



Detecting contacts in protein folds by solving the inverse Potts problem – a pseudolikelihood approach

Magnus Ekeberg

Supervisor: Erik Aurell
Examiner: Timo Koski
Year: 2012

Master's Thesis in Mathematical Statistics
at the Department of Mathematics
Royal Institute of Technology, Stockholm, Sweden

Typeset in L^AT_EX

Abstract

Spatially proximate amino acid positions in a protein tend to coevolve, so a protein's 3D-structure leaves an echo of correlations in the evolutionary record. Reverse engineering 3D-structures from such correlations is an open problem in structural biology, pursued with increasing vigor as new protein sequences continue to fill the data banks. Within this task lies a statistical stumbling block, rooted in the following: correlation between two amino acid positions can arise from firsthand interaction, but also be network-propagated via intermediate positions; observed correlation is not enough to guarantee proximity. The remedy, and the focus of this thesis, is to mathematically untangle the crisscross of correlations and extract *direct* interactions, which enables a clean depiction of coevolution among the positions.

Recently, analysts have used maximum-entropy modeling to recast this cause-and-effect puzzle as parameter learning in a Potts model (a kind of Markov random field). Unfortunately, a computationally expensive partition function puts this out of reach of straightforward maximum-likelihood estimation. Mean-field approximations have been used, but an arsenal of other approximate schemes exists. In this work, we reimplement an existing contact-detection procedure and replace its mean-field calculations with pseudolikelihood maximization. We then feed both routines real protein data and highlight differences between their respective outputs. Our new program seems to offer a systematic boost in detection accuracy.

Acknowledgements

My work was carried out at the Department of Computational Biology at the Royal Institute of Technology in Stockholm. I wish to thank my supervisor Erik Aurell for encouragement, guidance, and access to the community (which turned out essential to my results). Also, big thanks to Martin Weigt for generously providing data, code, and invaluable help with the biological side of the effort. Finally, thanks to my family and friends for support and for listening to me go on about the subject.

A note on language

Throughout this report, 'we' is written even if the reader is excluded and the reference is to me, the one author.

Contents

1	Introduction	1
1.1	Correlation vs. causation	1
1.2	Potts model for extracting immediate interactions	1
1.3	Potts model for protein structure prediction	2
1.4	Motivation and contribution	3
1.5	Outline	3
2	Protein structure prediction	5
2.1	Proteins and folds	5
2.2	Domain families	5
2.3	Sequence alignments	5
2.4	Structure recovery in domain families	6
3	The Potts model	7
3.1	Foundation and definition	7
3.2	Model properties	9
3.3	The inverse Potts problem	11
4	Method development	14
4.1	Naive mean-field inversion	14
4.2	Pseudolikelihood maximization	14
4.3	Other methods	17
4.4	Regularization	19
4.5	Sequence reweighting	21
4.6	Interaction scores	22
4.7	The finished algorithms	24
4.8	Implementation in MATLAB/C	24
5	Experiments and discussion	26
5.1	Families, crystal structures, and true-positive rates	26
5.2	The real distribution of distances	27
5.3	Set-up and preparatory tests	28
5.4	Main comparison	28
5.5	Other scores for naive mean-field inversion	36
5.6	Comparison using a common score	37
5.7	Running times	44
5.8	Concluding remarks	44
5.9	l_2 - vs. group l_1 -regularization	45
6	Summary and possible future work	48

1 Introduction

In biology, new and refined experimental techniques have brought on rapid increase in data availability during the last few years. Such progress needs to be accompanied by development of appropriate statistical tools to treat the growing data sets. In several branches of systems biology, a key statistical challenge is inferring interaction networks, i.e., determining which parts of a system are communicating with which. An example is neural networks, which is an active area of research (fundamental to the endeavor of understanding the human brain) where studying the underlying web of functional connections between neurons can shed light on their complex collective behavior (see e.g. Schneidman et al. (2006)). Another example is the focus of this project: a central topic in structural biology called *protein structure prediction* (PSP). As we shall see, one can accurately estimate the 3D-structure of a protein by identifying which amino acid positions in its chain have interacted over the evolutionary time scale. PSP is a field indeed undergoing intense growth in the amount of existing data (in the form of amino acid sequences). Also, nowadays much of the experimental output is readily accessible through public data bases such as Pfam (<http://pfam.sanger.ac.uk>, Punta et al. (2012)), which allows a large number of researchers to confront the information quantities.

1.1 Correlation vs. causation

A recurring difficulty when dealing with interacting systems is distinguishing the direct interactions, resulting from adjacent interplay between two units, from interactions mediated via multi-step paths across other elements. Consider, for instance, a system of three politicians A, B, and C, who can vote yes/no on some propositions. Suppose A and B are in cahoots and tend to vote identically, and suppose further that C has decided to always vote against B. Of course, C's votes will then tend to oppose A's. A casual check of voting records might entice someone to claim an explicit relationship between A and C, while the actual effect can be carried entirely via the third-party participant B (A and C may even be unaware of each other). Not to take mere correlation as evidence of immediate interaction is, in addition to logically sound, often critical in real-world situations.

Correlations are in general straightforward to compute from raw data, whereas parameters describing the true causal ties are not. The framework of direct interactions can be thought of as hidden beneath an observable weave of correlations, and untwisting this is a task of inherent intricacy. In PSP, using mathematical means to dispose of the network-mediated correlations can be necessary to get optimal results (Morcos et al., 2011) and can yield improvements worth the computational strain put on the analysis.

1.2 Potts model for extracting immediate interactions

The *Potts model* (Potts, 1952), an instance of what in statistics is referred to as a *Markov random field*, provides a useful platform for tackling this chain-effect issue. It is a many-state generalization of the well-known *Ising model*¹ (Ising, 1925), in which elements have a choice between only two states (as in the

¹Ising models have the same form as *Boltzmann machines* in machine learning.

politician example). Classically, Ising and Potts models are used in statistical physics as depictions of spin systems. Extended to general Markov random fields, however, their use stretches across a variety of statistical topics, including image processing (Cross and Jain, 1983; Geman and Geman, 1984), language analysis (Manning and Schütze, 1999), social network analysis (Kindermann and Snell, 1980), and systems biology (Cocco et al., 2009; Lezon et al., 2006; Weigt et al., 2009).

In a standard statistical-physics setting, (immediate) interaction parameters of a Potts model would usually be given, and the aim would be to calculate correlations (or other averages). In the context of the causation/correlation dilemma, the interest is in reversing this procedure by solving the *inverse Potts problem*, that is, to collect interaction parameters given correlations. Unfortunately, to do this using basic tactics such as maximum-likelihood (ML) estimation is computationally feasible for tiny systems only (see e.g. Welsh (1993)). In applications, one must in general rely on approximate, but in return tractable, ML schemes.

1.3 Potts model for protein structure prediction

Small spatial separation between amino acid positions in a protein encourages co-occurrence of mutations. This stuffs the sequence record with correlations, both direct and indirect ones, which tell of the protein’s structure (this is the thrust of chapter 2). Lapedes et al. (1997) addressed the ambiguities of a purely correlation-based route to protein sequence analysis and considered the use of Potts parameters to instead portray direct interactions. Weigt et al. (2009) successfully executed this, showcasing the accuracy increase achievable by lifting indirect exchanges out of the account. Traditional algorithms typically trusted covariation-like quantities and therefore in a fundamental way missed out on this advantage. For a fuller recounting of PSP advances, methods, and benefits, see for example the introductions and references of Jones et al. (2012) and Balakrishnan et al. (2011).

As mentioned, for most real system sizes the inverse Potts problem cannot be solved by means of everyday ML. Weigt et al. (2009) employed an iterative algorithm called *susceptibility propagation* to approximately recover ML estimates of the direct interaction parameters, but its long convergence times put restrictions on the extensiveness of the analysis. Morcos et al. (2011) gave an implementation using the simpler *naive mean-field inversion* (NMFI) technique, which not only completed the parameter estimation 10^3 – 10^4 times faster than susceptibility propagation (enabling a global analysis using much more of the available protein data) but in fact even showed some better accuracies. Another group, Balakrishnan et al. (2011), took an approach similar to what we do in this thesis (see the next section).

Other ways of deducing direct interactions in PSP, not motivated from the Potts model but somewhat similar in statistical manner, have also been suggested. A fast method utilizing Bayesian networks was provided by Burger and van Nimwegen (2010), and recently Jones et al. (2012) introduced a procedure called *PSICOV* (Protein Sparse Inverse COVariance). While Potts/NMFI and PSICOV both appear capable of outperforming the Bayesian network approach (Morcos et al., 2011; Jones et al., 2012), their relative efficiency currently seems open to investigation (this is not the focus of this thesis, though).

1.4 Motivation and contribution

The results of Morcos et al. (2011) suggest that NMFI yields much of the full power of the Potts model in PSP (see also Marks et al. (2011)). Still, the going knowledge of inverse Potts includes more sophisticated ways of approximate ML than NMFI (we include a small survey in section 4.3), and whether or not these can step up the structure detection capacity is, at present, not clear.

We recently published the superior performance of a candidate based on *pseudolikelihood maximization* (PLM) over NMFI (and over other standard methods) on synthetic data in the Ising model (Aurell and Ekeberg, 2012). In this project, we lift PLM out of the world of artificial data into the real setting of PSP. Specifically, we build a completely PLM-based contact detector and inspect whether or not its detections rival/beat those of NMFI. This hopefully can show what effect choice of Potts inverter has on PSP outputs, and so help guide efforts in the field going forward.

Solving the inverse Potts problem is but one step in the PSP procedure, surrounded by things such as data preprocessing and undersampling corrections. To compare fairly PLM and NMFI, we emulate Morcos et al. (2011) in all such secondary matters. Doing so also allows us to mimic the more biologically motivated decisions required in this work (this manuscript concludes a thesis in statistics, not biology). We essentially replicate the proceedings of their paper (in small scale) except in the Potts inversion where we install PLM in place of NMFI. The inner workings of PLM and NMFI differ quite a bit though, so we also adapt the supporting parts to allow compatibility with PLM.

Pseudolikelihoods for PSP is not a novel thought. Balakrishnan et al. (2011) have devised a version of this idea, but using a venue set up rather different from that of Morcos et al. (2011), regarding for example what portions of the data banks are used. Other measures of prediction accuracy were used, prohibiting direct inspection of PLM’s performance in relation to NMFI. Hence, there is room for a test in a common environment. Also, practical details of our PLM realization differ fairly from those of Balakrishnan et al. (2011).

1.5 Outline

In chapter 2, we debut the ideas of PSP by explaining the biological hypotheses linking protein 3D-structure to correlation among amino acid positions. The chapter is intended for (biologically literate) nonbiologists.

In chapter 3, we draft a derivation of the Potts model and describe the model’s statistical functioning. We also detail the ML approach as brought to bear on the inverse Potts problem (this provides the starting point for approximate routines) and clear up why it is impractical for most system sizes. Later in the chapter, we start the discussion on approximate ML, leaving the details for chapter 4.

In chapter 4, we derive the NMFI algorithm as used by Morcos et al. (2011), including steps separate from the Potts inversion. Alongside this derivation we successively assemble our PLM procedure, making changes and tweaks to these steps as we see fit. We also discuss the software created/used for the experiments in chapter 5.

In chapter 5, we present results from experiments using both NMFI and PLM and discuss how the materials put out are distinct.

In chapter 6, we recount briefly our findings, put in context their implications, and discuss where future priorities could land.

2 Protein structure prediction

In this chapter, we acquaint ourselves with proteins, domain families, and sequence alignments. We then formulate the biological supposition that redresses PSP (or at least the contact detection part) as the problem of inferring couplings in an interacting system. For a more thorough background on these topics, see for example Mount (2004).

2.1 Proteins and folds

Proteins are one of the fundamental building blocks of life and are present in nearly all biological processes. Chemically, they consist of amino acids held together in long chains by peptide bonds. Closely related to a protein's function is its *fold*, which refers to the 3D-structure into which the chain curls. Figure 1 shows an example of protein folding. Experimentally determining the fold, using for example X-ray crystallography, is rather costly and time-consuming. Retrieving just the amino acid sequence (which guides the folding process) is easier, and sequences for more and more proteins are being put out. Interest is therefore high in estimating folds directly from the sequence data.

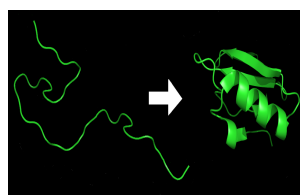


Figure 1: An hypothetical protein before and after folding.

2.2 Domain families

A *domain* is a protein section which tends to fold and, to some extent, evolve as a unit, i.e., separately from other parts of the chain. One domain can be found in many species. Domains in a *domain family* have common evolutionary origin and usually display similar properties. For example, a domain found in humans might have an analogous (or *homologous*) version (from the same family) in a rat (or even in yeast), with only modest changes in sequence, fold, and the function it facilitates.

Arrays with many sequences from the same family can today be conveniently obtained (e.g. from Pfam). The fold is expected similar across a family; changes in function and fold tend to be more moderate than those in sequence. As explained shortly, general claims about the family fold can be made by studying where sequence differences have manifested themselves during evolution.

2.3 Sequence alignments

When looking to quantify the variations within a collection of sequences, an initial step is putting the data into a format where they can be compared. This can be achieved by 'aligning' the sequences, done (roughly speaking) by matching up chain positions where the amino acids are often identical. This is a complicated problem with some rather successful heuristic solution techniques. The evolutionary events responsible for the diversity within a family include removal and insertion of amino acids, so satisfactory alignment requires the

adding of empty spaces, or *gaps*, into the sequences. The resulting data set, called a *multiple sequence alignment* (MSA), can look like in figure 2²³. Note that a letter tends to appear numerous times in a column.

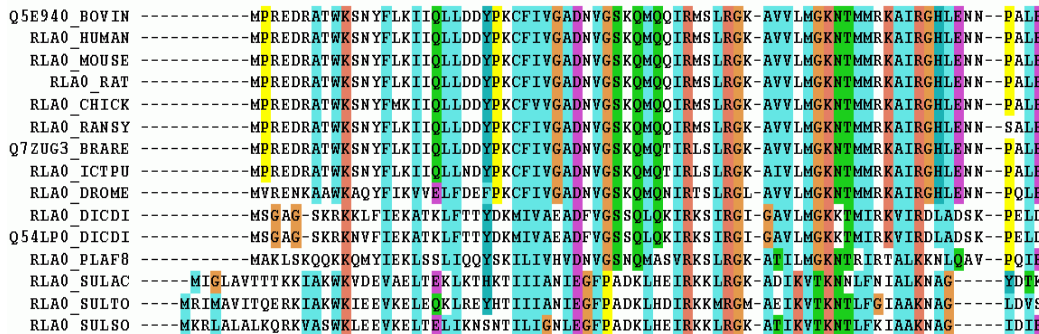


Figure 2: An example MSA. Rows represent sequences and columns represent amino acid positions, also referred to as *sites* or *residues*. Dashes indicate gaps. The colors visualize the conservation of amino acids in each column.

2.4 Structure recovery in domain families

Assembling sequences in this manner allows one to more easily probe for statistical dependencies in the data. To predict a family’s fold, the following hypothesis is used: a statistical tie between two columns in the MSA is likely evidence of spatial proximity between the corresponding positions. Figure 3 illustrates a simple example; amino acid S in one place always appears paired with an H in another, and similar for F and W. These two positions seem to have escorted each other through the space of possible sequences as evolution progressed. Since they are not particularly close in the sequence order, we can suspect the chain folds in on itself to form a *contact*, enabling the interaction.

For many domain families, Pfam supplies sequences from thousands of members, so MSAs like the ones in figures 2 and 3 can have thousands of rows, and the general idea demonstrated in figure 3 can be taken to a larger scale. By identifying a whole web of these correlations, many possible contacts can be suggested, providing the first step toward building an estimate of the 3D-structure of the family (Göbel et al., 1994). This is where sorting out network-induced correlations becomes necessary, to pinpoint pairs of *directly* interacting positions only and not, for example, pairs of distant positions which just happen to have many intermediate contacts between them.

²Source: Wikipedia, accessed on 15/2-2012. Document distributed under GNU Free Documentation License Version 1.3, 3 November 2008 Copyright (C) 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. Original uploader was Miguel Andrade.

³Amino acids are often represented by letters, for instance T=threonine, K=lysine, and M=methionine.

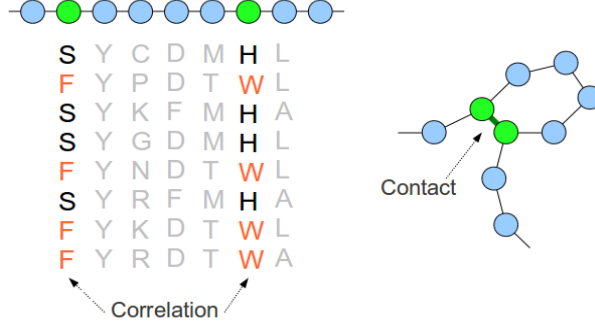


Figure 3: On the left: A made-up MSA which hints at geometrical proximity between two positions. On the right: A hypothesized corresponding chain conformation.

3 The Potts model

We start this chapter by motivating and formalizing the Potts model in the context of PSP. We then show how to learn the direct interaction parameters in typical ML fashion. As we have mentioned several times now, it is fundamental to this thesis that such straightforward parameter retrieval is out of reach computationally for basically all domain lengths. We straighten out why this is and lead into a segment setting up the topic of approximate ML schemes.

