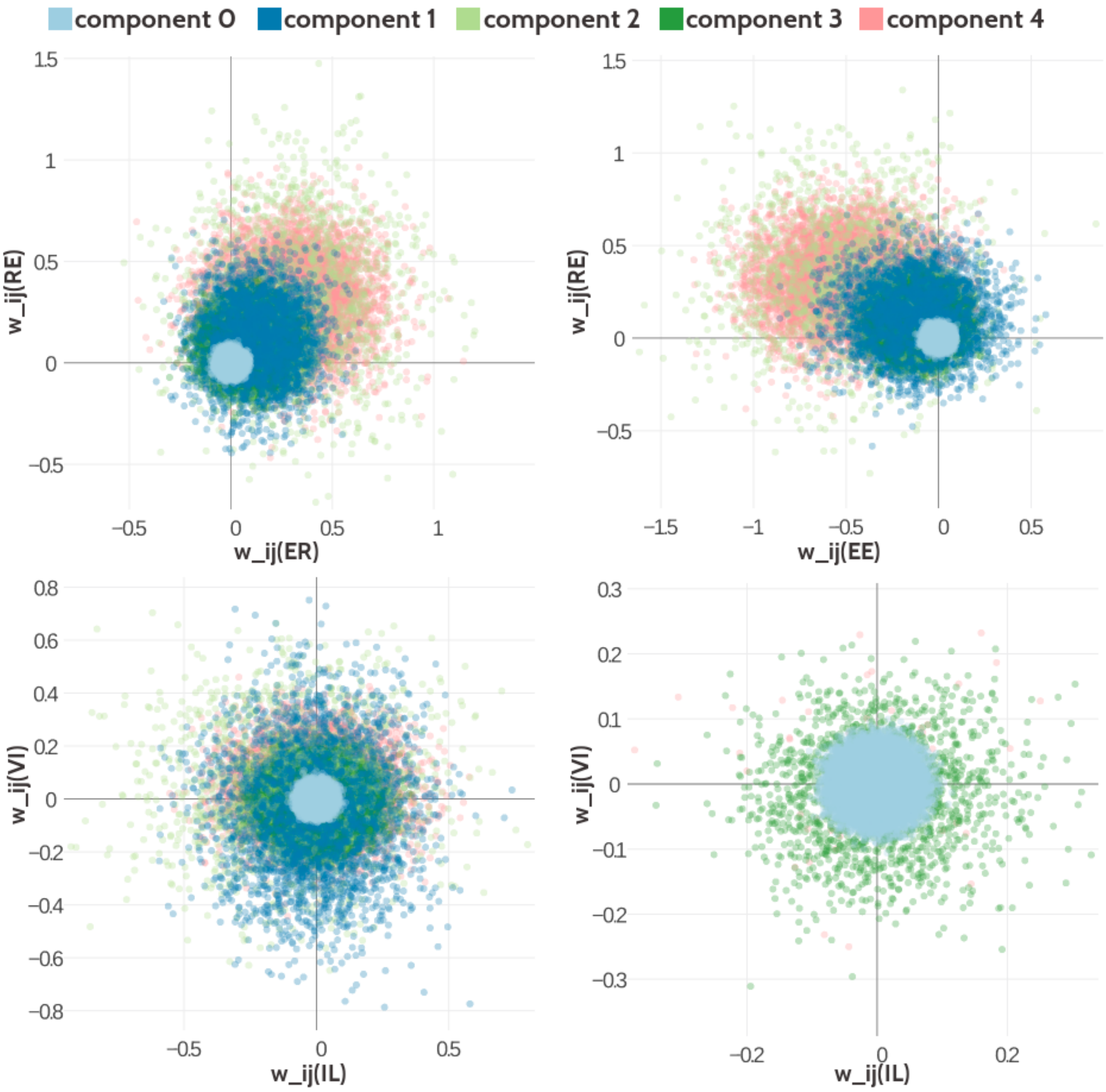


Figure G.13: Visualisation of two-dimensional projections of the five-component Gaussian mixture model for the contact-dependent coupling prior. Hyperparameters, $\gamma_k(c_{ij})$, μ_k and $\mathbf{\Lambda}_k$, have been trained on 300,000 residue pairs per contact class and using *contrastive divergence* couplings to estimate the Hessian. 10,000 values have been samples from the Gaussian mixture model. Light blue: values that have been generated by zero component. Dark blue: values that have been generated by Gaussian component 1. Light green: values that have been generated by Gaussian component 3. Dark green: values that have been generated by Gaussian component 4. Light pink: values that have been generated by Gaussian component 4. **Top Left** Two-dimensional projection for pairs (E,R) and (R-E) for contacts (using component weight $g_k(1)$). **Top Right** Two- dimensional projection for pairs (E,E) and (R,E) for contacts (using component weight $g_k(1)$). **Bottom Left** Two-dimensional projection for pairs (I,L) and (V,I) for contacts (using component weight $g_k(1)$). **Bottom Right** Two-dimensional projection for pair (I,L) and (V,I) for non-contacts (using component weight $g_k(0)$).