3.1 Foundation and definition

Let $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_N)$ represent the amino acid sequence of a domain with length N . For notational convenience, we replace the letters that commonly represent amino acids with consecutive integers starting at one. Hence, each σ_i takes on values in $\{1, 2, \dots, q\}$. q is taken as 21: one state for each of the 20 naturally occurring amino acids and one additional state to represent gaps. Thus, an MSA with B aligned sequences from a domain family can be written as an integer array,

$$\{\sigma^{(b)}\}_{b=1}^B = \begin{pmatrix} \sigma_1^{(1)} & \sigma_2^{(1)} & \dots & \sigma_N^{(1)} \\ \sigma_1^{(2)} & \sigma_2^{(2)} & \dots & \sigma_N^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_1^{(B)} & \sigma_2^{(B)} & \dots & \sigma_N^{(B)} \end{pmatrix},$$

with one row per sequence and one column per chain position. We will also refer to the sequences as *samples* and to the positions as *nodes* (thought of as parts of an interaction graph). The objective is to identify direct statistical couplings among columns in this array, exposing positions which are in each other's midst. Given an MSA, the empirical individual and pairwise *frequencies*

can be calculated as⁴

$$\begin{aligned}\mathbf{f}_i(k) &= \frac{1}{B} \sum_{b=1}^B I[\sigma_i^{(b)} = k], \\ \mathbf{f}_{ij}(k, l) &= \frac{1}{B} \sum_{b=1}^B I[\sigma_i^{(b)} = k] I[\sigma_j^{(b)} = l].\end{aligned}\quad (1)$$

I is an indicator function giving one if the statement in the brackets is true and zero otherwise. $\mathbf{f}_i(k)$ becomes the fraction of sequences for which position i has amino acid k . Similarly, $\mathbf{f}_{ij}(k, l)$ becomes the fraction of sequences in which the position pair (i, j) has the amino acid combo (k, l) . Correlations can now be formed as

$$\mathbf{c}_{ij}(k, l) = \mathbf{f}_{ij}(k, l) - \mathbf{f}_i(k) \mathbf{f}_j(l). \quad (2)$$

A maximum-entropy model

The Potts model is obtained by seeking a probabilistic model $P(\boldsymbol{\sigma})$ which can reproduce the empirically observed $\mathbf{f}_i(k)$ and $\mathbf{f}_{ij}(k, l)$ but otherwise is as general as possible. The frequency demands on $P(\boldsymbol{\sigma})$ can be formulated as

$$\begin{aligned}P(\sigma_i = k) &= \sum_{\substack{\boldsymbol{\sigma} \\ \sigma_i = k}} P(\boldsymbol{\sigma}) = \mathbf{f}_i(k), \\ P(\sigma_i = k, \sigma_j = l) &= \sum_{\substack{\boldsymbol{\sigma} \\ \sigma_j = l \\ \sigma_i = k}} P(\boldsymbol{\sigma}) = \mathbf{f}_{ij}(k, l),\end{aligned}\quad (3)$$

and, in keeping with the *maximum-entropy principle*, the desired most-general model is had by maximizing the entropy $S = -\sum_{\boldsymbol{\sigma}} P(\boldsymbol{\sigma}) \ln P(\boldsymbol{\sigma})$ while adhering to these. Maximization of S under (3) can be carried out through the introduction of Lagrange multipliers, giving, after some straightforward calculations, the Potts distribution

$$P(\boldsymbol{\sigma}) = \frac{1}{Z} \exp \left(\sum_{i=1}^N \mathbf{h}_i(\sigma_i) + \sum_{i=1}^{N-1} \sum_{j=i+1}^N \mathbf{J}_{ij}(\sigma_i, \sigma_j) \right), \quad (4)$$

in which multipliers remain as parameters to be fitted to data. This $P(\boldsymbol{\sigma})$, in an information-theory sense, makes minimal assumption about the world while still capable of staying true to our observed averages. Z is a normalizing constant making sure the total probability is one by summing over all possible states $\boldsymbol{\sigma}$,

$$Z = \sum_{\boldsymbol{\sigma}} \exp \left(\sum_{i=1}^N \mathbf{h}_i(\sigma_i) + \sum_{i=1}^{N-1} \sum_{j=i+1}^N \mathbf{J}_{ij}(\sigma_i, \sigma_j) \right). \quad (5)$$

⁴Unless stated otherwise, node indexes are assumed to take on values $1 \leq i \leq N$, pair indexes run through $1 \leq i < j \leq N$, and states take on $1 \leq k, l \leq q$.

In the Potts model (4), each node i has associated with it a vector of *fields* $\mathbf{h}_i = (h_{i,1}, h_{i,2}, \dots, h_{i,q})^T$ and each pair (i, j) a matrix of *couplings*

$$\mathbf{J}_{ij} = \begin{pmatrix} J_{ij,11} & J_{ij,12} & \cdots & J_{ij,1q} \\ J_{ij,21} & J_{ij,22} & \cdots & J_{ij,2q} \\ \vdots & \vdots & \ddots & \vdots \\ J_{ij,q1} & J_{ij,q2} & \cdots & J_{ij,qq} \end{pmatrix}.$$

Note that the probability of a sequence σ is calculated by picking out the entries from \mathbf{h}_i and \mathbf{J}_{ij} corresponding to the amino acids in positions i and j .

A Potts model in statistical physics is usually not this general. The coupling matrices are sometimes reduced as much as $\mathbf{J}_{ij} = \mathbf{J}\mathbf{I}_q$, requiring just one coupling parameter for the entire system. Still, we will by 'Potts model' mean (4).

3.2 Model properties

Interpreting the fields and couplings

The fields and couplings describe inclinations of positions to carry certain amino acids. Specifically, a large $\mathbf{h}_i(k)$ is a bias of position i toward preferring amino acid k , and a large $\mathbf{J}_{ij}(k, l)$ translates into a desire for positions i and j to jointly carry amino acids k and l . We can think of $\mathbf{h}_i(k)$ and $\mathbf{J}_{ij}(k, l)$ as the immediate-effect quantities we have talked about, contrasted to the observables $\mathbf{f}_i(k)$ and $\mathbf{c}_{ij}(k, l)$, which include network propagation of effects. Sensibly then, the fields and couplings, *not* the correlations, ought to be used to sharply report the interactive characteristics of the system. Hence, the whole game here is retrieving the set $\{\mathbf{h}, \mathbf{J}\}$ from $\{\sigma^{(b)}\}_{b=1}^B$, i.e., reverse designing the model parameters from observations of the system (the inverse Potts problem).

The number of free parameters

The pairs (i, j) and (j, i) are considered the same, and no pairs of the type (i, i) are included. Thus, the number of pairs equals the number of ways one can choose two elements from a collection of N without replacement or ordering: $N(N-1)/2$.

Because there are N nodes each with a field vector of length q and $N(N-1)/2$ node pairs each with a coupling matrix of size q^2 , the total number of parameters is $Nq + \frac{N(N-1)}{2}q^2$. But, it turns out that the model as it stands is over-parameterized, in the sense that distinct parameter sets can describe the same probability distribution. As a consequence, the problem of retrieving $\{\mathbf{h}, \mathbf{J}\}$ from a piece of data (an MSA) has multiple solutions. This can be bothersome when trying to get a learning algorithm to converge, and it also makes reproducing results harder. One would thus ideally dispose of the unneeded variables before attempting inference. Indeed, superfluous quantities are analogously present among the frequencies (1). Note for instance that $\mathbf{f}_i(q)$ is implied given $\mathbf{f}_i(1), \mathbf{f}_i(2), \dots, \mathbf{f}_i(q-1)$ since $\sum_{k=1}^q \mathbf{f}_i(k) = 1$. When eliminating all redundancies from the derivation, the number of free parameters in the model falls out as $N(q-1) + \frac{N(N-1)}{2}(q-1)^2$ (Weigt et al., 2009; Morcos et al., 2011). A way to account for this dimensional excess is to impose constraints on the

parameters, for example by setting

$$\mathbf{J}_{ij}(q, l) = \mathbf{J}_{ij}(k, q) = \mathbf{h}_i(q) = 0, \quad (6)$$

for all i, j, k , and l (as was done by Morcos et al. (2011)), which would mean measuring all biases and interactions with the last state as a reference level. This would make the solution to the inverse Potts problem unique. We sometimes fix parameters like this and sometimes use the full representation, for reasons explained in section 4.4.

Relation to the Ising model

As touched on in the introduction, the Potts model reduces to the Ising model when $q = 2$. Using the full parameterization, \mathbf{h}_i and \mathbf{J}_{ij} in the $q = 2$ case would be of dimensions 2 and 2×2 respectively. The typical Ising model formulation grants each node just one field parameter h_i and each node pair just one coupling parameter J_{ij} , a consequence of the above discussed need to represent the distribution uniquely. The parameter constraints commonly used in the Ising model is not (6) (although some of the literature uses it), but rather

$$\sum_{s=1}^q \mathbf{J}_{ij}(k, s) = \sum_{s=1}^q \mathbf{J}_{ij}(s, l) = \sum_{s=1}^q \mathbf{h}_i(s) = 0, \quad (7)$$

for all i, j, k , and l , i.e., the sum across any column or row in any coupling matrix, and the sum of any field vector, should be zero. By using these constraints and letting the variables σ_i take on values -1 and 1 instead of 1 and 2 , one gets the Ising distribution presented in the classical form

$$P(\boldsymbol{\sigma}) = \frac{1}{Z} \exp \left(\sum_{i=1}^N h_i \sigma_i + \sum_{i=1}^{N-1} \sum_{j=i+1}^N J_{ij} \sigma_i \sigma_j \right). \quad (8)$$

Sometimes an inverse temperature is also included, in which case the exponential can read $\beta \sum_{i=1}^N h_i \sigma_i + \beta \sum_{i=1}^{N-1} \sum_{j=i+1}^N J_{ij} \sigma_i \sigma_j$ where $\beta = \frac{1}{T}$ (see e.g. Aurell and Ekeberg (2012)).

So far, most research on the inverse Potts problem has been carried out for the $q = 2$ special case, termed the *inverse Ising problem*. However, reasonings for $q = 2$ can often carry over in a direct manner to $q > 2$. Formulas and derivations present neater when $q = 2$, so, for readability, we will take some later discussions in the setting of (8).

An accurate depiction of domain families?

Let us reconnect to PSP. When adopting a Potts model in this way, one is assuming that the sequences from a domain family are independent samples drawn at random, according to (4), from the q^N -dimensional space of all possible sequences. This model choice can of course be questioned.

First: why embed only two-node interplay in the model? Well, in correlated systems in nature, pairwise behavior often accounts for much, if not most, of the activity. Also, estimating something like third-order interaction variables would require dramatically larger data sets.

Second: are the sequences/samples really independent? This seems a stretch, especially when noting that many MSAs in Pfam have an unproportional amount of sequences nearly or exactly identical. Such critique is certainly justified, and Morcos et al. (2011) combated the issue by *reweighting*, a procedure we describe in section 4.5.

3.3 The inverse Potts problem

Maximum-likelihood estimation

Given a set of independent samples $\{\boldsymbol{\sigma}^{(b)}\}_{b=1}^B$ from (4), the ordinary statistical approach to inferring $\{\mathbf{h}, \mathbf{J}\}$ would be to let the estimates maximize the likelihood, often by minimizing the (rescaled) negative log-likelihood function

$$nll = -\frac{1}{B} \sum_{b=1}^B \ln P(\boldsymbol{\sigma}^{(b)}). \quad (9)$$

For the Potts model (4), this becomes

$$\begin{aligned} nll(\mathbf{h}, \mathbf{J}) &= -\frac{1}{B} \sum_{b=1}^B \ln \left[\frac{1}{Z} \exp \left(\sum_{i=1}^N \mathbf{h}_i(\sigma_i^{(b)}) + \sum_{i=1}^{N-1} \sum_{j=i+1}^N \mathbf{J}_{ij}(\sigma_i^{(b)}, \sigma_j^{(b)}) \right) \right] = \\ &= \ln Z - \frac{1}{B} \sum_{b=1}^B \left(\sum_{i=1}^N \mathbf{h}_i(\sigma_i^{(b)}) + \sum_{i=1}^{N-1} \sum_{j=i+1}^N \mathbf{J}_{ij}(\sigma_i^{(b)}, \sigma_j^{(b)}) \right) = \\ &= \ln Z - \underbrace{\sum_{i=1}^N \frac{1}{B} \sum_{b=1}^B \mathbf{h}_i(\sigma_i^{(b)})}_{\sum_{k=1}^q \mathbf{f}_i(k) \mathbf{h}_i(k)} - \underbrace{\sum_{i=1}^{N-1} \sum_{j=i+1}^N \frac{1}{B} \sum_{b=1}^B \mathbf{J}_{ij}(\sigma_i^{(b)}, \sigma_j^{(b)})}_{\sum_{k=1}^q \sum_{l=1}^q \mathbf{f}_{ij}(k, l) \mathbf{J}_{ij}(k, l)}. \end{aligned}$$

As indicated by the underbraces, the fraction of times the entries $\mathbf{h}_i(k)$ and $\mathbf{J}_{ij}(k, l)$ are picked across the B -sums can be represented by the frequencies, giving

$$nll(\mathbf{h}, \mathbf{J}) = \ln Z - \sum_{i=1}^N \sum_{k=1}^q \mathbf{f}_i(k) \mathbf{h}_i(k) - \sum_{i=1}^{N-1} \sum_{j=i+1}^N \sum_{k=1}^q \sum_{l=1}^q \mathbf{f}_{ij}(k, l) \mathbf{J}_{ij}(k, l). \quad (10)$$

The normalization constant Z of course depends on all the parameters, $Z = Z(\mathbf{h}, \mathbf{J})$. The ML estimates are obtained as

$$\{\mathbf{h}^{ML}, \mathbf{J}^{ML}\} = \underset{\{\mathbf{h}, \mathbf{J}\}}{\operatorname{argmin}} \{nll(\mathbf{h}, \mathbf{J})\}. \quad (11)$$

The negative log-likelihood objective nll is differentiable, so minimizing it means looking for a stationary point, i.e., a point at which $\partial_{\mathbf{h}_i(k)} nll = 0$ and $\partial_{\mathbf{J}_{ij}(k, l)} nll = 0$. Hence, ML estimates will satisfy

$$\begin{aligned} \partial_{\mathbf{h}_i(k)} \ln Z - \mathbf{f}_i(k) &= 0, \\ \partial_{\mathbf{J}_{ij}(k, l)} \ln Z - \mathbf{f}_{ij}(k, l) &= 0. \end{aligned} \quad (12)$$

By looking at the structure of Z (see (5)) one can note that

$$\begin{aligned}\partial_{\mathbf{h}_i(k)} \ln Z &= \sum_{\boldsymbol{\sigma}} I[\sigma_i = k] P(\boldsymbol{\sigma}) = P(\sigma_i = k), \\ \partial_{\mathbf{J}_{ij}(k,l)} \ln Z &= \sum_{\boldsymbol{\sigma}} I[\sigma_i = k] I[\sigma_j = l] P(\boldsymbol{\sigma}) = P(\sigma_i = k, \sigma_j = l),\end{aligned}\quad (13)$$

so what the equations (12) actually say is

$$\begin{aligned}P(\sigma_i = k) &= \mathbf{f}_i(k), \\ P(\sigma_i = k, \sigma_j = l) &= \mathbf{f}_{ij}(k, l),\end{aligned}\quad (14)$$

i.e., that the ML estimates should yield marginal probabilities that match the empirically observed frequencies.

Sufficient statistics

Z is a sum over all states $\boldsymbol{\sigma}$, so its evaluation is independent of the set $\{\boldsymbol{\sigma}^{(b)}\}_{b=1}^B$. Hence, the data enter into equations (12) only through $\mathbf{f}_i(k)$ and $\mathbf{f}_{ij}(k, l)$. Consequently, the ML estimates can actually be acquired from the frequencies alone, i.e., they do not demand the full configurations $\{\boldsymbol{\sigma}^{(b)}\}_{b=1}^B$. This can be motivated by familiar statistical theory as follows. By using indicator functions, (4) can alternatively be written

$$\begin{aligned}P(\boldsymbol{\sigma}) &= \\ \frac{1}{Z} \exp &\left(\sum_{i=1}^N \sum_{k=1}^q \mathbf{h}_i(k) I[\sigma_i = k] + \sum_{i=1}^{N-1} \sum_{j=i+1}^N \sum_{k=1}^q \sum_{l=1}^q \mathbf{J}_{ij}(k, l) I[\sigma_i = k] I[\sigma_j = l] \right).\end{aligned}\quad (15)$$

This distribution is a member of the *exponential family*, and the functions multiplying the parameters in the exponential are *sufficient statistics* (see e.g. Wainwright and Jordan (2008)). This means that no information (about the model parameters) in the full data set exists which, in theory, cannot also be extracted from the averages $\langle I[\sigma_i = k] \rangle$ and $\langle I[\sigma_i = k] I[\sigma_j = l] \rangle$. These are indeed $\mathbf{f}_i(k)$ and $\mathbf{f}_{ij}(k, l)$. The frequencies being sufficient for learning is quite intuitive; they were the starting point for the maximum-entropy derivation in the first place. It seems sensible that the quantities that instigated the model are able to specify it completely. We remark again that the model must be reduced from its over-parameterized status (e.g. by imposing (6) or (7)) for the sufficient statistics to generate a unique set of ML estimates.

The intractability of Z

As we've hinted, the ML route to finding estimates runs into a major barrier in practical situations. So, what is it that makes solving the equations (12) so difficult? The answer is a common one in statistical inference: the normalization constant Z is incredibly expensive computationally. It demands a summation over all possible states $\boldsymbol{\sigma}$, and the number of states increases exponentially with the system size N making this a daunting task even for relatively small systems (in the Ising case, $N = 20$ is sometimes mentioned as a limit). Methods in this business therefore generally must, in one way or another, sidestep an exhaustive evaluation of Z .

Graphical model selection and inverse Potts

The hunt for such methods has intensified over the last decade, and by now many ways to efficiently approximate Z have been put forth. It can be difficult to get a clear overview of the existing toolkit though, partly because this research lives in the cross section of several subjects: statistics, statistical physics and machine learning. Even though the core problem (parameter selection in distributions of the form (4)) is the same, notions differ slightly in what the input and output of an algorithm should be and what data assumptions can be made. Also, as is natural, terminology and notation is not consistent across the fields.

In the statistics community, the problem is commonly studied in relation to *graphical model selection* (see e.g. Ravikumar et al. (2010) and Jalali et al. (2011) who study PLM for $q = 2$ and $q > 2$ respectively). Efforts there put main theoretical focus on rebuilding a system’s underlying graph, in which an edge is said to exist between nodes i and j if one or more of the elements in \mathbf{J}_{ij} are nonzero (this definition depends on what parameter constraints are used, though). The main concern then is whether each $\mathbf{J}_{ij}(k, l)$ is zero or not; the actual parameter values are secondary. Yet, these methods generally *do* construct parameter estimates and can thus be used for this task as well. It is normally assumed here that the full set $\{\boldsymbol{\sigma}^{(b)}\}_{b=1}^B$ can be accessed at will.

Inverse Ising/Potts is more of a statistical-physics term. Although not strictly defined, it routinely refers to reacquiring $\{\mathbf{h}, \mathbf{J}\}$ using the frequencies (1) only. Thus, access to the full configurations $\{\boldsymbol{\sigma}^{(b)}\}_{b=1}^B$ is often not presumed here. Remember that the Potts model was credited as the least restricting distribution choice when given $\mathbf{f}_i(k)$ and $\mathbf{f}_{ij}(k, l)$, so there is some sense in feeding only these to the inverse methods.

Along those lines, one could ask: since the frequencies are ‘sufficient’, why would there ever be a point in hoarding the full sample collection? The answer is: even though theoretically no extra statistical ‘capital’ is sustained by doing so, it can simplify things computationally and allow otherwise out-of-reach styles of approximation.

The differences between strict inverse Potts and graphical model selection are characterized by NMFI and PLM. The former is typically backed by statistical-physics arguments and takes as input $\mathbf{f}_i(k)$ and $\mathbf{f}_{ij}(k, l)$, whereas the latter is more of a pure statistics concept (usually, Besag (1975) is credited) and uses all the data.

4 Method development

In this chapter, we trace the crucial steps of the NMFI usage by Morcos et al. (2011) and in parallel explain our chosen corresponding actions for PLM. At the end of the chapter, we state the full versions of both algorithms and inform on how we engage them numerically.

In some segments with technical content, we revert to the Ising setting (8). As is standard in the literature on inverse Ising, we then express the relevant quantities as means, or *magnetizations*, $m_i = \langle \sigma_i \rangle$ and correlations $c_{ij} = \langle \sigma_i \sigma_j \rangle - m_i m_j$ (instead of frequencies) and use spin variables $\sigma_i = \pm 1$.

Section 4.3 contains a quick survey of some contenders for inverse Potts other than NMFI and PLM. That section is not essential and can be skipped at will.

4.1 Naive mean-field inversion

We now lay out the NMFI execution in the Ising case and follow up with a generalization to the Potts model. NMFI rests on the approximate supposition that contributions to m_i come through the averages of the other spins m_j , $j \neq i$ (via the interaction strengths J_{ij}), and from the node's own local field h_i . This casts a set of coupled equations as

$$m_i \approx \frac{1e^{h_i + \sum_{j \neq i} J_{ij} m_j} + (-1)e^{-h_i - \sum_{j \neq i} J_{ij} m_j}}{e^{h_i + \sum_{j \neq i} J_{ij} m_j} + e^{-h_i - \sum_{j \neq i} J_{ij} m_j}}, \quad (16)$$

or, rearranged,

$$\tanh^{-1}(m_i) \approx h_i + \sum_{j \neq i} J_{ij} m_j. \quad (17)$$

Differentiation with respect to m_j , $j \neq i$, gives a convenient expression for the couplings (see e.g. Roudi et al. (2009)):

$$J_{ij}^{NMFI} = -(\mathbf{C}^{-1})_{ij}, \quad (18)$$

where \mathbf{C} is the matrix with c_{ij} in position (i, j) . The generalization to $q > 2$ turns out to be direct. Using the constraints (6), c_{ij} from the $q = 2$ case gets replaced by a $(q-1) \times (q-1)$ submatrix built from $\mathbf{c}_{ij}(k, l)$ (as defined by (2)) for all k and l except the last state q . This assembles a $N(q-1) \times N(q-1)$ correlation matrix \mathbf{C} , from which the couplings are calculated as

$$J_{ij,kl}^{NMFI} = -(\mathbf{C}^{-1})_{ab}, \quad (19)$$

where $a = (q-1)(i-1) + k$ and $b = (q-1)(j-1) + l$. In (19), k and l run from 1 to $q-1$ (under (6), the rest of the parameters are zero). For a derivation of NMFI in the Potts setting, see Morcos et al. (2011).

4.2 Pseudolikelihood maximization

We now derive our version of PLM. The general principle is to boot out the proper objective (10), which contains the problematic Z , and fabricate a reduced quasiversion which avoids a full-space normalization. Several realizations of this idea are possible, but we use the (somewhat standard) one where

each sample $\sigma^{(b)}$ contributes to the likelihood *not* through its full probability (as in (9)), but through the probability of one σ_r conditioned on all the other variables. Thus, we consider $P(\sigma_r = \sigma_r^{(b)} | \sigma_{\setminus r} = \sigma_{\setminus r}^{(b)})$, where $\sigma_{\setminus r} = (\sigma_1, \dots, \sigma_{r-1}, \sigma_{r+1}, \dots, \sigma_N)$, instead of $P(\sigma = \sigma^{(b)})$. We can derive the expression for $P(\sigma_r = \sigma_r^{(b)} | \sigma_{\setminus r} = \sigma_{\setminus r}^{(b)})$ using the formula for conditional probabilities, $P(A|B) = \frac{P(A \cap B)}{P(B)}$, as

$$\begin{aligned} P(\sigma_r = \sigma_r^{(b)} | \sigma_{\setminus r} = \sigma_{\setminus r}^{(b)}) &= \\ &= \frac{P(\sigma_r = \sigma_r^{(b)}, \sigma_{\setminus r} = \sigma_{\setminus r}^{(b)})}{P(\sigma_{\setminus r} = \sigma_{\setminus r}^{(b)})} = \frac{P(\sigma_r = \sigma_r^{(b)}, \sigma_{\setminus r} = \sigma_{\setminus r}^{(b)})}{\sum_{l=1}^q P(\sigma_r = l, \sigma_{\setminus r} = \sigma_{\setminus r}^{(b)})}. \end{aligned} \quad (20)$$

Both the numerator and the terms in the denominator are probabilities of full states σ , and we can therefore plug in (4). But, what distinguishes parts of (20) are only the varying values for σ_r , so all parts of (4) not concerning the state of node r will be identical and cancel out, including Z . What remains is

$$P(\sigma_r = \sigma_r^{(b)} | \sigma_{\setminus r} = \sigma_{\setminus r}^{(b)}) = \frac{\exp\left(\mathbf{h}_r(\sigma_r^{(b)}) + \sum_{\substack{i=1 \\ i \neq r}}^N \mathbf{J}_{ri}(\sigma_r^{(b)}, \sigma_i^{(b)})\right)}{\sum_{l=1}^q \exp\left(\mathbf{h}_r(l) + \sum_{\substack{i=1 \\ i \neq r}}^N \mathbf{J}_{ri}(l, \sigma_i^{(b)})\right)}, \quad (21)$$

where, for notational convenience, we take $\mathbf{J}_{ri}(l, k)$ to mean $\mathbf{J}_{ir}(k, l)$ when $i < r$. This quantity contains no nasty normalization. In a sense, normalization *is* still going on though; the denominator can be seen as our 'new Z ', particular to the node r . The dependent variable σ_r takes on just q states (contrasted to σ with its q^N states), so this normalization is compatible with large N . Given an MSA, we can maximize the conditional likelihood by minimizing

$$f_r(\mathbf{h}_r, \mathbf{J}_r) = -\frac{1}{B} \sum_{b=1}^B \ln \left[P_{\{\mathbf{h}_r, \mathbf{J}_r\}}(\sigma_r = \sigma_r^{(b)} | \sigma_{\setminus r} = \sigma_{\setminus r}^{(b)}) \right]. \quad (22)$$

Note that this only depends on \mathbf{h}_r and $\mathbf{J}_r = \{\mathbf{J}_{ir}\}_{i \neq r}$, that is, on the parameters featuring node r . We form our final objective function by adding f_r for all nodes:

$$np\ell(\mathbf{h}, \mathbf{J}) = \sum_{r=1}^N f_r(\mathbf{h}_r, \mathbf{J}_r) = \sum_{r=1}^N -\frac{1}{B} \sum_{b=1}^B \ln \left[P_{\{\mathbf{h}_r, \mathbf{J}_r\}}(\sigma_r = \sigma_r^{(b)} | \sigma_{\setminus r} = \sigma_{\setminus r}^{(b)}) \right]. \quad (23)$$

The shortening *np\ell* stands for *negative pseudo-log-likelihood*. We define our PLM estimates as

$$\{\mathbf{h}^{PLM}, \mathbf{J}^{PLM}\} = \underset{\{\mathbf{h}, \mathbf{J}\}}{\operatorname{argmin}} \{np\ell(\mathbf{h}, \mathbf{J})\}. \quad (24)$$

It is important to underline that minimizers of *np\ell* generally do not minimize *nll*; the replacement of likelihood with pseudolikelihood *does* alter the outcome. This is the fee we pay to access nontrivial N . PLM is, however, an experimentally well-backed method whose solution trajectories often run remarkably close to those of conventional ML.

Parallel execution of PLM

Another implementation of PLM is to minimize each f_r separately, which saves one of having to forge a big substitute of nll as in (23). This approach hands out two, generally different, estimates of each \mathbf{J}_{ij} : one when σ_i is considered the dependent variable and one when σ_j is. Symmetry must then be imposed heuristically, for example by taking averages. Although slightly cruder, this type of PLM allows trivial parallelization of the numerical work, since it splits the original problem into N subproblems which can be solved independently. Höfling and Tibshirani (2009) explored the behavior of the two versions and found that the one-big-optimization variant (which we use) was preferable to the N -split one in terms of accuracy, although both versions performed well.

Consistency

A qualitative difference between PLM and many other estimation schemes is the *consistent* nature of its estimates. Consistency means that the method is sure to conjure the true parameters as $B \rightarrow \infty$, if the samples are in fact drawn (independently) from the distribution in question. It is a general characteristic of pseudolikelihood estimates, but we settle for a demonstration for (8), the Ising model (recall that $\sigma_i = \pm 1$ in that formulation). The conditional probability (21) then reads

$$P_{\{h_r, \mathbf{J}_r\}}(\sigma_r = \sigma_r^{(b)} | \boldsymbol{\sigma}_{\setminus r} = \boldsymbol{\sigma}_{\setminus r}^{(b)}) = \frac{1}{1 + e^{-2\sigma_r^{(b)}[h_r + \sum_{i \neq r} J_{ir}\sigma_i^{(b)}]}}. \quad (25)$$

We take for now $\{\mathbf{h}, \mathbf{J}\}$ as the true parameters of the system. As B grows larger, the empirical mean of (22) will eventually emerge as a true average, evaluated as $\langle A \rangle = \sum_{\boldsymbol{\sigma}} AP_{\{\mathbf{h}, \mathbf{J}\}}(\boldsymbol{\sigma})$. So, f_r can be expressed as

$$\begin{aligned} f_r(h'_r, \mathbf{J}'_r) &\approx \langle -\ln(P_{\{h'_r, \mathbf{J}'_r\}}(\sigma_r | \boldsymbol{\sigma}_{\setminus r})) \rangle \\ &= \sum_{\boldsymbol{\sigma}} \ln\left(1 + e^{-2\sigma_r[h'_r + \sum_{i \neq r} J'_{ir}\sigma_i]}\right) P_{\{\mathbf{h}, \mathbf{J}\}}(\boldsymbol{\sigma}), \end{aligned} \quad (26)$$

with equality expected in the limit $B \rightarrow \infty$. In this limit, the derivatives of f_r become

$$\frac{\partial f_r}{\partial J'_{sr}}(h'_r, \mathbf{J}'_r) = \sum_{\boldsymbol{\sigma}} \frac{-2\sigma_s \sigma_r}{e^{2\sigma_r[h'_r + \sum_{i \neq r} J'_{ir}\sigma_i]} + 1} P_{\{\mathbf{h}, \mathbf{J}\}}(\boldsymbol{\sigma}), \quad (27)$$

for $s = 1, \dots, N$ (and similarly for the field derivative). At the true parameters $\{h_r, \mathbf{J}_r\}$, the summands can be adjusted as follows:

$$\begin{aligned}
\frac{\partial f_r}{\partial J'_{sr}}(h_r, \mathbf{J}_r) &= \\
&= \sum_{\sigma} \frac{-2\sigma_s \sigma_r}{e^{2\sigma_r[h_r + \sum_{i \neq r} J_{ir} \sigma_i]} + 1} P_{\{\mathbf{h}, \mathbf{J}\}}(\sigma) = \\
&= \frac{1}{Z\{\mathbf{h}, \mathbf{J}\}} \sum_{\sigma} \frac{-2\sigma_s \sigma_r}{e^{2\sigma_r[h_r + \sum_{i \neq r} J_{ir} \sigma_i]} + 1} e^{\sum_{i=1}^N h_i \sigma_i + \sum_{i=1}^{N-1} \sum_{j=i+1}^N J_{ij} \sigma_i \sigma_j} = \\
&= -\frac{1}{Z\{\mathbf{h}, \mathbf{J}\}} \sum_{\sigma} \sigma_s \sigma_r \frac{e^{-\sigma_r[h_r + \sum_{i \neq r} J_{ir} \sigma_i]} e^{\sum_{i=1}^N h_i \sigma_i + \sum_{i=1}^{N-1} \sum_{j=i+1}^N J_{ij} \sigma_i \sigma_j}}{\frac{1}{2} \left(e^{\sigma_r[h_r + \sum_{i \neq r} J_{ir} \sigma_i]} + e^{-\sigma_r[h_r + \sum_{i \neq r} J_{ir} \sigma_i]} \right)} = \\
&= -\frac{1}{Z\{\mathbf{h}, \mathbf{J}\}} \sum_{\sigma} \sigma_s \sigma_r \frac{e^{\sum_{i \neq r} h_i \sigma_i + \sum_{\substack{i < j \\ i, j \neq r}} J_{ij} \sigma_i \sigma_j}}{\cosh(\sigma_r[h_r + \sum_{i \neq r} J_{ir} \sigma_i])} . \tag{28}
\end{aligned}$$

The quotients in the last sum are independent of whether σ_r is 1 or -1 (\cosh is an even function). Each state has exactly one other state identical except for an antialigned σ_r , and the contributions from such companions are equal in size but opposite in sign, so the sum above vanishes. Hence, $\frac{\partial f_r}{\partial J'_{sr}}(h_r, \mathbf{J}_r) = 0$. Calculations are analogous for the variation with respect to h_r . This means f_r has a stationary point at the true parameters. Appropriate definiteness of this point can be checked to ensure a minimum. Assuming that we can locate this point, our PLM procedure is exact in the limit of large sample size. Note also that the cancellation occurs within each separate f_r , so the consistency argument is valid also for the parallel PLM variant described earlier.

We remark that consistency may be of limited relevance in our setting, since MSAs are, at best, only approximately generated from (4). It is nevertheless an appealing theoretical property.

4.3 Other methods

We now describe some other current Potts inversion techniques. Everything is handled in the Ising regime here, but in principle all methods should be extendable to $q > 2$. The list presented in this section is not exhaustive.

Boltzmann learning

Boltzmann learning (Ackley et al., 1985) is a popular method, the idea behind which is simple: $\{\mathbf{m}, \mathbf{C}\}$ can be generated from $\{\mathbf{h}, \mathbf{J}\}$ using Monte Carlo sampling, so just tune the fields and couplings until the corresponding means and correlations that come out match the empirical ones (in accordance with equations (14)). The update rules can look like

$$\begin{aligned}
\delta J_{ij} &= \eta(\langle \sigma_i \sigma_j \rangle_{data} - \langle \sigma_i \sigma_j \rangle_{\{\mathbf{h}, \mathbf{J}\}}), \\
\delta h_i &= \eta(\langle \sigma_i \rangle_{data} - \langle \sigma_i \rangle_{\{\mathbf{h}, \mathbf{J}\}}), \tag{29}
\end{aligned}$$

for some constant or function η . This incremental update scheme is guaranteed to converge to the ML estimates given sufficient time. But, the Monte Carlo

sampling sessions needed to generate $\langle \sigma_i \sigma_j \rangle_{\{\mathbf{h}, \mathbf{J}\}}$ and $\langle \sigma_i \rangle_{\{\mathbf{h}, \mathbf{J}\}}$ can be tedious, and a simple set-up of this method is unrealistic to apply to most relevant system sizes. Yet, it *can* sometimes be worth paying the time-cost up front. For instance, when testing approximate ML routines on real data sets and one is in need of 'true' parameters to compare with, Boltzmann estimates can serve as such. Also, cleverly accelerated variants of this procedure make up an interesting class of Potts inverters (see e.g. Broderick et al. (2007)).

Thouless-Anderson-Palmer inversion

It is straightforward to include the next order of the mean-field approximation (17), which, informally speaking, adds the two-step effect that spin i exhibits on itself via another spin and back (Roudi et al., 2009),

$$\tanh^{-1}(m_i) \approx h_i + \sum_{j \neq i} J_{ij} m_j - \sum_{j \neq i} J_{ij}^2 m_i (1 - m_j^2). \quad (30)$$

One can again differentiate with respect to m_j , which gives equations solvable for the *Thouless-Anderson-Palmer* (TAP) (Thouless et al., 1977) inversion estimates,

$$J_{ij}^{TAP} = -(\mathbf{C}^{-1})_{ij} - 2(J_{ij}^{TAP})^2 m_i m_j. \quad (31)$$

J_{ij}^{TAP} coincides with J_{ij}^{NMFI} for zero means, but yields a better estimate in general. Morcos et al. (2011) tried the Potts version of the TAP expansion (NMFI and TAP are in a sense expansions in small couplings) for PSP, but found no significant improvement over NMFI.

Sessak-Monasson expansion

Sessak and Monasson (2009) provided a systematic technique to expand the likelihood in small correlations. A resulting expression for the couplings up to a certain order was included, which can be written

$$J_{ij}^{SM} = J_{ij}^{NMFI} + J_{ij}^{IP} - \frac{c_{ij}}{(1 - m_i^2)(1 - m_j^2) - c_{ij}^2}, \quad (32)$$

where J_{ij}^{IP} is the *independent-pair* approximation

$$J_{ij}^{IP} = \frac{1}{4} \ln \left[\frac{((1 + m_i)(1 + m_j) + c_{ij})((1 - m_i)(1 - m_j) + c_{ij})}{((1 - m_i)(1 + m_j) - c_{ij})((1 + m_i)(1 - m_j) - c_{ij})} \right]. \quad (33)$$

Roudi et al. (2009) showed that this expansion outperforms other state-of-the-art methods (particularly in hybrid with TAP) on artificial neural data. Sessak and Monasson (2009) took the order higher at zero magnetization, giving

$$\begin{aligned} J_{ij}^{SM\{m=0\}} &= J_{ij}^{SM} + \\ &+ \sum_{\substack{k=1 \\ k \neq i, j}}^N \frac{1}{4} \ln \left[\frac{(1 + c_{ij} - c_{ik} - c_{jk})(1 - c_{ij} - c_{ik} + c_{jk})(1 + c_{ij} + c_{ik} + c_{jk})}{(1 - c_{ij} - c_{ik} + c_{jk})(1 - c_{ij} + c_{ik} - c_{jk})(1 - c_{ij} - c_{ik} + c_{jk})} \right] \\ &- \sum_{\substack{k=1 \\ k \neq i, j}}^N \left(J_{ij}^{IP} + \frac{c_{ij} - c_{ik} c_{jk}}{1 - c_{ij}^2 - c_{ik}^2 - c_{jk}^2 + 2c_{ij} c_{jk} c_{ki}} - \frac{c_{ij}}{1 - c_{ij}^2} \right). \end{aligned} \quad (34)$$

More methods

The susceptibility-propagation method (an iterative message-passing procedure, Mezard and Mora (2009)) used by Weigt et al. (2009) works best for treelike interaction networks, i.e., when the underlying graph is not loopy.