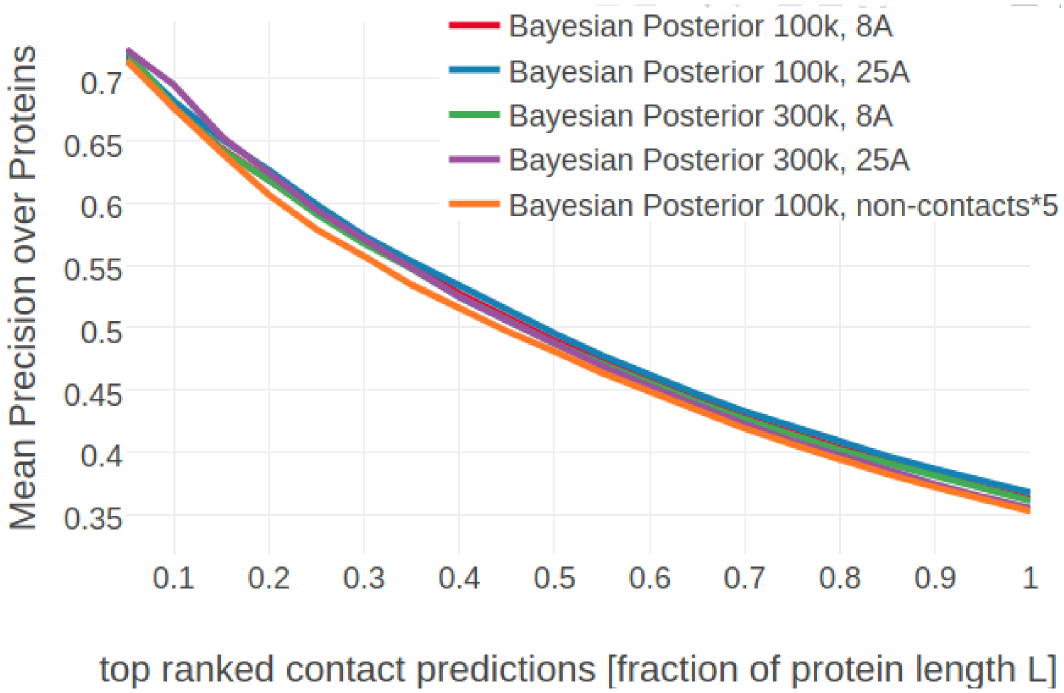


Figure G.14: Mean precision for top ranked contact predictions over 500 proteins. The "Bayesian Posterior" methods compute the posterior probability of contacts with the Bayesian framework employing a three component Gaussian mixture coupling prior based on couplings computed with *pseudo-likelihood*. Hyperparameters for the coupling prior have been trained on different dataset sizes as specified in the legend. Furthermore residue pairs not in physical contact are defined either by $25\mathring{A}$ or a $8\mathring{A}$ C_β distance cutoff that is also specified in the legend. **Bayesian Posterior 100k, non-contacts 5** : Bayesian model trained on 100,000 contacts and 500,000 non-contacts with non-contacts defined by an $25\mathring{A}$ C_β distance threshold.

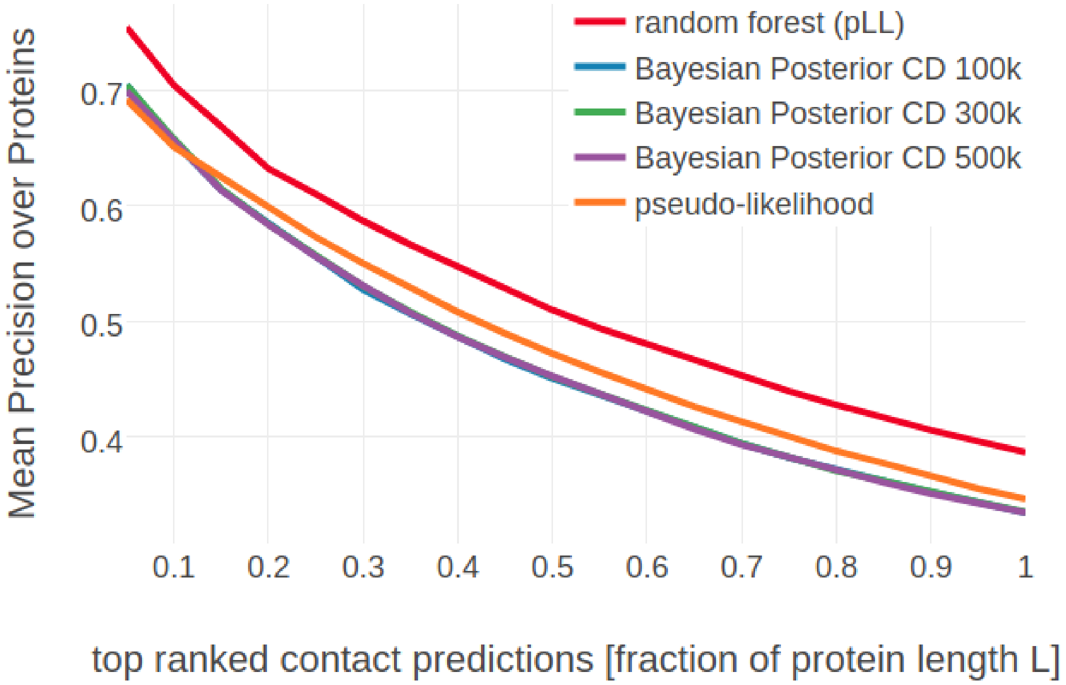


Figure G.15: Mean precision for top ranked contact predictions over 500 proteins. The "Bayesian Posterior" methods compute the posterior probability of contacts with the Bayesian framework employing a three component Gaussian mixture coupling prior based on couplings computed with *contrastive divergence*. Hyperparameters for the coupling prior have been trained on different dataset sizes as specified in the legend. **random forest (pLL)** random forest model trained on sequence features and and additional pseudo-likelihood contact score feature. **Bayesian Posterior 100k**: Bayesian model trained on 100,000 residue pairs per contact class. **Bayesian Posterior 300k**: Bayesian model trained on 300,000 residue pairs per contact class. **Bayesian Posterior 500k**: Bayesian model trained on 500,000 residue pairs per contact class. **pseudo-likelihood**: contact score is computed as APC corrected Frobenius norm of the couplings computed from pseudo-likelihood.