One of the most interesting candidates at present is the *cluster expansion* of Cocco and Monasson (2011a–2011b). A cluster is a node subset of the system, such as $\{i, j, k\} = \{2, 7, 9\}$. This method, loosely speaking, accepts or casts aside a cluster’s contribution to the $\{\mathbf{h}, \mathbf{J}\}$ -estimates depending on whether or not that cluster’s entropy contribution exceeds some threshold Θ .

Other relevant methods are *Bethe reconstruction* (Nguyen and Berg, 2012; Ricci-Tersenghi, 2011), *contrastive divergence* (Carreira-Perpinan and Hinton, 2005), which is closely tied to pseudolikelihood (Hyvärinen, 2007), and *minimum probability flow* (Sohl-Dickstein et al., 2011).

4.4 Regularization

When learning from limited data, true model properties and data weaknesses can often yield deceptively similar outputs. Consequently, there is always a risk that a seemingly relevant find is just a display of undersampling. This issue, termed *overfitting* in statistics, tends to plague inverse problems on large systems. *Regularization* is a term commonly used for add-on techniques aimed at reducing this risk. In PSP, current data sets are certainly dirty enough to earn the attention of such enhancement methods.

Pseudocount for NMFI

Morcos et al. (2011) used a counter tactic built on pretend sightings of states. Mathematically, the frequencies were adjusted using a regularization variable λ , called the *pseudocount*, as

$$\begin{aligned} \mathbf{f}_i(k) &= \frac{1}{\lambda + B} \left(\frac{\lambda}{q} + \sum_{b=1}^B I[\sigma_i^{(b)} = k] \right), \\ \mathbf{f}_{ij}(k, l) &= \frac{1}{\lambda + B} \left(\frac{\lambda}{q^2} + \sum_{b=1}^B I[\sigma_i^{(b)} = k] I[\sigma_j^{(b)} = l] \right). \end{aligned} \quad (35)$$

This insertion dampens the eagerness to fit the data meticulously, and it also promotes invertibility of the matrix in (19).

Penalty term for PLM

Seeing that PLM does not concern itself with frequencies (it uses all the data), we do not simply copy the approach above. Instead, we take the standard route of adding a penalty term to the objective function:

$$\{\mathbf{h}^{PLM}, \mathbf{J}^{PLM}\} = \underset{\{\mathbf{h}, \mathbf{J}\}}{\operatorname{argmin}} \{npll(\mathbf{h}, \mathbf{J}) + R(\mathbf{h}, \mathbf{J})\}. \quad (36)$$

The turnout is then a trade-off between likelihood maximization and whatever qualities R is pushing for. We stick to the penalty-term pedagogy here, but this

common technique has other interpretations. It is, for example, equivalent to enforcing prior distributions on the model parameters. Our main selection for R is

$$R_{l_2}(\mathbf{h}, \mathbf{J}) = \lambda_h \sum_{r=1}^N \|\mathbf{h}_r\|_2^2 + \lambda_J \sum_{i=1}^{N-1} \sum_{j=i+1}^N \|\mathbf{J}_{ij}\|_2^2, \quad (37)$$

where

$$\begin{aligned} \|\mathbf{h}_i\|_2^2 &= \sum_{k=1}^q \mathbf{h}_i(k)^2, \\ \|\mathbf{J}_{ij}\|_2^2 &= \sum_{k=1}^q \sum_{l=1}^q \mathbf{J}_{ij}(k, l)^2. \end{aligned} \quad (38)$$

This is called *l_2 -regularization*. R_{l_2} clearly punishes large parameter magnitudes and nudges the optimum toward the origin of the parameter space. Indeed, as far as R_{l_2} is concerned the ideal solution would be all parameters at zero. This addition will serve to calm the estimator's drive toward too picky a fit. It is up to the user to control the compromise between R and the original objective by prescribing well-sized regularization constants λ_h and λ_J .

Balakrishnan et al. (2011) instead used the *group l_1 -regularization* term

$$R_{gl_1}(\mathbf{h}, \mathbf{J}) = \lambda_h \sum_{r=1}^N \|\mathbf{h}_r\|_2^2 + \lambda_J \sum_{i=1}^{N-1} \sum_{j=i+1}^N \|\mathbf{J}_{ij}\|_2. \quad (39)$$

In R_{gl_1} , coupling parameters associated with the same pair are gathered inside a square root (each pair is a 'group' in this case). This way, the parameters belonging to the same \mathbf{J}_{ij} are collectively repressed and pushed toward zero as a pack. l_1 -regularization generally gives sparser solutions (solutions where more parameters are exactly zero) than l_2 -regularization does; a tiny parameter contributes less to the penalty term if squared first. With group l_1 -regularization, one should expect pairwise sparsity, i.e., expect optimums at which many \mathbf{J}_{ij} are zero-matrices.

Either of these regularization terms would be a sensible choice for our problem. R_{l_2} is a differentiable function, so it adds little labor to the optimization. R_{gl_1} , on the other hand, suffers from nondifferentiability at points where a zero occurs under a root sign. We use R_{l_2} as our chief regularizer, but in section 5.9 also provide some comparing results for R_{gl_1} .

R_{l_2} , R_{gl_1} , and overparameterization

In section 3.2, we pointed out that constraints like (6) or (7) ensure a unique solution to the inverse Potts problem. It turns out that R_{l_2} and R_{gl_1} actually autoserve as enforcers of such constraints. Suppose, for a visualizable example of why that could be, that minima under the full parameterization occur along a curve in 3D-space (not a unique point). R_{l_2} and R_{gl_1} both encourage solutions close to the origin, hence preferring a particular point on the curve. More specifically, the use of any strictly convex regularizer will remove the freedom of the full parameterization (Schmidt, 2010). This allows us the luxury of using all parameters in our PLM implementation.

To conclude this section: for NMFI, we follow Morcos et al. (2011), i.e., regularize with pseudocounts under the constraints (6). For PLM, we regularize with R_{l_2} (and in section 5.9 R_{gl_1}) under the full parameterization.

4.5 Sequence reweighting

As discussed at the end of section 3.2, an overlook of the MSAs in Pfam disagrees with the premise of independent samples. The observed skewing has several probable reasons: bias in choices of which species to sequence from and *phylogenetic* bias caused by evolutionary relatedness between the species, to name but two.

A blunt way to mitigate effects of sequence interdependence would be to step through the MSA and remove alike sequences. But, that would unnecessarily dispose of a chunk of the information at hand. A better approach, employed by Morcos et al. (2011), is to equip each sequence $\sigma^{(b)}$ with a *weight* w_b which regulates the impact on the parameter estimates. Sequences judged as unworthy of independent-sample status (too similar to other sequences) can then have their weight lowered.

Morcos et al. (2011) measured the similarity of two sequences as the fraction of positions where the amino acids are identical. They were deemed 'too' similar if this fraction exceeded some predefined margin x , $0 < x < 1$. The weight put on sequence $\sigma^{(b)}$ was $w_b = \frac{1}{m_b}$, where m_b was the number of sequences in the MSA similar to $\sigma^{(b)}$:

$$m_b = |\{a, 1 \leq a \leq B : \text{similarity}(\sigma^{(a)}, \sigma^{(b)}) \geq x\}|. \quad (40)$$

This way, distinct sequences (as specified by x) were awarded a weight of 1, whereas a sequence with, for example, 50 other similar ones in the MSA (as specified by x) received a weight of $1/50$. A suitable threshold x was found to be 0.8 (results were only weakly dependent on this choice throughout $0.7 < x < 0.9$).

Featuring this reweighting idea in a parameter selection procedure means rescaling each sequence quantity by its weight. Morcos et al. (2011) did this right at the calculation of the frequencies, as

$$\begin{aligned} \mathbf{f}_i(k) &= \frac{1}{B_{eff}} \sum_{b=1}^B w_b I[\sigma_i^{(b)} = k], \\ \mathbf{f}_{ij}(k, l) &= \frac{1}{B_{eff}} \sum_{b=1}^B w_b I[\sigma_i^{(b)} = k] I[\sigma_j^{(b)} = l], \end{aligned} \quad (41)$$

where $B_{eff} = \sum_{b=1}^B w_b$ becomes a measure of the number of nonredundant, or *effective*, sequences. The final frequency formulas used, combining reweighting and pseudocounts, were

$$\begin{aligned}
\mathbf{f}_i(k) &= \frac{1}{\lambda + B_{eff}} \left(\frac{\lambda}{q} + \sum_{b=1}^B w_b I[\sigma_i^{(b)} = k] \right), \\
\mathbf{f}_{ij}(k, l) &= \frac{1}{\lambda + B_{eff}} \left(\frac{\lambda}{q^2} + \sum_{b=1}^B w_b I[\sigma_i^{(b)} = k] I[\sigma_j^{(b)} = l] \right). \quad (42)
\end{aligned}$$

We now translate this step to PLM. The objective function $npll$ acknowledges each sample with a sum across the N nodes (see (23)). Appropriate rescaling ought to multiply this whole sum by the sequence weight. Therefore, our choice of objective function for PLM is not (23), but

$$npll(\mathbf{h}, \mathbf{J}) = -\frac{1}{B_{eff}} \sum_{b=1}^B w_b \sum_{r=1}^N \ln \left[P_{\{\mathbf{h}_r, \mathbf{J}_r\}}(\sigma_r = \sigma_r^{(b)} | \sigma_{\setminus r} = \sigma_{\setminus r}^{(b)}) \right]. \quad (43)$$

4.6 Interaction scores

As the procedures have been described so far, each pair (i, j) spawns a whole matrix \mathbf{J}_{ij}^* of estimates. It is not our goal to rank interconnection strengths between specific position/amino acid combinations, which the elements $\mathbf{J}_{ij}(k, l)$ represent; the end product should be a single magnitude for each pair (i, j) . We therefore need a fair way to break each matrix \mathbf{J}_{ij} into one score.

Mutual information

We start by mentioning a classical scoring tool not involving the Potts model. It is called *mutual information* (MI), and is computed directly from the frequencies as

$$MI_{ij} = \sum_{k=1}^q \sum_{l=1}^q \mathbf{f}_{ij}(k, l) \ln \left(\frac{\mathbf{f}_{ij}(k, l)}{\mathbf{f}_i(k) \mathbf{f}_j(l)} \right). \quad (44)$$

Formally, it is the Kullback-Leibler divergence between the joint and the marginal distributions of the node variables. It serves as a dependence measure taking on zero if positions i and j are not interacting and nonzero values otherwise. MI does not distinguish direct from indirect interactions well. Being outperformed by NMFI (Morcos et al., 2011) and PSICOV (Jones et al., 2012), it helped highlight the power of a direct-coupling analysis for PSP.

Direct information

For NMFI, Morcos et al. (2011) used a score called *direct information* (DI) (previously introduced by Weigt et al. (2009)) whose construction goes as follows. For each pair (i, j) , (the estimate of) \mathbf{J}_{ij} is used to set up a 'direct distribution' involving just nodes i and j ,

$$P_{ij}^{(dir)}(k, l) \sim \exp(\mathbf{J}_{ij}(k, l) + h'_{i,k} + h'_{j,l}). \quad (45)$$

$h'_{i,k}$ and $h'_{j,l}$ are new fields, computed as to ensure agreement with the individual frequencies,

$$\begin{aligned}
\mathbf{f}_i(k) &= \sum_{l=1}^q P_{ij}^{(dir)}(k, l), \\
\mathbf{f}_j(l) &= \sum_{k=1}^q P_{ij}^{(dir)}(k, l).
\end{aligned} \tag{46}$$

After a simple normalization of $P_{ij}^{(dir)}$, the DI score is calculated analogously to MI,

$$DI_{ij} = \sum_{k=1}^q \sum_{l=1}^q P_{ij}^{(dir)}(k, l) \ln \left(\frac{P_{ij}^{(dir)}(k, l)}{\mathbf{f}_i(k) \mathbf{f}_j(l)} \right). \tag{47}$$

DI recognizes only exchange which is direct between nodes i and j . Moreover, it can be shown to be independent of what parameter constraints are used; corresponding parameter sets under (6) and (7) (or other choices) generate identical DI.

Frobenius norm

Although we could certainly use DI for PLM, this would require a pseudocount λ to regularize the frequencies in the DI computation, introducing a third regularization variable in addition to λ_h and λ_J . Another possible quantity by which to rank, mentioned by Weigt et al. (2009), is the *Frobenius norm*

$$\|\mathbf{J}_{ij}\|_2 = \sqrt{\sum_{k=1}^q \sum_{l=1}^q \mathbf{J}_{ij}(k, l)^2}. \tag{48}$$

Unlike DI, this measurement is *not* independent of constraint choice, so we must be a bit careful. (7) are the constraints that shift as much as possible of the exponential in (4) into the fields ((7) minimizes the Frobenius norm, as Weigt et al. (2009) had noted), in a sense making (7) the appropriate fix under which to compute the score (48). Recall from section 4.4 that our algorithm uses the full representation and lets R_{l_2} or R_{gl_1} autofix the parameters. Luckily, it is straightforward to transfer between constraints *after* we have our estimates. To switch into (7), we use

$$\mathbf{J}'_{ij}(k, l) = \mathbf{J}_{ij}(k, l) - \mathbf{J}_{ij}(\cdot, l) - \mathbf{J}_{ij}(k, \cdot) + \mathbf{J}_{ij}(\cdot, \cdot), \tag{49}$$

where ' \cdot ' denotes average. One can show that (49) preserves the probabilities of (4) (after altering the fields appropriately) and that $\mathbf{J}'_{ij}(k, l)$ satisfy (7). A possible Frobenius norm score is hence

$$FN_{ij} = \|\mathbf{J}'_{ij}\|_2 = \sqrt{\sum_{k=1}^q \sum_{l=1}^q \mathbf{J}'_{ij}(k, l)^2}. \tag{50}$$

Jones et al. (2012), whose PSICOV method also used a norm rank (but l_1 -norm instead of Frobenius norm), adjusted their scores with an *average product correction* (APC) term. APC was used for MI by Dunn et al. (2008) to suppress

effects from phylogenetic bias and insufficient sampling. We also incorporate this correction, making our final output

$$CN_{ij} = FN_{ij} - \frac{FN_{\cdot j}FN_{i \cdot}}{FN_{\cdot \cdot}}, \quad (51)$$

where CN stands for 'corrected norm'.

4.7 The finished algorithms

Below, we recap all the steps from MSA to finished interaction scores. Input variables for NMFI are reweighting threshold x and pseudocount λ , and the steps are:

1. Calculate weights $w_b = \frac{1}{m_b}$ according to (40) (takes x).
2. Calculate frequencies using (42) (takes λ).
3. Assemble and invert the correlation matrix, and collect coupling estimates from (19).
4. Calculate DIs as described in section 4.6.

For PLM, the user-specified variables are reweighting threshold x and regularization constants λ_h and λ_J , and the steps are:

1. Calculate weights $w_b = \frac{1}{m_b}$ according to (40) (takes x).
2. Get coupling estimates by solving (36) with $npll$ as in (43) (takes λ_h and λ_J).
3. Impose the constraints (7) using (49) on the coupling estimates.
4. Calculate CNs using (50)-(51).

4.8 Implementation in MATLAB/C

One of the appeals of NMFI is that the Potts inversion is so simple; a single MATLAB command will invert a matrix rather efficiently. Out of the steps in the previous section, only PLM's step two (the minimization) is challenging to realize in code. Under l_2 -regularization, even this step is quite straightforward since the objective is smooth.

There is a large library of efficient algorithms for solving smooth, unconstrained (convex) optimization problems. Some of the most well-known are Newton descent, quasi-Newton descent, and conjugate gradient (CG). We use CG, since it is a *first-order* method, i.e., requires no higher-order derivative than the gradient, and since it uses little memory. Newton and quasi-Newton methods involve the Hessian (albeit approximate in the quasi case), which could require time-costly evaluations or strain the RAM because of large storing demands. CG needs the gradient for $npll$, which we now show how to calculate.

From (21) we have, for some sample $\sigma^{(b)}$,

$$\begin{aligned} \ln \left[P(\sigma_r = \sigma_r^{(b)} | \sigma_{\setminus r} = \sigma_{\setminus r}^{(b)}) \right] = \\ \mathbf{h}_r(\sigma_r^{(b)}) + \sum_{\substack{i=1 \\ i \neq r}}^N \mathbf{J}_{ri}(\sigma_r^{(b)}, \sigma_i^{(b)}) - \ln \left[\sum_{l=1}^q \exp \left(\mathbf{h}_r(l) + \sum_{\substack{i=1 \\ i \neq r}}^N \mathbf{J}_{ri}(l, \sigma_i^{(b)}) \right) \right]. \end{aligned} \quad (52)$$

Note that $\ln \left[P(\sigma_r = \sigma_r^{(b)} | \sigma_{\setminus r} = \sigma_{\setminus r}^{(b)}) \right]$ depends only on parameters associated with node r . The derivative with respect to a coupling is

$$\begin{aligned} \frac{\partial}{\partial \mathbf{J}_{ir}(k, s)} \ln \left[P(\sigma_r = \sigma_r^{(b)} | \sigma_{\setminus r} = \sigma_{\setminus r}^{(b)}) \right] = \\ = I[\sigma_i^{(b)} = k] \exp \left(\mathbf{h}_r(s) + \sum_{\substack{i=1 \\ i \neq r}}^N \mathbf{J}_{ri}(s, \sigma_i^{(b)}) \right) \\ - \frac{I[\sigma_i^{(b)} = k] I[\sigma_r^{(b)} = s]}{\sum_{l=1}^q \exp \left(\mathbf{h}_r(l) + \sum_{\substack{i=1 \\ i \neq r}}^N \mathbf{J}_{ri}(l, \sigma_i^{(b)}) \right)} = \\ = I[\sigma_i^{(b)} = k] \left\{ I[\sigma_r^{(b)} = s] - P(\sigma_r = s | \sigma_{\setminus r} = \sigma_{\setminus r}^{(b)}) \right\}, \end{aligned} \quad (53)$$

and the field derivatives fall out similarly,

$$\frac{\partial}{\partial \mathbf{h}_r(s)} \ln \left[P(\sigma_r = \sigma_r^{(b)} | \sigma_{\setminus r} = \sigma_{\setminus r}^{(b)}) \right] = I[\sigma_r^{(b)} = s] - P(\sigma_r = s | \sigma_{\setminus r} = \sigma_{\setminus r}^{(b)}). \quad (54)$$

In combination with (21) and (43), (53)–(54) enable evaluation of the gradient of $npll$.

M. Schmidt provides a versatile optimization package (<http://www.di.ens.fr/~mschmidt/Software/>, see e.g. Schmidt et al. (2008)) which, among other things, executes parameter learning in the Potts model. It is wrapped by a MATLAB interface and outsources heavy crunching to C. Support is included for CG but also for techniques targeted toward nondifferentiable optimization. The latter enables use under group l_1 -regularization. The package uses the full parameterization for \mathbf{J} but places $\mathbf{h}_i(q) = 0$. This introduces a slight asymmetry in our regularization (the states are not interchangeable), but the effect of this should be small.

Code manipulation is required to fully integrate our plans with this optimizer. For instance, no sequence reweighting is included in the original program (the package is not geared toward PSP specifically), and neither is field regularization. Our end product is a MATLAB file which carries out PLM's steps 1, 3, and 4, and for step 2 calls this (by us modified) optimization package. For NMFI, we use code provided by Morcos et al. (2011). For the data analysis and plot generating (see the next chapter), we use our own MATLAB scripts.

5 Experiments and discussion

We have performed experiments using NMFI and PLM on domain families from Pfam, and the results are reported and discussed in this chapter.

5.1 Families, crystal structures, and true-positive rates

The speed of NMFI enabled Morcos et al. (2011) to conduct a large analysis using 131 families. PLM is computationally more demanding than NMFI, and we were limited in processing power and working memory, so we settled for a subcollection of 17 of these families. These are listed in table 1. To ease the numerical effort, we targeted families with relatively small N .

To reliably assess how good a contact prediction is, something to regards as ground truth is helpful. All 131 families chosen by Morcos et al. (2011) had at least two accessible X-ray crystal structures of resolution $< 3\text{\AA}$. When multiple structures are available for a family, deciding which to use is a task in itself. The crystal data was provided to us by Morcos et al. (2011), so our pick was simply the same as theirs (see their paper for details).

Family ID	N	B	B_{eff} (90%)
PF00011	102	7151	3481
PF00013	58	11484	3785
PF00014	53	3090	1812
PF00017	77	4403	1741
PF00018	48	8993	3354
PF00027	91	17830	9036
PF00028	93	18808	8317
PF00035	67	5584	2254
PF00041	85	26172	10631
PF00043	95	9619	5141
PF00046	57	15445	3314
PF00076	70	31837	14125
PF00081	82	5867	1510
PF00084	56	9816	4345
PF00105	70	4842	1277
PF00107	130	28022	12114
PF00111	78	11941	5805

Table 1: Domain families included in our study, listed with Pfam ID, length N , number of sequences B , and number of effective sequences B_{eff} (under 90% reweighting).

Accuracy results here are reported primarily using *true-positive* (TP) rates, also the principal measurement of Morcos et al. (2011) and Jones et al. (2012). The TP rate for p is the fraction of the p strongest-scored pairs which are actually contacts in the crystal structure (defined using a cutoff distance of 8.5\AA , a choice motivated shortly). To exemplify TP rates, let us jump ahead here and look at fig. 5. For PLM and PF00076, the TP rate is one up to $p = 80$, which means that all 80 highest-CN pairs are genuine contacts in the crystal

structure. At $p = 200$, the TP rate has dropped to 0.78, so $0.78 \cdot 200 = 156$ of the top 200 highest-CN pairs are contacts, and 44 are not.

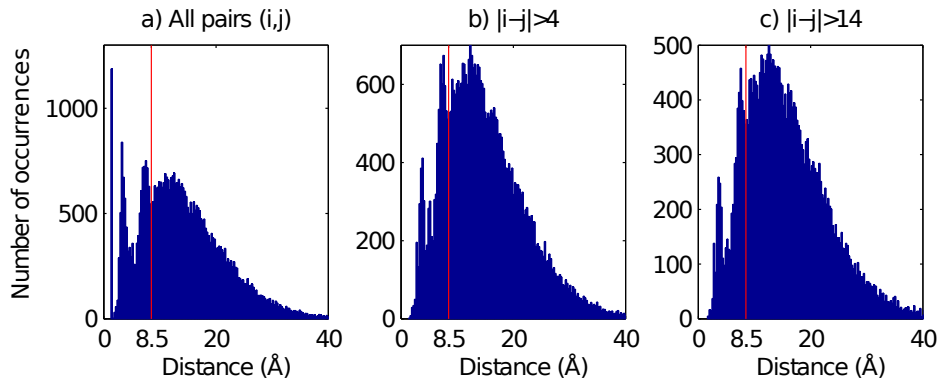


Figure 4: Histograms of crystal-structure distances pooled from all 17 families. The headers state the types of pairs included. The red line is our contact cutoff 8.5Å.

5.2 The real distribution of distances

Figure 4a shows the distribution of position-position distances $d(i, j)$ in all our families, as available from the crystal structures. Three peaks protrude from the background distribution: one distinct at a short distance and two less pronounced around 3-5Å and 8Å. The first one likely corresponds to sequence neighbors, i.e., pairs with small $|i - j|$. The other two were observed in the NMFI output of Morcos et al. (2011), and the interpretations provided said that the second peak "presumably arises from short-ranged interactions like hydrogen bonding or pairings involved in secondary structure formation" and that the third peak "likely corresponds to long-ranged, possibly water mediated, contacts".

When ignoring pairs with $|i - j| \leq 4$, as done in fig. 4b, the first peak vanishes. We conclude that it must indeed arise from pairs of neighboring positions (e.g. as defined by $|i - j| \leq 4$). Uncovering such contacts is in a sense uninteresting, since chain neighbors are trivially expected to be in each other's vicinity. Digging up the positions that are close in 3D-space but *distant* in sequence order is what can really tell us something about the chain's spatial conformation. The second and third peaks, which remain in the $|i - j| > 4$ and $|i - j| > 14$ plots, thus hold the bulk of what is interesting to catch. Morcos et al. (2011) evaluated accuracies using pairs with $|i - j| > 4$ only, and many of the results we present here do the same (when and when not is stated in each individual case). This regards only the post-inference analysis, though; in the parameter selection all pairs were allowed to impact.

The cutoff distance to define 'contact' was by Morcos et al. (2011) put at 8Å. Guided by our distance distributions, we chose to raise this cutoff to 8.5Å to accept the entire third peak into the definition.

5.3 Set-up and preparatory tests

To set the stage for the comparison, we started by running initial trials on our 17 families using both NMFI and PLM with many different regularization and reweighting strengths. Reweighting indeed raised the TP rates, and, as was reported by Morcos et al. (2011) for the 131 families, results seemed robust toward the exact choice of the limit x around $0.7 \leq x \leq 0.9$. We chose $x = 0.9$ to use throughout the study.

In what follows, NMFI results are reported using the same list of pseudocounts as in fig. S11 of Morcos et al. (2011): $\lambda = w \cdot B_{eff}$ with $w = \{0.11, 0.25, 0.43, 0.67, 1.0, 1.5, 2.3, 4.0, 9.0\}$. During our analysis we also ran many intermediate values, but we found this covering sufficiently dense. We will give outputs from two versions of NMFI: NMFI-DI and NMFI-DI(true). The former uses pseudocounts for all calculations, whereas the latter switches to true frequencies when it gets to the DI evaluations.

With l_2 -regularization (our main focus) outcomes were robust against the precise choice of λ_h ; TP rates were pretty much identical when λ_h was changed from 0.001, to 0.01, to 0.1 (yet, $\lambda_h > 0$ seemed necessary). We therefore chose to fix $\lambda_h = 0.01$ for all experiments. What mattered, rather, was the coupling regularization λ_J , for which we did a systematic scan from $\lambda_J = 0$ and up using stepsize 0.005.

So, to summarize, the turnouts reported here are based on $x = 0.9$, cutoff 8.5Å, and $\lambda_h = 0.01$, and draw λ and λ_J from the collections described above.

5.4 Main comparison

Figure 5 shows TP rates for the different families and methods. We see that PLM's TP rates are consistently greater than NMFI's, especially for families with large B_{eff} . As for the two NMFI versions: NMFI-DI(true) dodges the complete failure seen in NMFI-DI for PF00084, but for other families, such as PF00014 and PF00081, the performance instead drops using true DIs.

For both NMFI-DI and NMFI-DI(true), the best regularization was found to be $\lambda = 1 \cdot B_{eff}$, with $\lambda = 1.5 \cdot B_{eff}$ and $\lambda = 2.3 \cdot B_{eff}$ as the runners-up. This is a nice outcome, since $\lambda = 1 \cdot B_{eff}$ also showed optimal performance for the 131 families of Morcos et al. (2011). For PLM, the best was $\lambda_J = 0.01$, followed by $\lambda_J = 0.005$ and $\lambda_J = 0.015$. Interestingly, these same regularization strengths were optimal for basically all families. This is somewhat surprising, since N and especially B_{eff} span quite wide ranges (48-130 and 1277-14225 respectively).

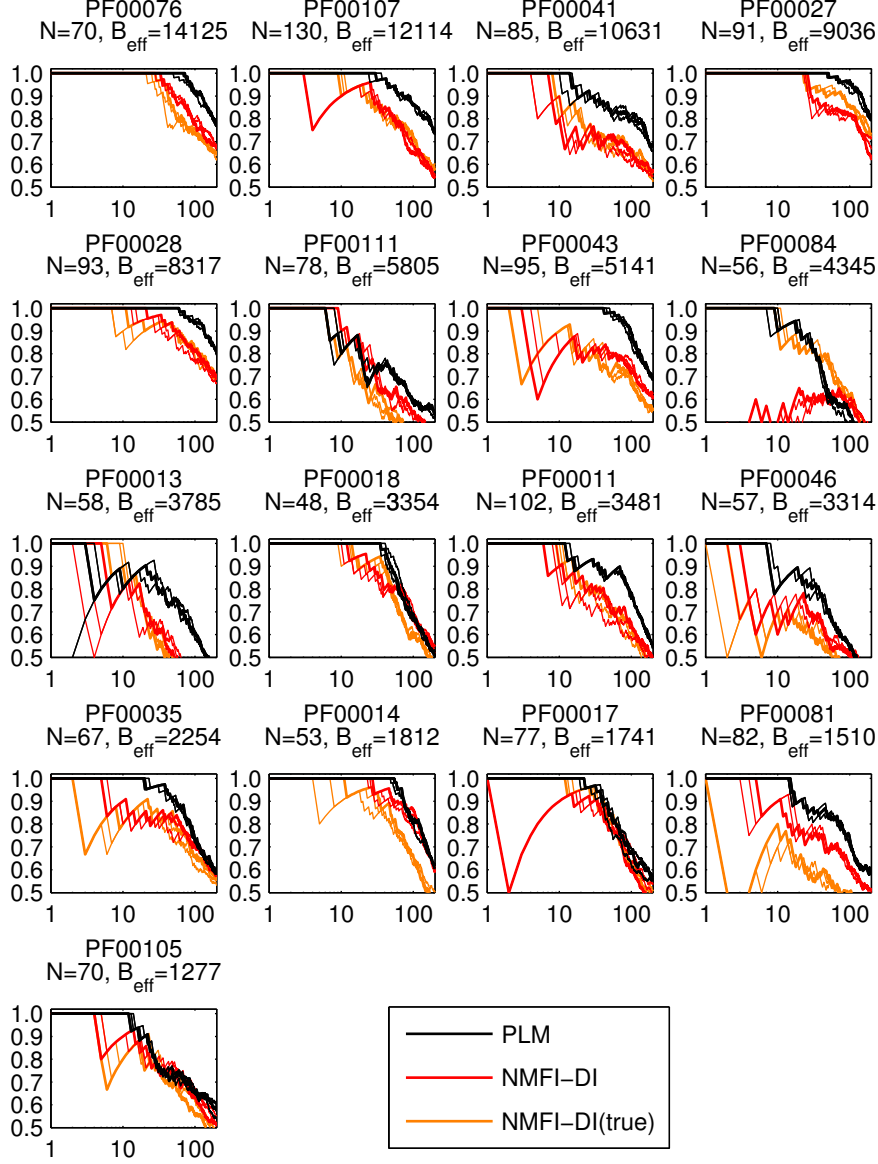


Figure 5: Contact-detection results for all the families in our study, sorted by B_{eff} . Y-axes are TP rates and x-axes are the number of predicted contacts p , based on pairs with $|i - j| > 4$. The three curves for each method are the three regularization levels yielding highest TP rates across all families. The thickened curve highlights the best one out of these three ($\lambda = B_{eff}$ for NMFI and $\lambda_J = 0.01$ for PLM).

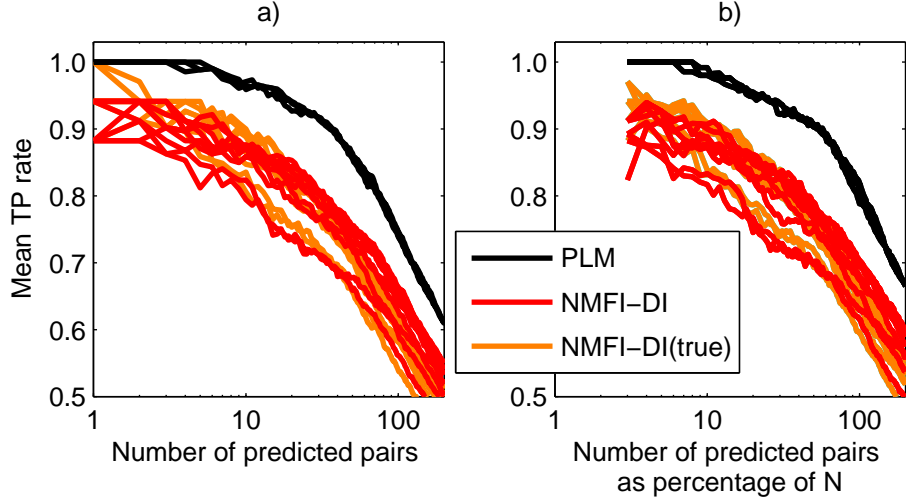


Figure 6: Mean TP rates, using pairs with $|i - j| > 4$, taken over all families, a) without and b) with adjustment for varying domain lengths. For each method, the five best regularization choices are shown.

Mean TP rates

Figure 6a shows the mean TP rates over all 17 families. Since the number of pairs differs quite a bit between families (1128–8385), the mean TP rate at ‘top p pairs’ might be a skewed way to measure. We therefore also present fig. 6b: the mean TP rate as a function of p/N instead of p . PLM’s curves clearly hover higher than all those of NMFI. For $N/2$ predicted pairs (50% in figure 6b), for example, PLM identifies about 90% contacts and NMFI about 80%.

So, this first survey suggests PLM indeed offers some interesting improvements. We should not be too hasty in attributing this to pseudolikelihoods, however, since both regularization and scoring differ between the methods (we return to this soon).

Now follows a more in-depth breakdown of the output. We leave out results for NMFI-DI(true) and focus on PLM vs. NMFI-DI (the version used by Morcos et al. (2011)). All plots remaining in section 5.4 use the optimal regularization values: $\lambda = B_{eff}$ for NMFI and $\lambda_J = 0.01$ for PLM.

Score vs. distance

TP rates only classify pairs as contacts ($d(i, j) < 8.5\text{\AA}$) or noncontacts ($d(i, j) \geq 8.5\text{\AA}$). To give a more detailed view of how score correlates with spatial separation, we show in fig. 7 distance distributions for all top-30 ranked nonneighbor pairs. PLM and NMFI-DI both manage to raise the two peaks seen in the true distance distribution of fig. 4b. The peak at $3\text{--}5\text{\AA}$ clearly dominates the top-30, especially according to PLM’s read. As expected from its lesser mean TP rate at $p = 30$, NMFI-DI shifts more of its top-30 onto distant pairs.

Figure 8 shows score vs. distance for all pairs in all families. Both methods

agree that interactions get progressively weaker going from peak one, to two, to three (in order as they appear in fig. 4a). Note that the dots splash differently across the PLM and NMFI-DI figures, probably reflecting that two separate scoring techniques are being used. We can see here how sparse the signal we are seeking to extract is; most close pairs do not show up statistically coupled. Conversely though, almost all strongly coupled pairs are close, so the biological hypothesis of chapter 2 is well supported here.

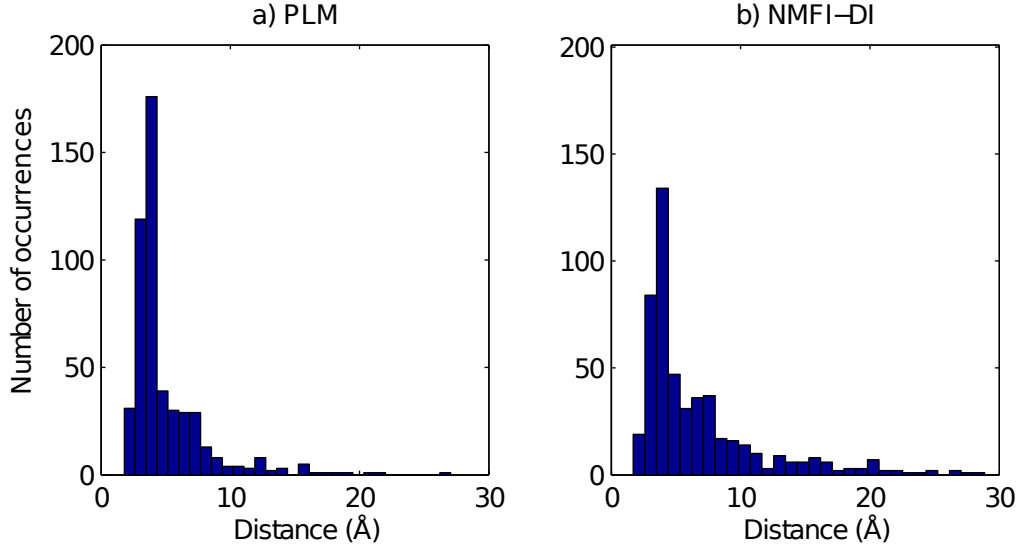


Figure 7: Histograms of crystal-structure distances for the collection obtained by joining the top-30 ranked $|i - j| > 4$ -pairs from each family.