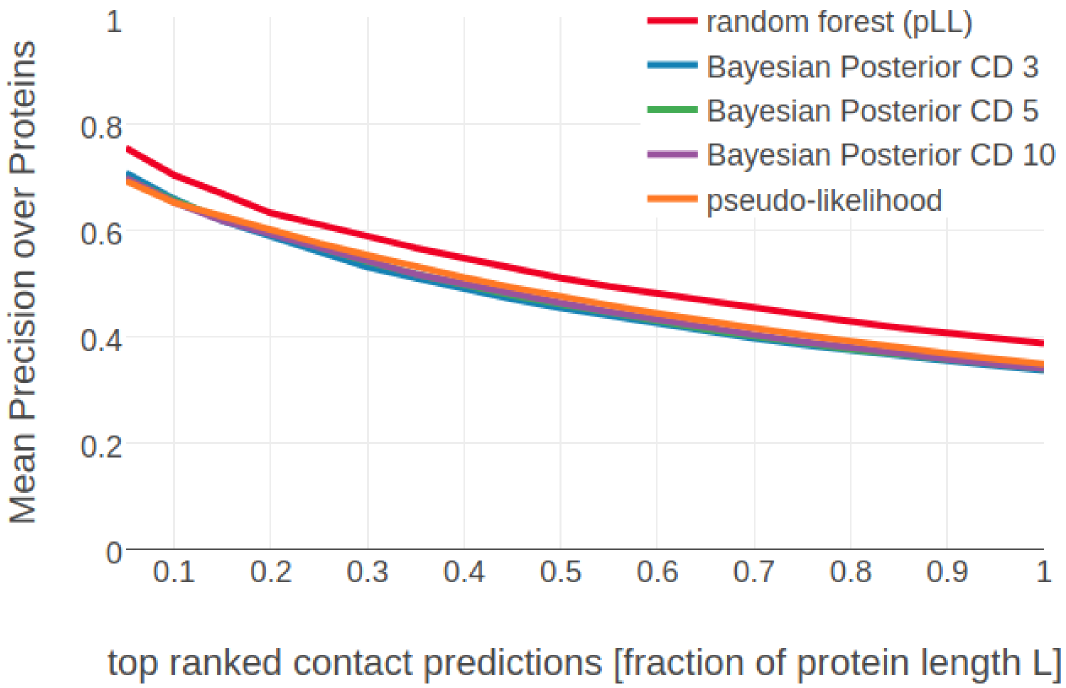


Figure G.16: Mean precision for top ranked contact predictions over 500 proteins. The "Bayesian Posterior" methods compute the posterior probability of contacts with the Bayesian framework employing a Gaussian mixture coupling prior based on couplings computed with *contrastive divergence*. Hyperparameters for the coupling prior have been trained on 100,000 residue pairs per contact class. The number of Gaussian components in the Gaussian mixture model is specified in the legend. **random forest (pLL)** random forest model trained on sequence features and and additional pseudo-likelihood contact score feature. **Bayesian Posterior 3**: Bayesian model utilizing a three component Gaussian mixture. **Bayesian Posterior 5**: Bayesian model utilizing a five component Gaussian mixture. **Bayesian Posterior 10**: Bayesian model utilizing a ten component Gaussian mixture. **pseudo-likelihood**: contact score is computed as APC corrected Frobenius norm of the couplings computed from pseudo-likelihood.

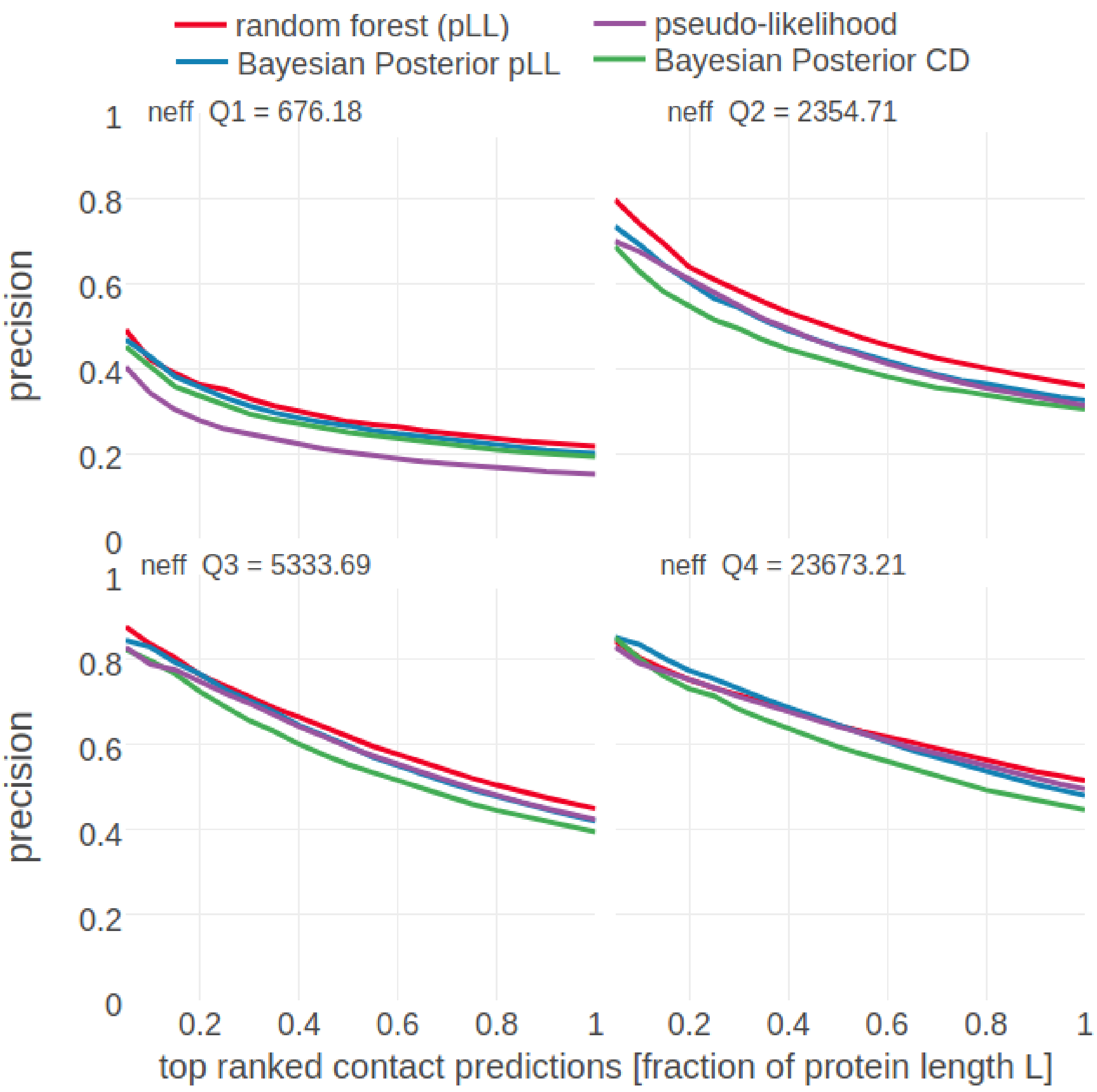


Figure G.17: Mean precision for top ranked contact predictions over 500 proteins splitted into four equally sized subsets with respect to Neff. Upper left: Subset of proteins with Neff < Q1. Upper right: Subset of proteins with Q1 <= Neff < Q2. Lower left: Subset of proteins with Q2 <= Neff < Q3. Lower right: Subset of proteins with Q3 <= Neff < Q4. **random forest (pLL)** random forest model trained on sequence features and and additional pseudo-likelihood contact score feature. **Bayesian Posterior pLL**: Bayesian model computing the posterior probability of contacts with a three component Gaussian mixture coupling prior based on *pseudo-likelihood* couplings. Hyperparameters for the coupling prior have been trained on 300,000 residue pairs per contact class. **Bayesian Posterior CD**: Bayesian model computing the posterior probability of contacts with a three component Gaussian mixture coupling prior based on *contrastive divergence* couplings. Hyperparameters for the coupling prior have been trained on 300,000 residue pairs per contact class. **pseudo-likelihood**: contact score is computed as APC corrected Frobenius norm of the couplings computed from pseudo-likelihood.