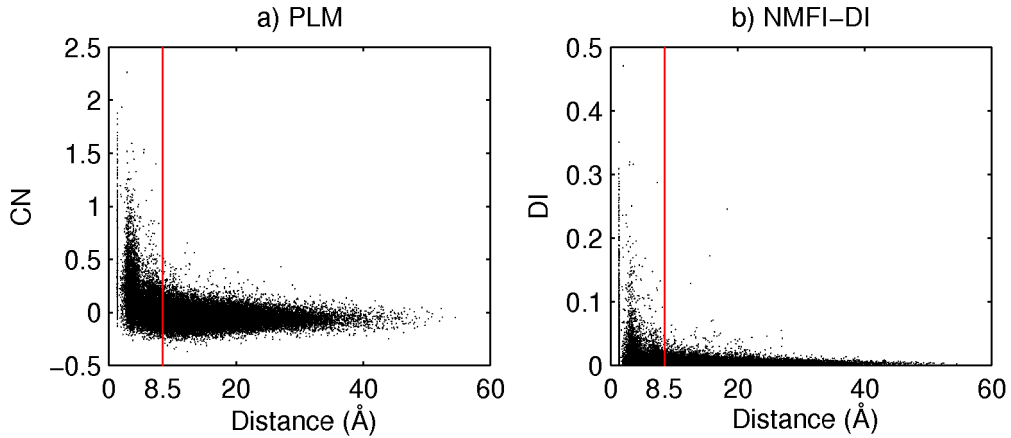


Figure 8: Score plotted against distance for all pairs in all 17 families. The red line is our contact cutoff at 8.5Å.

Scatter plots

Figure 9 shows scatter plots for PLM and NMFI-DI for some selected families⁵. The points assemble around a vaguely linelike shape, so, to some extent, PLM and NMFI-DI agree on the interaction strengths. Many of PLM’s top contacts are also top contacts for NMFI-DI and vice versa. A striking contrast, though, is how much stronger NMFI-DI responds to pairs with small $|i - j|$; the blue crosses tend to shoot out to the right. PLM agrees that many of these neighbor pairs interact strongly, but, unlike NMFI-DI, it also shows rivaling strengths for many $|i - j| > 4$ -pairs. It appears that NMFI’s ‘neighbor enthusiasm’ might somewhat derange its judgment of the more interesting $|i - j| > 4$ -pairs.

Contact maps

Another way to visualize the comparative performance of the two methods is *contact maps*, shown in fig. 10. The tendency observed in the scatter plots remains: NMFI-DI has a larger portion of highly scored pairs in the neighbor zone (the middle stretch of the figures). Clusters of strongly interacting neighbors are caught by both algorithms, but PLM marks such sections using fewer pairs. NMFI-DI displays somewhat loopy behavior in these regions; where PLM, for instance, identifies pairs of the type $(1, 2)$, $(2, 3)$, and $(3, 4)$, NMFI-DI tends to in addition include pairs like $(1, 3)$, $(1, 4)$, and $(2, 4)$, which could be argued to be somewhat redundant.

Circle plots

To get a sense of how false positives distribute across the domains, we draw interactions into circles in fig. 11. Some loopy attitudes are observed among the erroneously claimed contacts, especially for NMFI-DI; the blue lines tend to ‘bounce around’ in the circles. It seems that relatively few nodes are responsible for many of the false positives. We performed an explicit check of the data columns belonging to these ‘bad’ nodes, and we found that they often contained strongly biased data, i.e., had a few large $\mathbf{f}_i(k)$. In such cases, it seemed that NMFI-DI was quicker than PLM to ‘cry interaction’.

⁵Some figures (the scatter plots, contact maps, and circle plots) show results from a few families only. However, the patterns brought up here were observed for all families.

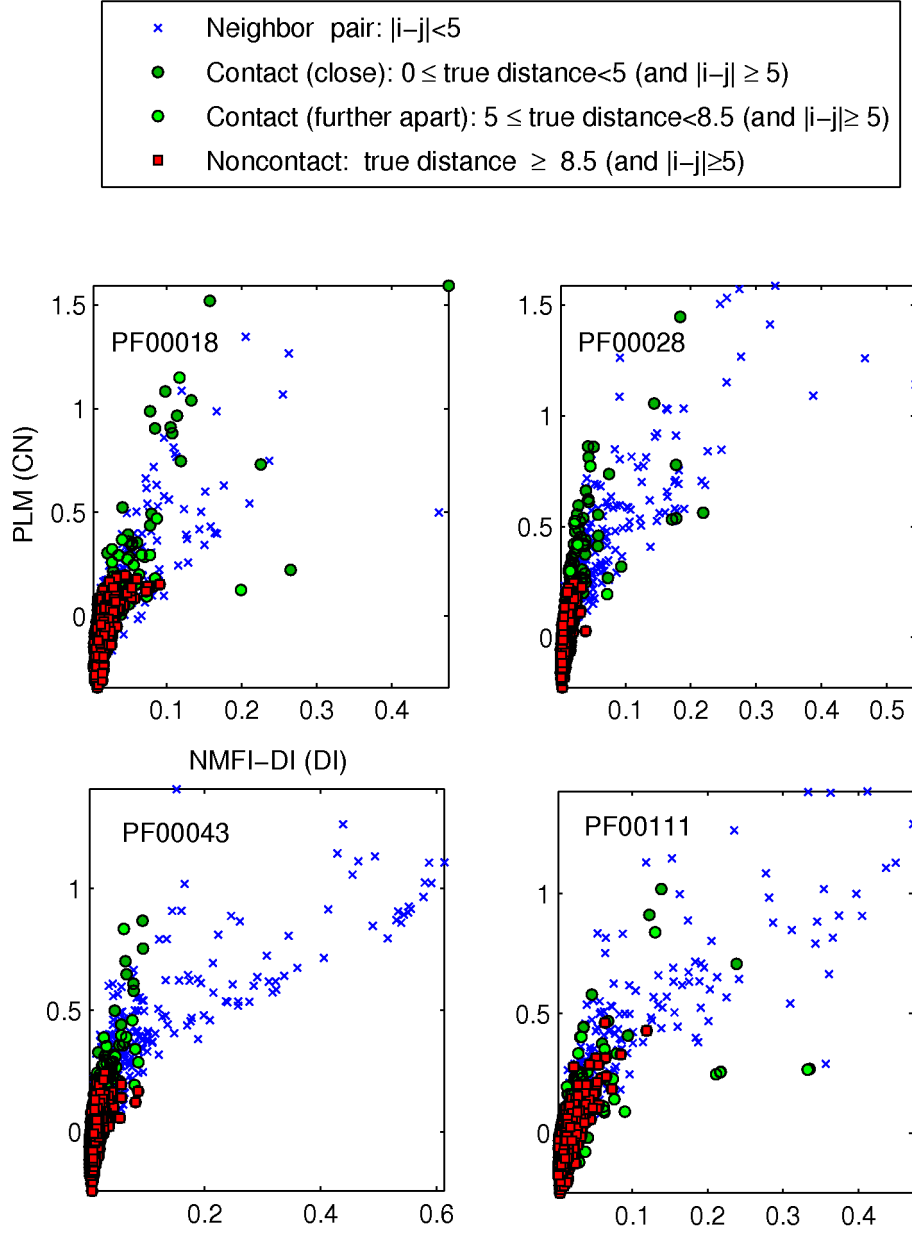


Figure 9: Scatter plots of interaction scores for PLM and NMFI-DI from four families. For all plots, the axes are as indicated by the top left one. The distance unit in the top box is Å.

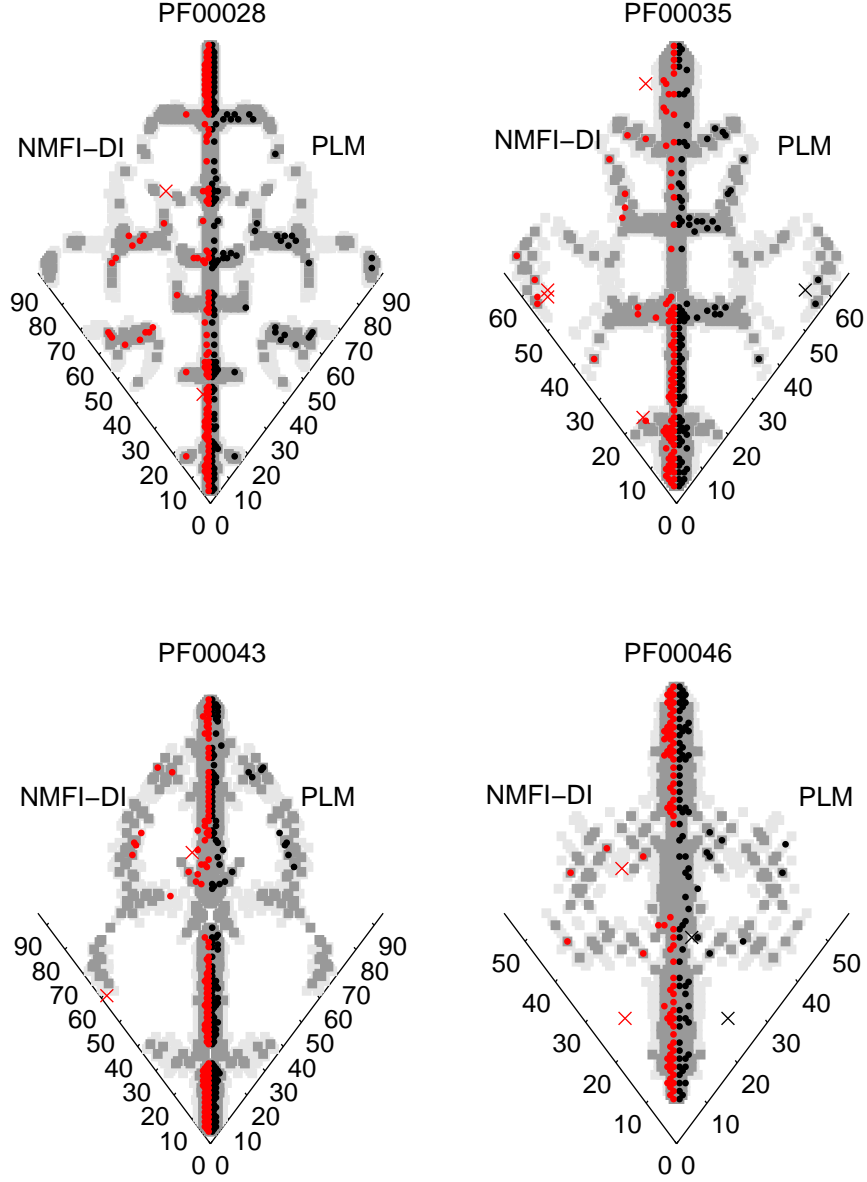


Figure 10: Contact maps for PLM and NMFI-DI from four families. A pair (i, j) 's placement in the plots is found by matching positions i and j on the axes. Contacts are indicated by gray (dark for $d(i, j) < 5\text{\AA}$ and light for $5\text{\AA} \leq d(i, j) < 8.5\text{\AA}$). True and false positives are represented by circles and crosses, respectively. Each figure shows the $1.5N$ strongest ranked pairs (including neighbors) for that family.

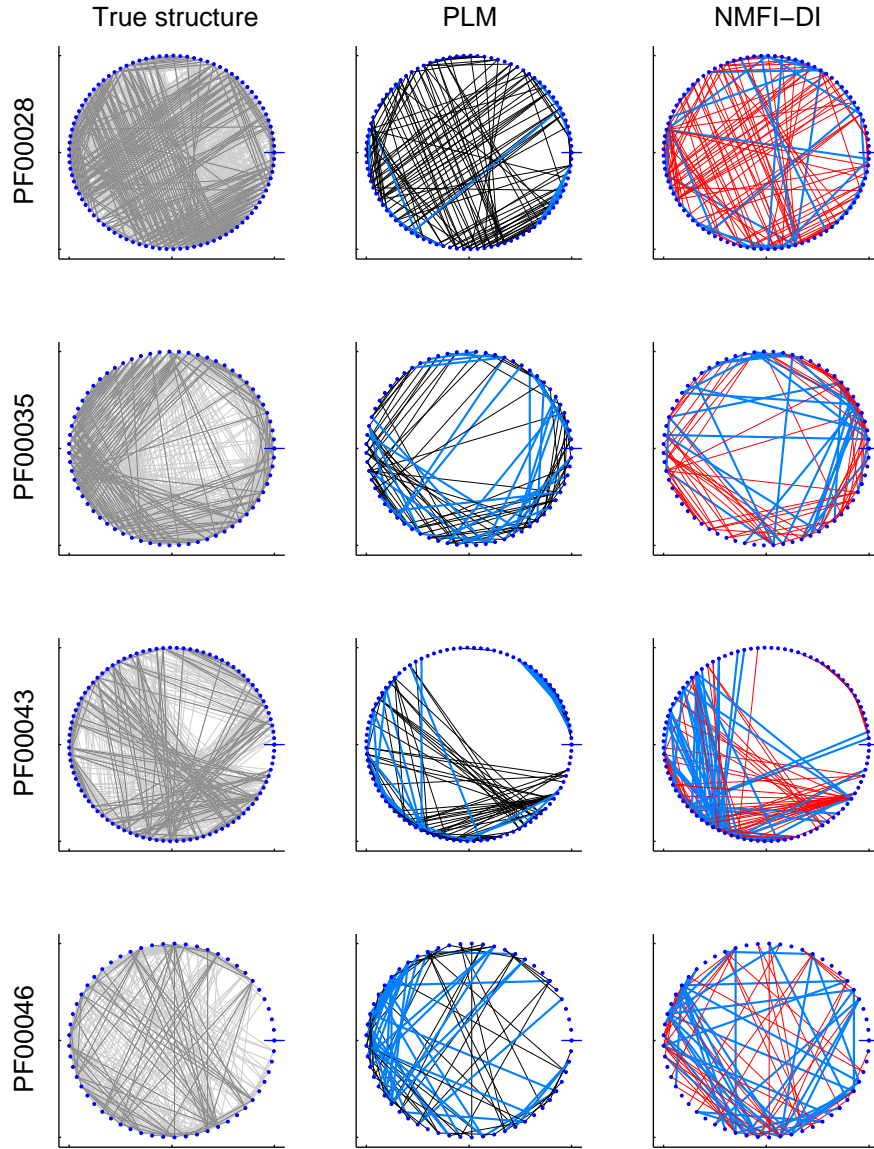


Figure 11: Connections for four families overlaid on circles. Position '1' is indicated by a dash. The leftmost column shows contacts in the crystal structure (dark gray for $d(i, j) < 5 \text{ \AA}$ and light gray for $5 \text{ \AA} \leq d(i, j) < 8.5 \text{ \AA}$). The other two columns show the top $1.5N$ strongest ranked $|i - j| > 4$ -pairs for PLM and NMFI, with black/red for true positives and blue for false positives.

This concludes the comparison between PLM and the original NMFI-DI of Morcos et al. (2011). Our PLM program does offer a boost in TP rates. It seems as if the separation in performance might be traceable to NMFI’s fixation on neighbor pairs. It could be debated whether this fixation is a weakness or not; neighbors are close in space, after all. Indeed, recreating fig. 5 with *all* pairs (data not shown) shows a much more even race between NMFI and PLM, where NMFI mainly capitalizes on the many $|i - j| \leq 4$ -pairs it correctly classifies as contacts.

Moreover, none of this can be confidently attached to the Potts inversion yet, since scoring and regularization differ. The split in performance could, for example, be entirely thanks to the APC.

5.5 Other scores for naive mean-field inversion

We also attempted to raise NMFI performance by using the APC term for the DI measurements. In addition, we tried the CN score for NMFI (first switching the parameter constraints from (6) to (7) using (49)). Mean TP rates using the old and new scores are shown in fig. 12. APC increases TP rates slightly but not enough to touch the PLM curves in fig. 6. Interestingly, the CN measurement gives the best results (except for $p = 2$).

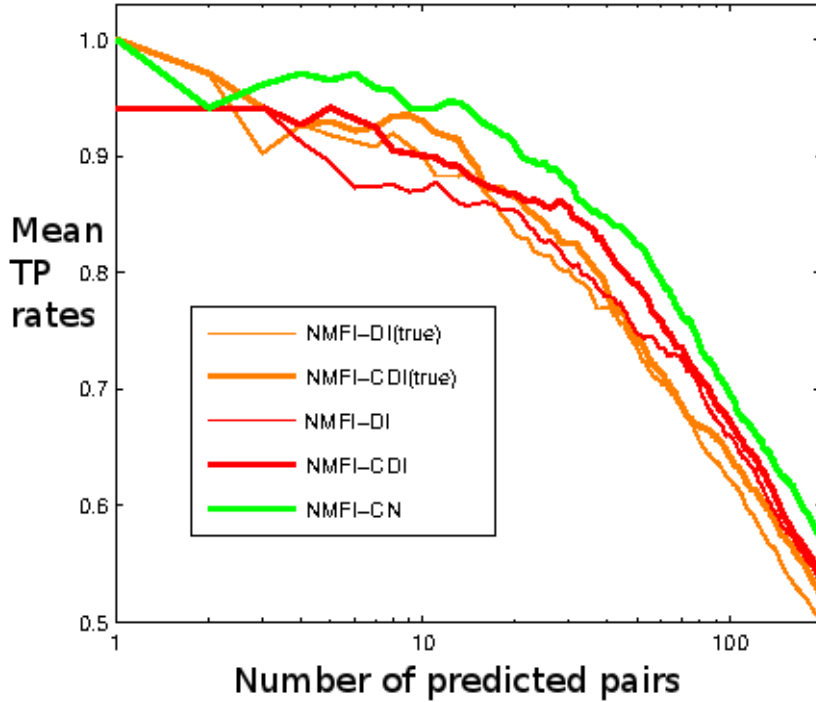


Figure 12: Mean TP rates, using pairs with $|i - j| > 4$, for NMFI with old scores DI and DI(true), new APC scores CDI and CDI(true), and the norm score CN. Each curve corresponds to the best λ for that particular score.

5.6 Comparison using a common score

Motivated by the results of fig. 12, we decided to match NMFI and PLM under the CN score. All figures in this section show the optimal regularization for each method, unless said otherwise. Figure 13 shows score vs. distance for all $|i - j| > 4$ -pairs in all families. Unlike fig. 8a–b, the two plots now show very similar profiles. Note, however, that NMFI’s CN scores are consistently two to three times larger than PLM’s (the scales on the vertical axes are different). Perhaps this is inherent within the methods, or simply a consequence of the different regularization types (for which the optimal strength may be different).

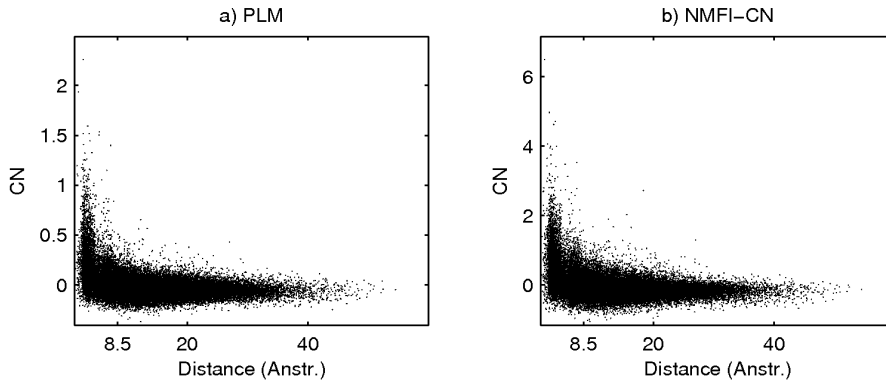


Figure 13: Score plotted against distance for all $|i - j| > 4$ -pairs in all 17 families.

TP rates

Figure 14 recreates fig. 5 with CN as a measure for NMFI. The three best regularization choices for NMFI-CN turned out the same as before, i.e. $\lambda = 1 \cdot B_{eff}$, $\lambda = 1.5 \cdot B_{eff}$ and $\lambda = 2.3 \cdot B_{eff}$, but the best out of these three was $\lambda = 2.3 \cdot B_{eff}$ (instead of $\lambda = 1 \cdot B_{eff}$). NMFI here performs closer to PLM; for several families, the prediction quality is pretty much the same for both methods. Still, PLM maintains slightly higher TP rates overall.

Scatter plots

Figure 15 shows renewed scatter plots for the families of fig. 9. This time, using CN for NMFI, the points are more clearly administered along a line, so the bends in fig. 9 must have been a consequence of differing scores. Yet, the trends seen in fig. 9 stay: NMFI gives more attention to neighbor pairs than PLM does.

Contact maps

In fig. 16 we recreate the contact maps of fig. 10 with NMFI-CN in place of NMFI-DI. The plots are more symmetric now. As expected, asymmetry is seen primarily for small $|i - j|$; NMFI tends to crowd these regions with lots of loops.

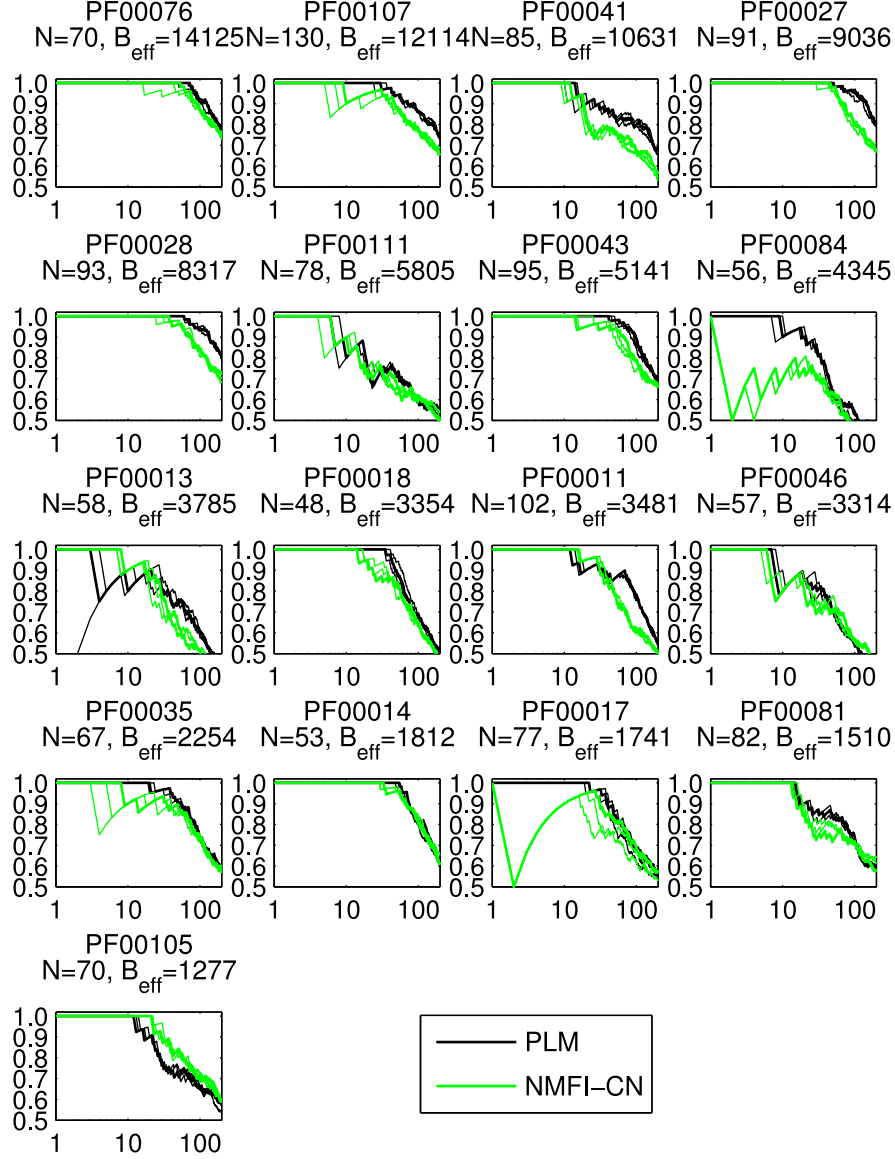


Figure 14: Contact-detection results for all the families in our study (sorted by B_{eff}), now with the CN score for NMFI. Y-axes are TP rates and x-axes are the number of predicted contacts p , based on pairs with $|i - j| > 4$. The three curves for each method are the three regularization levels yielding highest TP rates across all families. The thickened curve highlights the best one out of these three ($\lambda = 2.3 \cdot B_{eff}$ for NMFI-CN and $\lambda_J = 0.01$ for PLM).

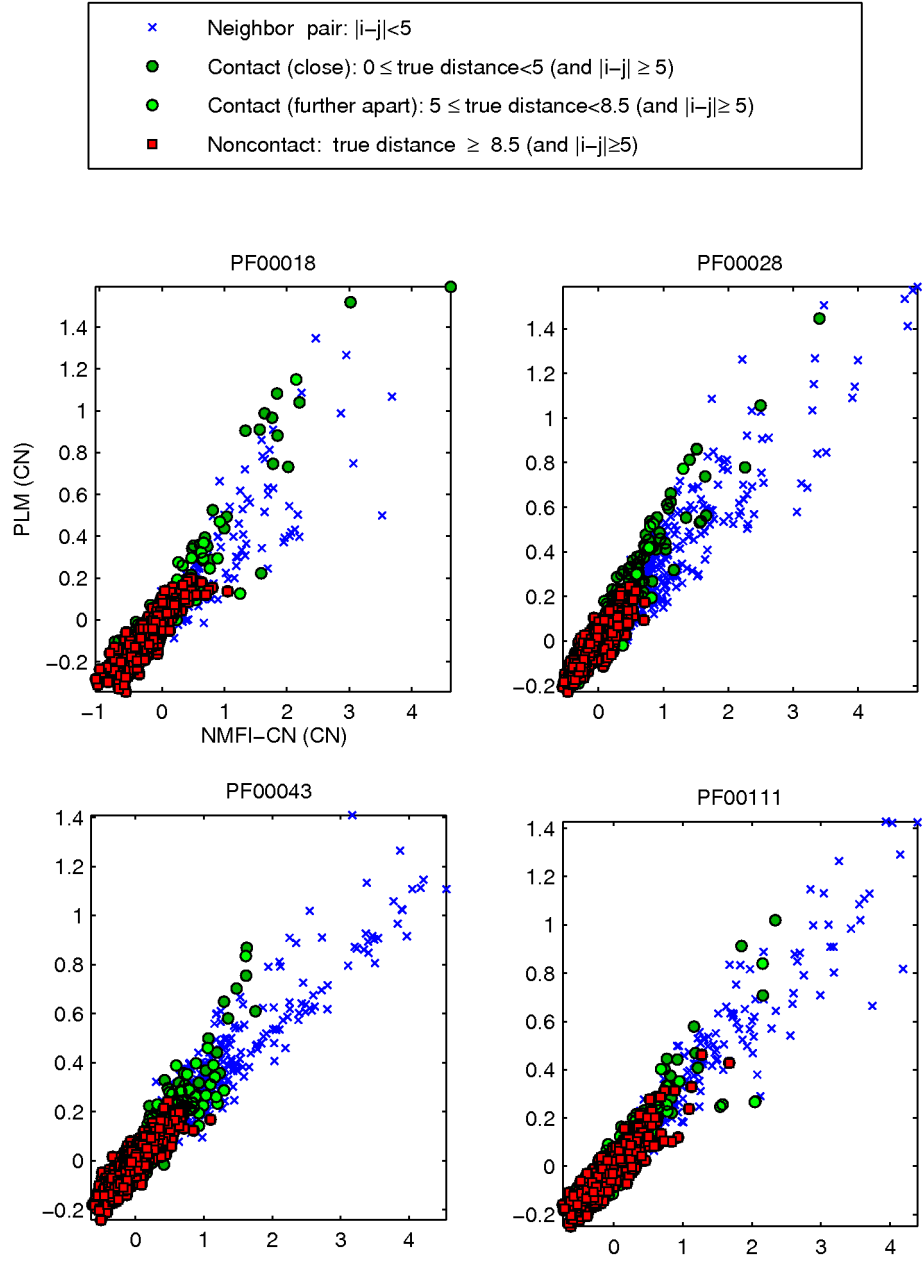


Figure 15: Scatter plots of interaction scores for PLM and NMFI-CN from four families. For all plots, the axes are as indicated by the top left one. The distance unit in the top box is Å.

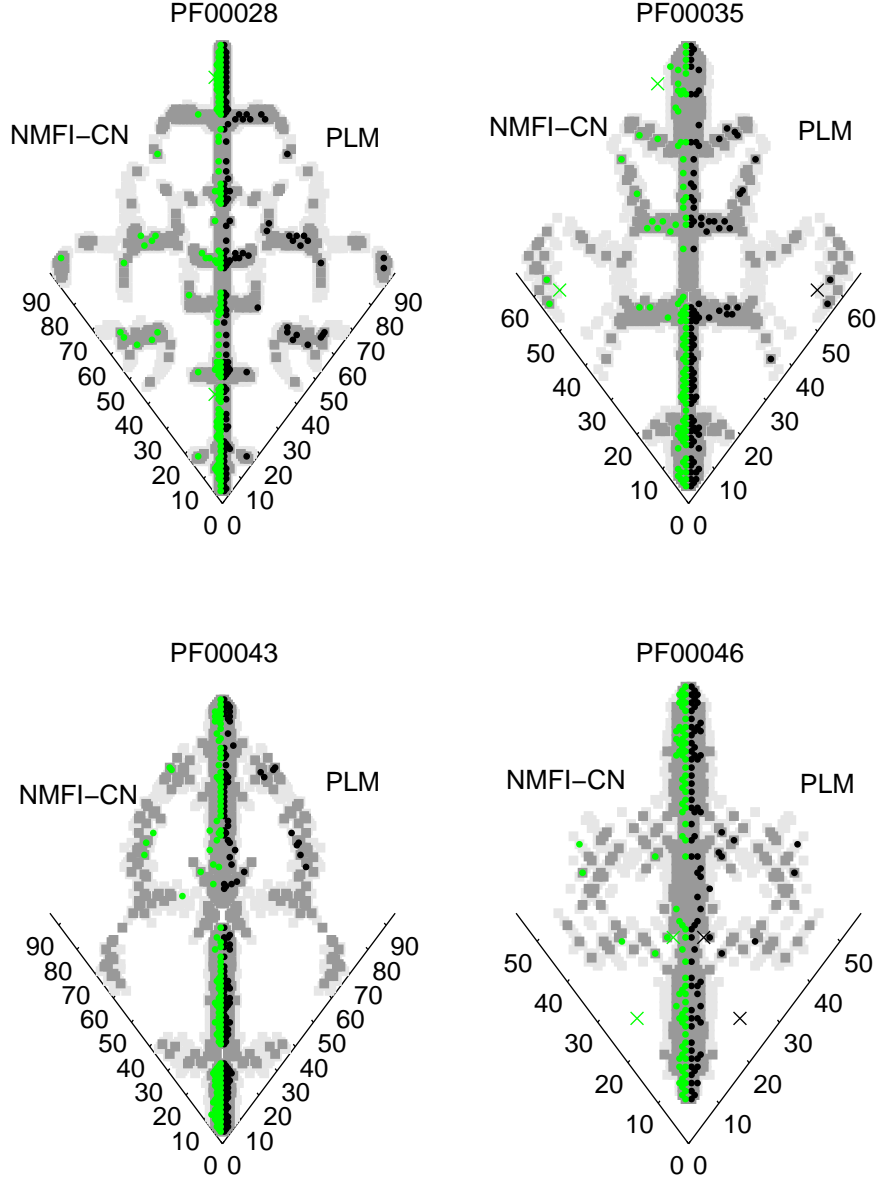


Figure 16: Contact maps for PLM and NMFI-CN from four families. A pair (i, j) 's placement in the plots is found by matching positions i and j on the axes. Contacts are indicated by gray (dark for $d(i, j) < 5\text{\AA}$ and light for $5\text{\AA} \leq d(i, j) < 8.5\text{\AA}$). True and false positives are represented by circles and crosses, respectively. Each figure shows the $1.5N$ strongest ranked pairs (including neighbors) for that family.

Gap-gap interactions

To investigate why NMFI assembles so many top-scored pairs in certain neighbor regions, we performed an explicit check of the associated MSA columns. A relevant regularity was observed: when gaps appear in a sequence, they tend to do so in lengthy strands. It tends to look something like the following made up MSA (in our implementation, the gap state is 1):

```

... 6 5 9 7 2 6 8 7 4 4 2 2 ...
... 1 1 1 1 1 1 1 1 1 1 2 8 ...
... 6 5 2 7 2 3 8 9 5 4 2 3 ...
... 3 7 4 7 2 6 8 7 9 4 2 3 ...
... 3 7 4 7 2 3 8 8 9 4 2 9 ...
... 1 1 1 1 1 1 1 4 5 4 2 9 ...
... 8 5 9 7 2 9 8 7 4 4 2 4 ...
... 1 1 1 1 1 1 1 1 1 1 2 4 ...

```

These injections are necessary for satisfactory alignment (as described in section 2.3) and could perhaps explain the split behavior of our two methods. This type of pattern is bound to induce large $\mathbf{J}_{ij}(1,1)$ for some pairs with small $|i-j|$. Remember that we treat gaps as just another amino acid, with associated interaction parameters.

Figure 17 shows scatter plots for all coupling parameters $\mathbf{J}_{ij}(k,l)$ in PF00014, which has a modest amount of gap sections, and in PF00043, which has relatively many. As suspected, the $\mathbf{J}_{ij}(1,1)$ -parameters are among the largest in magnitude, especially for PF00043. Note how the red dots steer to the right; NMFI clearly reacts harder to the gap-gap interactions than PLM.

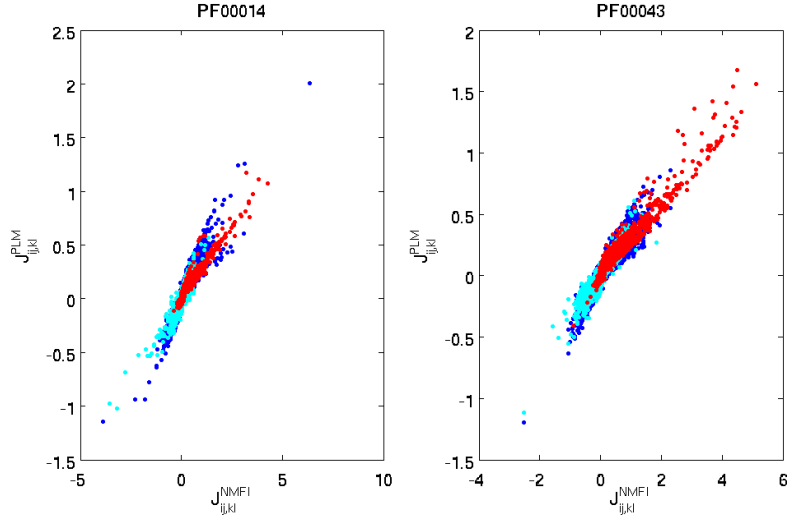


Figure 17: Scatter plots of estimated $J_{ij,kl} = \mathbf{J}_{ij}(k,l)$ from PF00014 and PF00043. Red dots are 'gap-gap' interactions ($k = l = 1$), turquoise dots are 'gap-amino-acid' interactions ($k = 1$ and $l \neq 1$, or $k \neq 1$ and $l = 1$), and blue dots are 'amino-acid-amino-acid' interactions ($k \neq 1$ and $l \neq 1$).

Jones et al. (2012) ignored contributions from gaps in their scoring by simply skipping the gap state when doing their norm summations. We tried this but found no significant improvement for either method. The change seemed to affect only pairs with small $|i - j|$ (which is reasonable), and our TP rates are based on pairs with $|i - j| > 4$. If gap interactions are indeed responsible for reduced prediction qualities, removing their input during scoring is just a band-aid type solution. A better way would be to suppress them already in the parameter estimation step. That way, all interplay would have to be accounted for without them. Whether or not gaps *should* be recognized as normal amino acids is an issue which goes somewhat beyond the biological scope of this work, however.