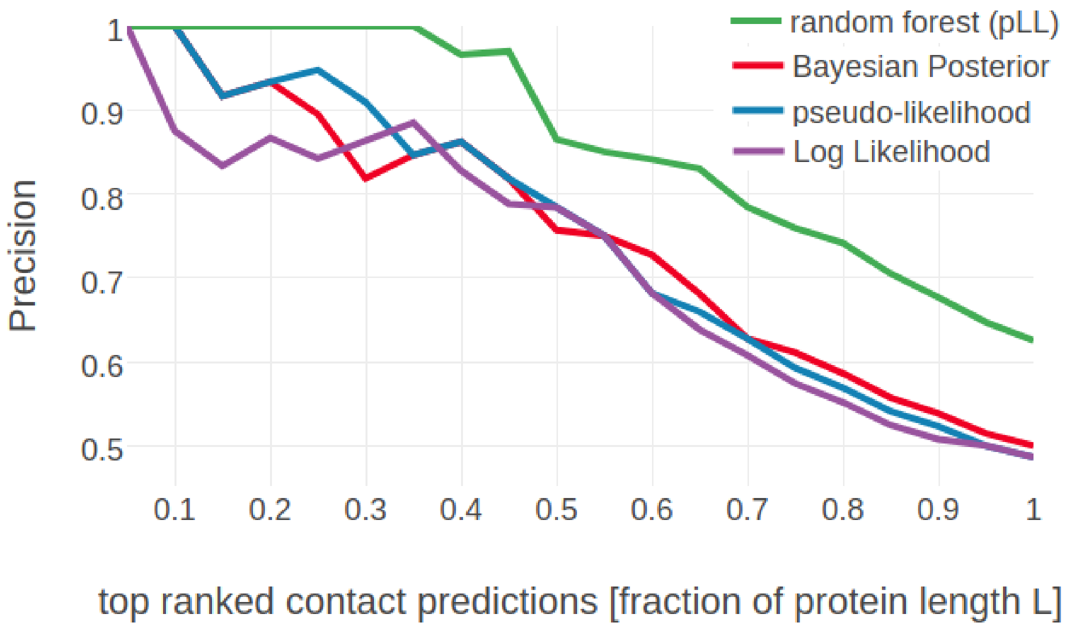


Figure G.18: Precision of top ranked contact predictions for protein 1c75A00. **random forest (pLL)**: random forest model trained on sequence features and and additional pseudo-likelihood contact score feature. **Bayesian Posterior**: Posterior probabilities computed with a three component Gaussian mixture coupling prior based on pseudo-likelihood couplings. **pseudo-likelihood**: Contact scores are computed as the APC corrected Frobenius norm of the pseudo-likelihood couplings. **Log Likelihood**: Log Likelihood of observing a contact as given in eq. (5.31). Coupling prior is modelled as three component Gaussian mixture coupling prior based on pseudo-likelihood couplings.

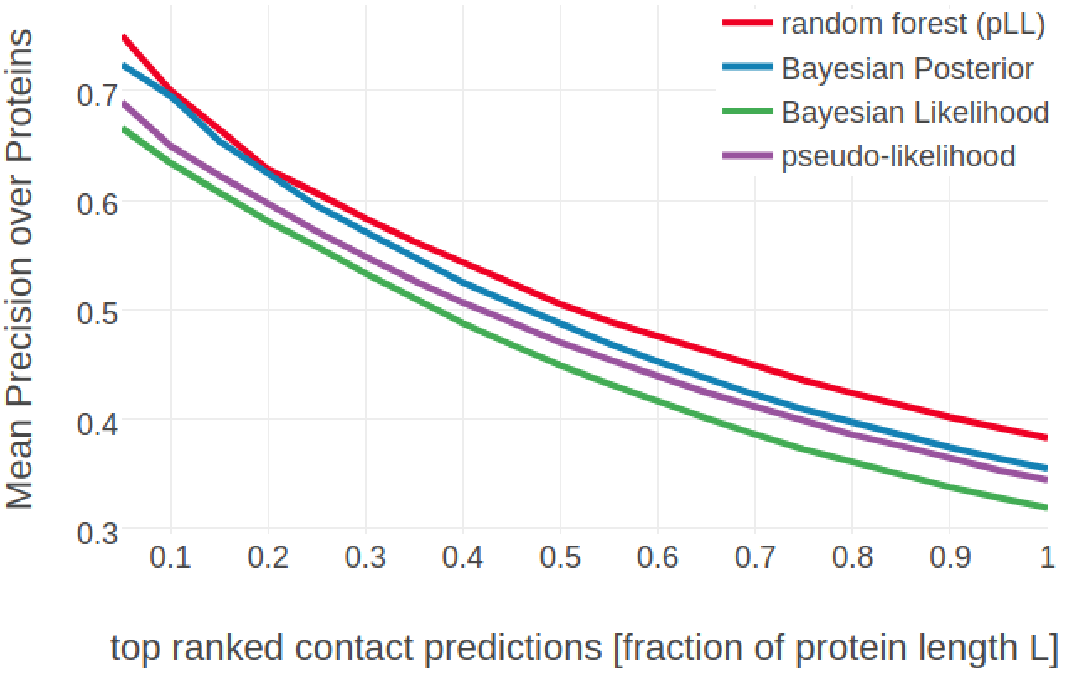


Figure G.19: Mean precision for top ranked contact predictions over 500 proteins. **random forest (pLL)** random forest model trained on sequence features and and additional pseudo-likelihood contact score feature. **random forest (pLL, CD, BayPost)**: random forest model trained on sequence features and and additional contact score features computed from pseudo-likelihood, contrastive divergence and posterior contact probabilities from Bayesian model. **random forest (pLL, CD)**: random forest model trained on sequence features and and additional contact score features computed from pseudo-likelihood and contrastive divergence. **pseudo-likelihood**: contact score is computed as APC corrected Frobenius norm of the couplings computed from pseudo-likelihood. **random forest**: random forest model trained on sequence features.

List of Tables

References

1. Anfinsen, C.B. (1973). Principles that Govern the Folding of Protein Chains. Sci. (80-.). *181*, 223–230., doi: 10.1126/science.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., doi: 10.1006/jmbi.1999.3110.

3. Fraser, P.E. (2014). Prions and prion-like proteins. J. Biol. Chem. *289*, 19839–40., doi: 10.1074/jbc.R114.583492.

4. Samish, I., Bourne, P.E., and Najmanovich, R.J. (2015). Achievements and challenges in structural bioinformatics and computational biophysics. Bioinformatics *31*, 146–150., doi: 10.1093/bioinformatics/btu769.

5. Schwede, T. (2013). Protein modeling: what happened to the "protein structure gap"? Structure *21*, 1531–40., doi: 10.1016/j.str.2013.08.007.

6. Levinthal, C. (1969). How to Fold Graciously. 22–24.

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., doi: 10.1016/0022-2836(80)90373-3.

8. Sander, C., and Schneider, R. (1991). Database of homology-derived protein structures and the structural meaning of sequence alignment. Proteins *9*, 56–68., doi: 10.1002/prot.340090107.

9. Chothia, C., and Lesk, A.M. (1986). The relation between the divergence of sequence and structure in proteins. EMBO J. *5*, 823–6.

10. Martí-Renom, M.A., Stuart, A.C., Fiser, A., Sánchez, R., Melo, F., and ali, A. (2000). Comparative Protein Structure Modeling of Genes and Genomes. Annu. Rev. Biophys. Biomol. Struct. *29*, 291–325., 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., doi: 10.1016/j.compbiolchem.2014.10.001.

12. Berman, H.M. (2000). The Protein Data Bank. Nucleic Acids Res. *28*, 235–242., doi: 10.1093/nar/28.1.235.

13. The UniProt Consortium (2017). UniProt: the universal protein knowledgebase. Nucleic Acids Res. *45*, D158–D169., 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., doi: 10.1016/j.sbi.2008.07.001.

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., doi: 10.1016/j.bbamem.2013.07.010.

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., doi: 10.1016/S0022-2836(02)00470-9.

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., doi: 10.1111/j.1742-4658.2011.08005.x.

18. Billeter, M., Wagner, G., and Wüthrich, K. (2008). Solution NMR structure determination of proteins revisited. J. Biomol. NMR *42*, 155–8., doi: 10.1007/s10858-008-9277-8.

19. Egelman, E.H. (2016). The Current Revolution in Cryo-EM. Biophysj *110*, 1008–1012., doi: 10.1016/j.bpj.2016.02.001.

20. Fernandez-Leiro, R., and Scheres, S.H.W. (2016). Unravelling biological macromolecules with cryo-electron microscopy. Nature *537*, 339–46., doi: 10.1038/nature19948.

21. Human Genome Sequencing Consortium, I. (2004). Finishing the euchromatic sequence of the human genome. Nature *431*, 931–945., doi: 10.1038/nature03001.

22. Reuter, J.A., Spacek, D.V., and Snyder, M.P. (2015). High-throughput sequencing technologies. Mol. Cell *58*, 586–97., doi: 10.1016/j.molcel.2015.05.004.

23. 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., doi: 10.1038/nrg.2016.49.

24. NovaSeq System Specifications | The next era of sequencing starts now.

25. Tringe, S.G., and Rubin, E.M. (2005). Metagenomics: DNA sequencing of environmental samples. Nat. Rev. Genet. *6*, 805–814., doi: 10.1038/nrg1709.

26. Hugenholtz, P., and Tyson, G.W. (2008). Microbiology: Metagenomics. Nature *455*, 481–483., doi: 10.1038/455481a.

27. Wooley, J.C., Godzik, A., and Friedberg, I. (2010). A primer on metagenomics. PLoS Comput. Biol. *6*, e1000667., doi: 10.1371/journal.pcbi.1000667.

28. 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., doi: 10.1038/nature12352.

29. Mukherjee, S., Seshadri, R., Varghese, N.J., Eloe-Fadrosh, E.A., Meier-Kolthoff, J.P., Göker, 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., doi: 10.1038/nbt.3886.

30. Forster, S.C. (2017). Illuminating microbial diversity. Nat. Rev. Microbiol. *15*, 578–578., doi: 10.1038/nrmicro.2017.106.

31. Zerihun, M.B., and Schug, A. (2017). Biomolecular coevolution and its applications: Going from structure prediction toward signaling, epistasis, and function. Biochem. Soc. Trans., BST20170063., doi: 10.1042/BST20170063.

32. Dukka, B.K. (2016). Recent advances in sequence-based protein structure prediction. Brief. Bioinform. *31*, 1–12., doi: 10.1093/bib/bbw070.

33. Ornes, S. (2016). Let the structural symphony begin. Nature *536*, 361–363., doi: 10.1038/536361a.

34. Ward, A.B., Sali, A., and Wilson, I.A. (2013). Biochemistry. Integrative structural

biology. Science *339*, 913–5., doi: 10.1126/science.1228565.

35. 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*.

36. Li, W., Zhang, Y., and Skolnick, J. (2004). Application of sparse NMR restraints to large-scale protein structure prediction. Biophys. J. *87*, 1241–8., doi: 10.1529/biophysj.104.044750.

37. 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., doi: 10.1016/j.sbi.2013.02.008.

38. 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., doi: 10.1016/j.jsb.2010.10.014.

39. 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., doi: 10.1073/pnas.0805923106.

40. 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., doi: 10.1371/journal.pone.0028766.

41. Sadowski, M.I. (2013). Prediction of protein domain boundaries from inverse covariances. Proteins *81*, 253–260., doi: 10.1002/prot.24181.

42. 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.

43. Hopf, T.A., Ingraham, J.B., Poelwijk, F.J., Schärfe, C.P.I., Springer, M., Sander, C., and Marks, D.S. (2017). Mutation effects predicted from sequence co-variation. Nat. Biotechnol. *35*, 128–135., doi: 10.1038/nbt.3769.

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

45. 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.

46. Duarte, J.M., Sathyapriya, R., Stehr, H., Filippis, I., and Lappe, M. (2010). Optimal contact definition for reconstruction of contact maps. BMC Bioinformatics *11*., doi: 10.1186/1471-2105-11-283.

47. Göbel, U., Sander, C., Schneider, R., and Valencia, A. (1994). Correlated mutations and residue contacts in proteins. Proteins *18*, 309–17., doi: 10.1002/prot.340180402.

48. Godzik, A., and Sander, C. (1989). Conservation of residue interactions in a family of Ca-binding proteins. "Protein Eng. Des. Sel. *2*, 589–596., doi: 10.1093/protein/2.8.589.

49. Neher, E. (1994). How frequent are correlated changes in families of protein sequences? Proc. Natl. Acad. Sci. U. S. A. *91*, 98–102.

50. Taylor, W.R., and Hatrick, K. (1994). Compensating changes in protein multiple sequence alignments. "Protein Eng. Des. Sel. *7*, 341–348., doi: 10.1093/protein/7.3.341.

51. Oliveira, L., Paiva, A.C.M., and Vriend, G. (2002). Correlated mutation analyses on very large sequence families. Chembiochem *3*, 1010–7., doi: 10.1002/1439-

7633(20021004)3:10<1010::AID-CBIC1010>3.0.CO;2-T.

52. Shindyalov, I.N., Kolchanov, N.A., 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., doi: 10.1093/protein/7.3.349.

53. Clarke, N.D. (1995). Covariation of residues in the homeodomain sequence family. Protein Sci. *4*, 2269–78., doi: 10.1002/pro.5560041104.

54. 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., doi: 10.1073/pnas.90.15.7176.

55. 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*, 4116–4124., doi: 10.1093/bioinformatics/bti671.

56. 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., doi: 10.1093/oxfordjournals.molbev.a026229.

57. 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.

58. Tillier, E.R., and Lui, T.W. (2003). Using multiple interdependency to separate functional from phylogenetic correlations in protein alignments. Bioinformatics *19*, 750–755., doi: 10.1093/bioinformatics/btg072.

59. 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., doi: 10.1186/1748-7188-2-12.

60. 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., doi: 10.1093/bioinformatics/btm604.

61. Kass, I., and Horovitz, A. (2002). Mapping pathways of allosteric communication in GroEL by analysis of correlated mutations. Proteins *48*, 611–617., doi: 10.1002/prot.10180.

62. 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., doi: 10.1093/protein/gzi029.

63. Lapedes, A., Giraud, B., Liu, L., and Stormo, G. (1999). Correlated mutations in models of protein sequences: phylogenetic and structural effects. *33*, 236–256.

64. Burger, L., and Nimwegen, E. van (2010). Disentangling direct from indirect co-evolution of residues in protein alignments. PLoS Comput. Biol. *6*, e1000633., doi: 10.1371/journal.pcbi.1000633.

65. Juan, D. de, Pazos, F., and Valencia, A. (2013). Emerging methods in protein coevolution. Nat. Rev. Genet. *14*, 249–61., doi: 10.1038/nrg3414.

66. 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., doi: 10.1093/bioinformatics/btr638.

67. 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., doi:

10.1038/msb4100203.

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

69. Wu, S., and Zhang, Y. (2008). A comprehensive assessment of sequence-based and template-based methods for protein contact prediction. Bioinformatics *24*, 924–31.

70. Li, Y., Fang, Y., and Fang, J. (2011). Predicting residue-residue contacts using random forest models. Bioinformatics *27*., doi: 10.1093/bioinformatics/btr579.

71. 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., doi: 10.1371/journal.pone.0026767.

72. Wang, Z., and Xu, J. (2013). Predicting protein contact map using evolutionary and physical constraints by integer programming. Bioinformatics *29*, i266–73.

73. 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.

74. Shackelford, G., and Karplus, K. (2007). Contact prediction using mutual information and neural nets. Proteins *69 Suppl 8*, 159–64., doi: 10.1002/prot.21791.

75. 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., doi: 10.1002/PROT.20160.

76. 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.

77. 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.

78. Eickholt, J., and Cheng, J. (2012). Predicting protein residue–residue contacts using deep networks and boosting. *28*, 3066–3072., doi: 10.1093/bioinformatics/bts598.

79. Di Lena, P., Nagata, K., and Baldi, P. (2012). Deep architectures for protein contact map prediction. Bioinformatics *28*, 2449–57.

80. 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., doi: 10.1186/1472-6807-10-S1-S2.

81. 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., doi: 10.1093/bioinformatics/btu791.

82. 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.

83. 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.

84. Schneider, M., and Brock, O. (2014). Combining Physicochemical and Evolutionary Information for Protein Contact Prediction. PLoS One *9*, e108438.

85. 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.

86. 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., doi: 10.1371/journal.pcbi.1005324.

87. 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., doi: 10.1186/s12859-017-1713-x.

88. 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 naïve Bayes classifiers. Bioinformatics., doi: 10.1093/bioinformatics/btx164.

89. Andreani, J., and Söding, J. (2015). Bbcontacts: Prediction of $$-strand pairing from direct coupling patterns. Bioinformatics *31*, 1729–1737.

90. 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.

91. 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., doi: 10.1002/prot.24943.

92. Jaynes, E.T. (1957). Information Theory and Statistical Mechanics I. Phys. Rev. *106*, 620–630., doi: 10.1103/PhysRev.106.620.

93. Jaynes, E.T. (1957). Information Theory and Statistical Mechanics. II. Phys. Rev. *108*, 171–190., doi: 10.1103/PhysRev.108.171.

94. Wainwright, M.J., and Jordan, M.I. (2007). Graphical Models, Exponential Families, and Variational Inference. Found. Trends Mach. Learn. *1*, 1–305., doi: 10.1561/2200000001.

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

96. 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., doi: 10.1073/pnas.1111471108.

97. Cocco, S., Feinauer, C., Figliuzzi, M., Monasson, R., and Weigt, M. (2017). Inverse Statistical Physics of Protein Sequences: A Key Issues Review. arXiv.

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

99. Ekeberg, M., Lövkvist, 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., doi: 10.1103/PhysRevE.87.012707.

100. 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.

101. Seemayer, S., Gruber, M., and Söding, J. (2014). CCMpred-fast and precise prediction of protein residue-residue contacts from correlated mutations. Bioinformatics, btu500.

102. 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.

103. 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., doi: 10.1073/pnas.1314045110.

104. Lapedes, A., Giraud, B., and Jarzynski, C. (2012). Using Sequence Alignments to Predict Protein Structure and Stability With High Accuracy.

105. 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., doi: 10.1002/prot.22934.

106. Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. Biostatistics *9*, 432–41., doi: 10.1093/biostatistics/kxm045.

107. 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.

108. 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., doi: 10.1371/journal.pone.0092721.

109. Besag, J. (1975). Statistical Analysis of Non-Lattice Data. Source Stat. *24*, 179–195.

110. Gidas, B. (1988). Consistency of maximum likelihood and pseudo-likelihood estimators for Gibbs Distributions. Stoch. Differ. Syst. Stoch. Control Theory Appl.

111. Feinauer, C., Skwark, M.J., Pagnani, A., and Aurell, E. (2014). Improving contact prediction along three dimensions. 19.

112. Zhang, H., Huang, Q., Bei, Z., Wei, Y., and Floudas, C.A. (2016). COMSAT: Residue contact prediction of transmembrane proteins based on support vector machines and mixed integer linear programming. Proteins Struct. Funct. Bioinforma., n/a–n/a., doi: 10.1002/prot.24979.

113. Yu, X., Wu, X., Bermejo, G.A., Brooks, B.R., and Taraska, J.W. (2013). Accurate high-throughput structure mapping and prediction with transition metal ion FRET. Structure *21*, 9–19., doi: 10.1016/j.str.2012.11.013.

114. 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., doi: 10.1038/nmeth.2222.

115. 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.

116. 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., doi: 10.1002/(SICI)1097-0134(19980901)32:4<475::AID-PROT6>3.0.CO;2-F.

117. Aszódi, 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., doi: 10.1006/JMBI.1995.0436.

118. Wu, S., Szilagyi, A., and Zhang, Y. (2011). Improving protein structure prediction using multiple sequence-based contact predictions. Structure *19*, 1182–1191., doi: 10.1016/j.str.2011.05.004.

119. 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., doi: 10.1002/prot.22714.

120. 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., doi: 10.1016/j.cell.2012.04.012.

121. 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.

122. Hopf, T.A., Schärfe, 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.

123. Hayat, S., Sander, C., Marks, D.S., and Elofsson, A. (2015). All-atom 3D structure prediction of transmembrane $$-barrel proteins from sequences. Proc. Natl. Acad. Sci. U. S. A. *112*, 5413–5418.

124. 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.

125. 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.

126. 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.

127. 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.

128. 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 (80-.). *355*, 294–298., doi: 10.1126/science.aah4043.

129. 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., doi: 10.1093/bioinformatics/btw316.

130. 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., doi: 10.1371/journal.pcbi.1004661.

131. 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.

132. 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.

133. 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.

134. Konopka, B.M., Ciombor, M., Kurczynska, M., and Kotulska, M. (2014). Automated Procedure for Contact-Map-Based Protein Structure Reconstruction. J. Membr. Biol., doi: 10.1007/s00232-014-9648-x.

135. Kosciolek, T., and Jones, D.T. (2014). De novo structure prediction of globular proteins

aided by sequence variation-derived contacts. PLoS One *9*, e92197.

136. 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., doi: 10.1073/pnas.1120036109.

137. 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., doi: 10.1371/journal.pcbi.1000584.

138. 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., doi: 10.1021/jp068963t.

139. 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*, 357–367., doi: 10.1109/TCBB.2008.27.

140. Adhikari, B., Bhattacharya, D., Cao, R., and Cheng, J. (2017). Assessing Predicted Contacts for Building Protein Three-Dimensional Models. In Methods mol. biol., pp. 115–126., doi: 10.1007/978-1-4939-6406-2_9.

141. 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., doi: 10.3390/a2010076.

142. Zhang, Y., Kolinski, A., and Skolnick, J. (2003). TOUCHSTONE II: a new approach to ab initio protein structure prediction. Biophys. J. *85*, 1145–64., doi: 10.1016/S0006-3495(03)74551-2.

143. 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., doi: 10.1093/nar/gkw307.

144. Adhikari, B., Bhattacharya, D., Cao, R., and Cheng, J. (2015). CONFOLD: Residue-residue contact-guided ab initio protein folding. Proteins *83*, 1436–49.

145. 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., doi: 10.1093/bioinformatics/btw618.

146. 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., doi: 10.1073/pnas.1611861114.

147. 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., doi: 10.1371/journal.pone.0149166.

148. Gueudré, 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.

149. 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., doi: 10.1073/pnas.1606762113.

150. 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 in-

terfaces by direct coupling analysis. Proc. Natl. Acad. Sci. *114*, E2662——E2671., doi: 10.1073/pnas.1615068114.

151. 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.

152. 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., doi: 10.1016/j.str.2015.10.025.

153. 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., doi: 10.1073/pnas.1508584112.

154. 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., doi: 10.1039/c3cp55275f.

155. 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*, 20533–20538.

156. 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.

157. Nawy, T. (2016). Structural biology: RNA structure from sequence. Nat. Methods *13*, 465–465., doi: 10.1038/nmeth.3892.

158. Weinreb, C., Gross, T., Sander, C., and Marks, D.S. (2015). 3D RNA from evolutionary couplings.

159. 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.

160. Suvarna Vani, K., and Praveen Kumar, K. (2018). Feature Extraction of Protein Contact Maps from Protein 3D-Coordinates. In Inf. commun. technol. adv. intell. syst. comput. (Springer, Singapore), pp. 311–320., doi: 10.1007/978-981-10-5508-9_30.

161. Woniak Paweand Kotulska, M., and Vriend, G. (2017). Correlated mutations distinguish misfolded and properly folded proteins. Bioinformatics *33*, 1497–1504., doi: 10.1093/bioinformatics/btx013.

162. 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., doi: 10.1093/bioinformatics/btw694.

163. 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.

164. 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., doi: 10.1371/journal.pgen.1006508.

165. Gao, C.-Y., Zhou, H.-J., and Aurell, E. (2017). Correlation-Compressed Direct Coupling

Analysis. arXiv.

166. 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., doi: 10.1186/s12864-015-2358-7.

167. 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., doi: 10.1093/molbev/msv211.

168. 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.

169. 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*, 9875–9880., doi: 10.1073/pnas.1409572111.

170. Franceus, J., Verhaeghe, T., and Desmet, T. (2016). Correlated positions in protein evolution and engineering. J. Ind. Microbiol. Biotechnol., 1–9., doi: 10.1007/s10295-016-1811-1.

171. Tian, P., and Best, R.B. (2017). How Many Protein Sequences Fold to a Given Structure? A Coevolutionary Analysis. Biophys. J. *113*, 1719–1730., doi: 10.1016/j.bpj.2017.08.039.

172. 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., doi: 10.1093/bioinformatics/btv592.

173. Monastyrskyy, B., Fidelis, K., Tramontano, A., and Kryshtafovych, A. (2011). Evaluation of residue-residue contact predictions in CASP9. Proteins *79 Suppl 1*, 119–125., doi: 10.1002/prot.23160.

174. 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*, 138–153.

175. Ashkenazy, H., Unger, R., and Kliger, Y. (2009). Optimal data collection for correlated mutation analysis. Proteins *74*, 545–55., doi: 10.1002/prot.22168.

176. Wang, S., Sun, S., and Xu, J. (2017). Analysis of deep learning methods for blind protein contact prediction in CASP12. Proteins Struct. Funct. Bioinforma., doi: 10.1002/prot.25377.

177. Kosciolek, T., and Jones, D.T. (2015). Accurate contact predictions using coevolution techniques and machine learning. Proteins Struct. Funct. Bioinforma., n/a–n/a.

178. 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., doi: 10.1002/0470867302.ch14.

179. 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., doi: 10.1073/pnas.1702664114.

180. Marks, D.S., Hopf, T.A., and Sander, C. (2012). Protein structure prediction from sequence variation. Nat. Biotechnol. *30*, 1072–1080., doi: 10.1038/nbt.2419.

181. 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., doi: 10.1093/bioinformatics/btp135.

182. The UniProt Consortium (2013). Update on activities at the Universal Protein Resource (UniProt) in 2013. Nucleic Acids Res. *41*, D43–7., doi: 10.1093/nar/gks1068.

183. 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., doi: 10.1093/nar/gkv1344.

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

185. Espada, R., Parra, R.G., Mora, T., Walczak, A.M., and Ferreiro, D. (2015). Capturing coevolutionary signals in repeat proteins. BMC Bioinformatics *16*, 207., doi: 10.1186/s12859-015-0648-3.

186. Toth-Petroczy, A., Palmedo, P., Ingraham, J.J., Hopf, T.A.T., Berger, B., Sander, C., Marks, D.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., doi: 10.1016/j.cell.2016.09.010.

187. 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.

188. 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., doi: 10.1093/bioinformatics/btp455.

189. 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., doi: 10.1002/prot.24974.

190. 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*., doi: 10.12688/f1000research.7186.1.

191. Burley, S., and Petsko, G. (1985). Aromatic-aromatic interaction: a mechanism of protein structure stabilization. Science (80-.). *229*, 23–28., doi: 10.1126/science.3892686.

192. 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., doi: 10.1063/1.4966156.

193. 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., doi: 10.1093/nar/gku947.

194. Hinton, G.E. (2002). Training Products of Experts by Minimizing Contrastive Divergence. Neural Comput. *14*, 1771–1800., doi: doi:10.1162/ 089976602760128018.

195. Andrieu, C., Freitas, N. de, Doucet, A., and Jordan, M.I. (2003). An Introduction to MCMC for Machine Learning. Mach. Learn. *50*, 5–43., doi: 10.1023/A:1020281327116.

196. 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., doi: 10.1007/978-3-642-33275-3_2.

197. Bengio, Y., and Delalleau, O. (2009). Justifying and Generalizing Contrastive Divergence. Neural Comput. *21*, 1601–21., doi: 10.1162/neco.2008.11-07-647.

198. Ruder, S. (2017). An overview of gradient descent optimization algorithms. arXiv.

199. Bottou, L. (2012). Stochastic Gradient Descent Tricks. In Neural networks: Tricks of the trade (Springer, Berlin, Heidelberg), pp. 421–436., doi: 10.1007/978-3-642-35289-8_25.

200. Bottou, L. (2010). Large-Scale Machine Learning with Stochastic Gradient Descent. 177–186., doi: 10.1007/978-3-7908-2604-3_16.

201. Schaul, T., Zhang, S., and Lecun, Y. (2013). No More Pesky Learning Rates. arXiv.

202. Zeiler, M.D. (2012). ADADELTA: An Adaptive Learning Rate Method. 6.

203. 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., doi: 10.1007/978-3-642-35289-8_26.

204. Mahsereci, M., Balles, L., Lassner, C., and Hennig, P. (2017). Early Stopping without a Validation Set. arXiv.

205. Carreira-Perpiñán, M. a, and Hinton, G.E. (2005). On Contrastive Divergence Learning. Artif. Intell. Stat. *0*, 17., doi: 10.3389/conf.neuro.10.2009.14.121.

206. Ma, X., and Wang, X. (2016). Average Contrastive Divergence for Training Restricted Boltzmann Machines. Entropy *18*, 35., doi: 10.3390/e18010035.

207. Tieleman, T. (2008). Training Restricted Boltzmann Machines using Approximations to the Likelihood Gradient. Proc. 25th Int. Conf. Mach. Learn. *307*, 7., doi: 10.1145/1390156.1390290.

208. 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., doi: 10.1007/978-3-642-15825-4_26.

209. Hyvärinen, A. (2006). Consistency of pseudolikelihood estimation of fully visible Boltzmann machines.

210. Hyvarinen, A. (2007). Connections Between Score Matching, Contrastive Divergence, and Pseudolikelihood for Continuous-Valued Variables. IEEE Trans. Neural Networks *18*, 1529–1531., doi: 10.1109/TNN.2007.895819.

211. 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.

212. Swersky, K., Chen, B., Marlin, B., and Freitas, N. de (2010). A tutorial on stochastic approximation algorithms for training Restricted Boltzmann Machines and Deep Belief Nets. In 2010 inf. theory appl. work. (IEEE), pp. 1–10., doi: 10.1109/ITA.2010.5454138.

213. Kingma, D., and Ba, J. (2014). Adam: A Method for Stochastic Optimization.

214. Chollet, F. others (2015). Keras.

215. Dieleman, S., Schlüter, J., Raffel, C., Olson, E., Sønderby, S.K., Nouri, D., Maturana, D., Thoma, M., Battenberg, E., and Kelly, J. *et al.* (2015). Lasagne: First release., doi: 10.5281/ZENODO.27878.

216. Ma, J., Wang, S., Wang, Z., and Xu, J. (2015). Protein contact prediction by integrating joint evolutionary coupling analysis and supervised learning. Bioinformatics, btv472.

217. Ho, T.K. (1998). The random subspace method for constructing decision forests. IEEE

Trans. Pattern Anal. Mach. Intell. *20*, 832–844., doi: 10.1109/34.709601.

218. Tin Kam Ho (1995). Random decision forests. In Proc. 3rd int. conf. doc. anal. recognit. (IEEE Comput. Soc. Press), pp. 278–282., doi: 10.1109/ICDAR.1995.598994.

219. Breiman, L. (2001). Random Forests. Mach. Learn. *45*, 5–32., doi: 10.1023/A:1010933404324.

220. 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., doi: 10.1186/1471-2105-10-213.

221. Louppe, G. (2014). Understanding Random Forests: From Theory to Practice.

222. 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., doi: 10.1186/1471-2105-8-25.

223. Bernard, S., Heutte, L., and Adam, S. (2009). Influence of Hyperparameters on Random Forest Accuracy. In (Springer, Berlin, Heidelberg), pp. 171–180., doi: 10.1007/978-3-642-02326-2_18.

224. 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., doi: 10.1002/prot.20098.

225. 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.

226. 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*., doi: 10.1186/1472-6807-9-51.

227. 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., doi: 10.1006/jmbi.1999.3091.

228. 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.

229. Atchley, W.R., Zhao, J., Fernandes, A.D., and Drüke, T. (2005). Solving the protein sequence metric problem. Proc. Natl. Acad. Sci. U. S. A. *102*, 6395–400., doi: 10.1073/pnas.0408677102.

230. Grantham, R. (1974). Amino acid difference formula to help explain protein evolution. Science *185*, 862–4.

231. 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., doi: 10.1093/nar/gkm998.

232. 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.

233. Wimley, W.C., and White, S.H. (1996). Experimentally determined hydrophobicity scale for proteins at membrane interfaces. Nat. Struct. Biol. *3*, 842–8.

234. Kyte, J., and Doolittle, R.F. (1982). A simple method for displaying the hydropathic character of a protein. J. Mol. Biol. *157*, 105–132., doi: 10.1016/0022-2836(82)90515-0.

235. 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., doi: 10.1016/0022-2836(87)90189-6.

236. 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., doi: 10.1006/jmbi.1996.0628.

237. Zhu, H., and Braun, W. (1999). Sequence specificity, statistical potentials, and three-dimensional structure prediction with self-correcting distance geometry calculations of beta-sheet formation in proteins. Protein Sci. *8*, 326–42., doi: 10.1110/ps.8.2.326.

238. 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.

239. Golkov, V., Skwark, M.J., Golkov, A., Dosovitskiy, A., Brox, T., Meiler, J., and Cremers, D. (2016). Protein contact prediction from amino acid co-evolution using convolutional networks for graph-valued images. In Adv. neural inf. process. syst. 29, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, eds. (Curran Associates, Inc.), pp. 4222–4230.

240. 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., doi: 10.1137/0916069.

241. Kukic, P., Mirabello, C., Tradigo, G., Walsh, I., Veltri, P., and Pollastri, G. (2014). Toward an accurate prediction of inter-residue distances in proteins using 2D recursive neural networks. BMC Bioinformatics *15*, 6., doi: 10.1186/1471-2105-15-6.

242. Livingstone, C.D., and Barton, G.J. (1993). Protein sequence alignments: a strategy for the hierarchical analysis of residue conservation. Bioinformatics *9*, 745–756., doi: 10.1093/bioinformatics/9.6.745.