Everything shown so far was done with reweighting margin $x = 0.9$. Perhaps the gap effect can be dampened by stricter definition of sequence uniqueness? We show in fig. 18 another bunch of TP rates, now for $x = 0.75$. We also include results for NMFI run on alignment files from which all sequences with more than 20% gaps have been removed. The best regularization choice for each method turned out the same as in fig. 14: $\lambda = 2.3 \cdot B_{eff}$ for NMFI and $\lambda_J = 0.01$ for PLM. Overall, PLM keeps the same modest advantage over NMFI it had in fig. 14. Removing gappy sequences seems to trim down more TP rates than it raises, probably since useful information in the nongappy parts is discarded unnecessarily.

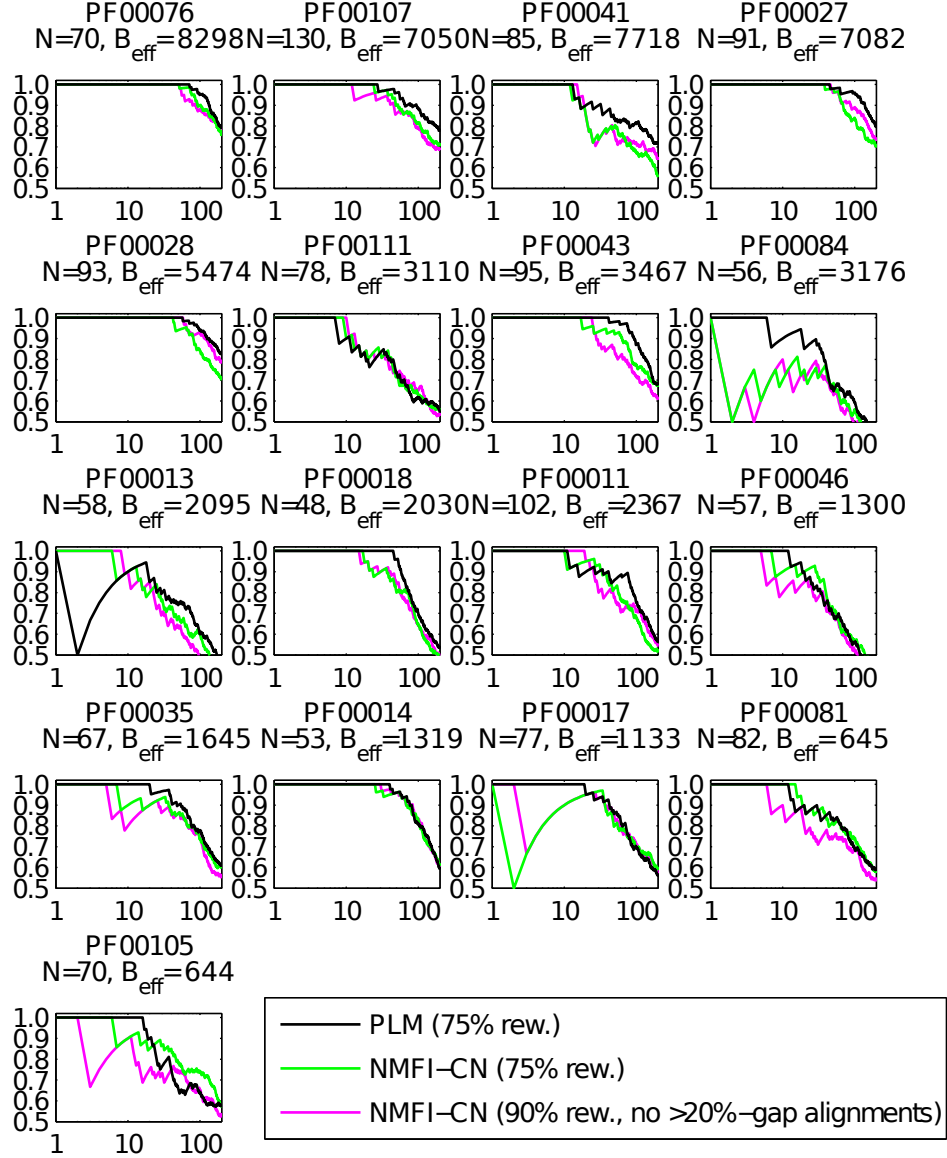


Figure 18: Contact-detection results for all the families in our study. Y-axes are TP rates and x-axes are the number of predicted contacts p , based on pairs with $|i-j| > 4$. The black and green curves are for reweighting margin $x = 0.75$, and the purple curve is for reweighting margin $x = 0.9$ after deletion of all sequences with more than 20% gaps. The curve for each method corresponds to the regularization level yielding highest TP rates across all families ($\lambda = 2.3 \cdot B_{eff}$ for both NMFI-CN and $\lambda_J = 0.01$ for PLM).

5.7 Running times

Since PLM has an objective function which accumulates additional terms for each sample, its execution time is heavily dependent on B . On one core of a standard desktop computer, running times using l_2 -regularization with $\lambda_J=0.01$ for PF00014, PF00017, and PF00018 — all with relatively small B — were 9, 22, and 12 minutes respectively. Using a L-BFGS method, which was faster but heavier on the RAM than CG, the same jobs took 5, 14, and 7 minutes. For PF00041, which had larger B (and N), one run using CG took 2.5 hours.

The matrix inversion of NMFI does not depend on the amount of samples used to compute the frequencies. In general, NMFI is very quick; most families in this study took only seconds to run through the NMFI code.

Possible speed-ups

Our use of PLM was geared more toward reliability than speed. For instance, we demanded changes in the seventh decimal or so before terminating the descent. Much time was therefore spent polishing the last decimals, providing only superficial accuracy in the end. Relaxing the precision properly could cut a significant chunk of the running time. Furthermore, we cold-started with all fields and couplings at zero, which is not very efficient. A clever starting guess using the frequencies, or perhaps the NMFI estimates, would further reduce the convergence time. These relatively simple tweaks were not necessary for our 17 small families, but they could become important for domains with larger N . If the scaling with N becomes really problematic, a more drastic way to tame it could be to allow only a predefined number of pairs to participate in the learning (perhaps top-MI pairs), as done by Weigt et al. (2009). Support for this is already included in the package by M. Schmidt.

If one were to reimplement PLM with speed as a priority, one should probably consider the broken up variant described in section 4.2. It divides the problem into N (multiclass) logistic regression problems which are completely open to simultaneous solving.

Other ways to reduce the minimization time are of course also possible.

5.8 Concluding remarks

With that, we terminate the comparison of NMFI with l_2 -regularized PLM. So, what is the final verdict? Certainly, the systematic increase in TP rates seen in fig. 5 is worthy of some attention; our new program seems to bring 'something' to the table. We managed to tie part of PLM's lead to the CN score. Indeed, this new, easy-to-calculate score is a relevant find in itself: CN consistently outputs higher TP rates than the more cumbersome DI, at least for our 17 (small) families. In fact, the accuracy jump from DI to CN is as big as (if not bigger than) the remaining jump from NMFI to PLM.

Under the same score, the methods depart under extreme circumstances, such as clustered neighbor interactions (induced by gap chains in the data) or extreme node-bias. In these cases, PLM stands out as a guarded alternative to the quick, zealous NMFI. That said, the agitative impulses of NMFI may be thwartable by clever modifications to the inversion, or even by just substituting the pseudocount for a penalty term. To linger on that point, we remark that

it is, in principle, possible that the pseudolikelihood/mean-field choice matters less than the style of regularization; we have not matched the methods under the same regularization type.

5.9 l_2 - vs. group l_1 -regularization

We end with a quick analysis of some group l_1 -regularization trials which we ran to see if obvious improvements were to be had by switching from R_{l_2} to R_{gl_1} . These trials were done using $\lambda_h = 0.01$, $x = 0.9$, and the same type of scan over λ_J as before. We restricted ourselves to a subset of six families with relatively small N and B . Figure 19 shows the results. For PF00105 we see significant improvement: group l_1 -regularization has a TP rate of one up to about $p = 25$ compared with about $p = 12$ for l_2 -regularization. For PF00014 and PF00035, we instead see a drop in TP rate using the new regularization. We cannot say that this clearly beats the R_{l_2} -results.

The λ_J which gave the highest (average) TP rate for R_{gl_1} was 0.01. Under this strength, solutions were not noticeably sparse. This hints that we're not seeing the real potential of R_{gl_1} here; edgewise sparsity is supposed to be an appealing consequence of grouping the couplings. Perhaps the TP rate is ill-placed as a measurement when the focus is on plausible structure (rather than precise interaction strengths). With this in mind, we ignored the dropping TP rates and increased λ_J until we saw pronounced sparsity. Figure 20 shows scatter plots for $\lambda_J = 0.13$. Most scores are forced to zero (or slightly below zero because of the APC). As far as prediction quality goes, these plots send somewhat mixed messages: while contacts seem more reluctant to go to zero than noncontacts, there are also bunches of contacts which R_{gl_1} suppresses which are highly ranked under R_{l_2} (note the strings of green dots on the rightmost part of the 'zero-lines').

The study by Balakrishnan et al. (2011), which we came upon during the course of this work, goes all out in exploiting the sparsity pressure of R_{gl_1} . In contrast to our 'manual' pick, they engage in a rigorous λ_J -calibration which commits to a structure using the data. Perhaps higher TP rates *would* emerge if R_{l_2} -estimation was executed on top of the R_{gl_1} -inferred structure skeleton (this is also mentioned in their paper). Their package, called *GREMLIN* (Generative REgularized ModeLS of proteINs), can be run through their webserver (<http://langmeadlab.org/gremlin.php>).

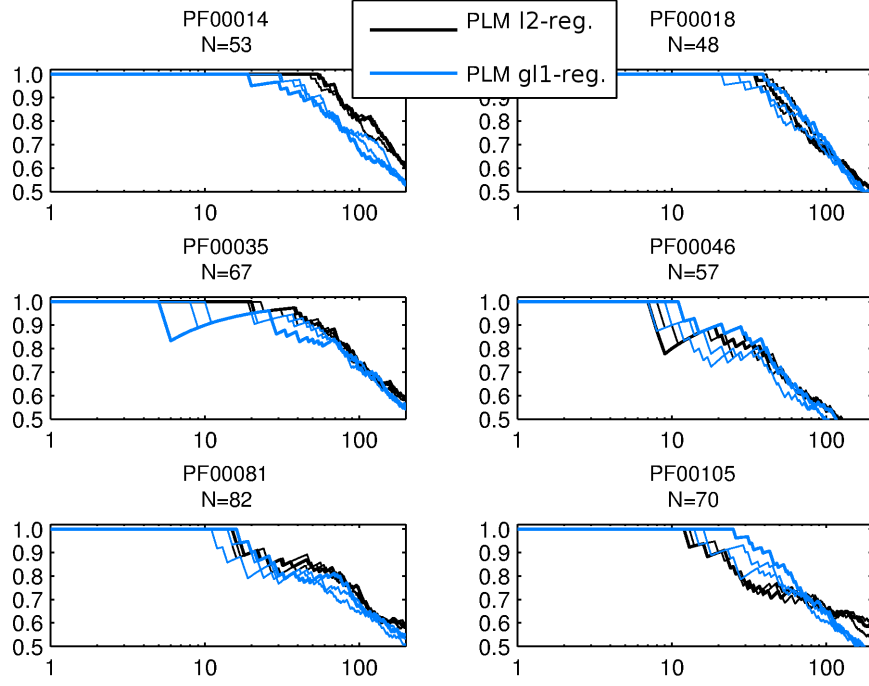


Figure 19: Contact-detection results for six families using PLM with group l_1 -regularization and l_2 -regularization. Y-axes are TP rates and x-axes are the number of predicted contacts p , based on pairs with $|i - j| > 4$. The three curves for each method are the three regularization levels yielding highest TP rates across the six families. The thickened curve highlights the best one out of these three ($\lambda_J = 0.01$ for both R_{l_2} and R_{gl_1}).

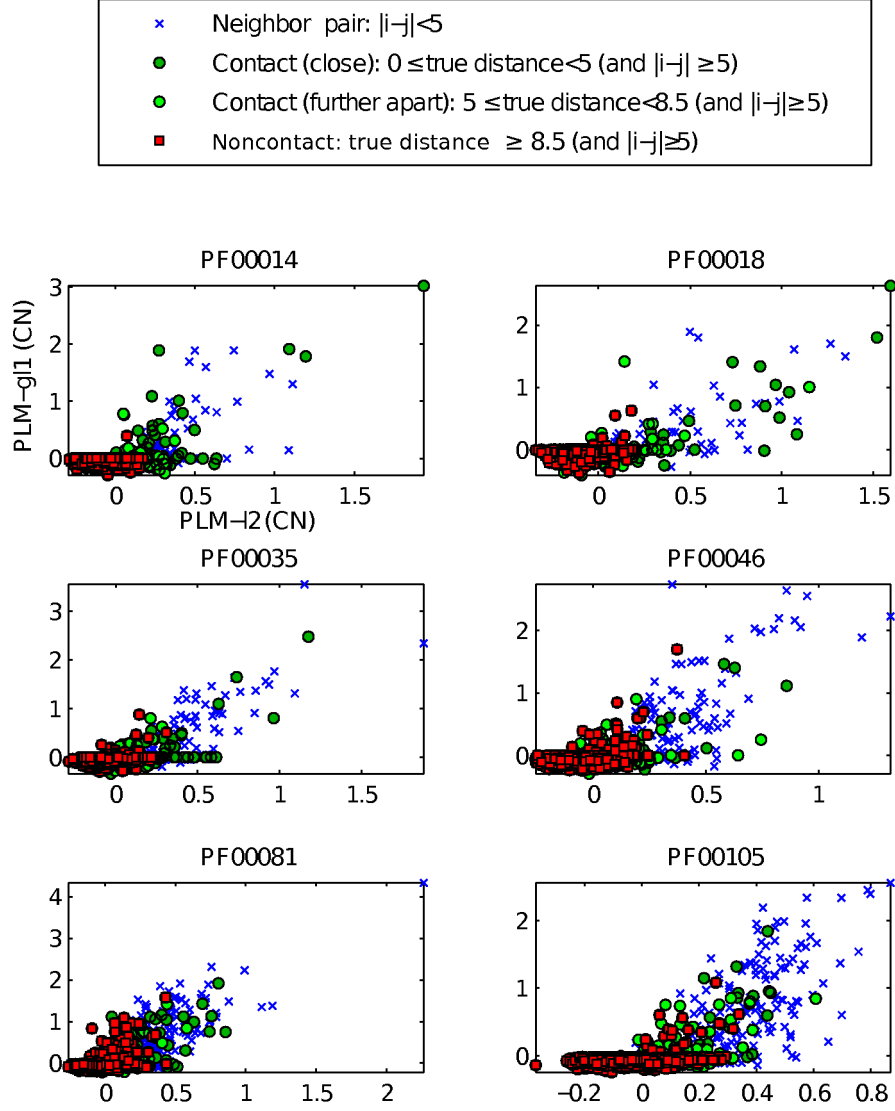


Figure 20: Scatter plots of interaction scores for six families using PLM with group l_1 -regularization ($\lambda_J = 0.13$) and l_2 -regularization ($\lambda_J = 0.01$). For all plots, the axes are as indicated by the top left one. The distance unit in the top box is Å.

6 Summary and possible future work

In this thesis, we have broken apart a freshly conceived, mean-field based protein-contact detector and reassembled it to run on maximized pseudolikelihoods. We have also put to use a new score, CN, for ranking interconnection strengths. For a batch of small-sized domain families from Pfam, our new routine gave predictions (systematically) more precise than those of the mean-field approach, an enhancement which we could partially attribute to the new score. After updating the mean-field code to also score by CN, our pseudolikelihood variant still maintained slight advantages. These were tied to slices of data deviating heavily from the model assumptions (such as sequences with gap-trains tens of positions long).

It is hard to argue that PLM should replace NMFI in general. If an analysis is tarnished by network-transmitted correlations, NMFI can give incredibly rapid causation/correlation disentanglement which likely makes the substantial changes that really matter (if the particular application is ripe for it). For example, the improvements reported here are much smaller than those seen when going from MI to NMFI (as in Morcos et al., (2011)). Perhaps PLM's main role is to serve as a fallback routine to be brought in if NMFI, for one reason or another, breaks down or performs poorly (as we have seen happen). As the methods stand, PLM seems to offer a fair trade-in of speed for reliability, but an exact zone of benefit is difficult to size up.

An extension of this project could be letting an experienced numerical programmer attempt a faster realization of PLM, perhaps as laid out in section 5.7. This could be used to, for example, tackle all the 131 families used by Morcos et al. (2011). It is, at present, unclear whether or not our positive results extend outside the regime of small families. Another route going forward could be for someone biologically schooled to implant more PSP specific operations into the program. Several are imaginable (these overlap a bit, though): using cluster considerations to treat whole groups of interacting positions (as mentioned by Morcos et al. (2011)), theoretically backed special treatment of gap states, and more advanced reweighting techniques, to name but a few. Also, because choice of score seems as potent in its capacity to influence as choice of Potts inverter (at least in this work), trying many scores and sorting out exactly what they capture/miss would be interesting as well.

References

- D.H. Ackley, G.E. Hinton, and T.J. Sejnowski, *A learning algorithm for Boltzmann machines*, Cognitive Science **9**, 147 (1985).
- E. Aurell and M. Ekeberg, *Inverse Ising inference using all the data*, Physical Review Letters **108**, 090201 (2012).
- S. Balakrishnan, H. Kamisetty, J.G. Carbonell, S.I. Lee, and C.J. Langmead, *Learning generative models for protein fold families*, Proteins: Structure, Function, and Bioinformatics **79**, 1061 (2011).
- J. Besag, *Statistical analysis of non-lattice data*, The Statistician **24**, 179 (1975).
- T. Broderick, M. Dudik, G. Tkacik, R.E. Schapire, and W. Bialek, *Faster solutions of the inverse pairwise Ising problem*, arXiv:0712.2437 (2007).
- L. Burger and E. van Nimwegen, *Disentangling direct from indirect co-evolution of residues in protein alignments*, Public Library of Science: Computational Biology **6**, e1000633 (2010).
- M.A. Carreira-Perpinan and G. Hinton, *On contrastive divergence learning*, Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTATS 2005), Barbados (2005).
- S. Cocco, S. Leibler, and R. Monasson, *Neuronal couplings between retinal ganglion cells inferred by efficient inverse statistical physics methods*, Proceedings of the National Academy of Sciences of the United States of America **106**, 14058 (2009).
- S. Cocco and R. Monasson, *Adaptive cluster expansion for inferring Boltzmann machines with noisy data*, Physical Review Letters **106**, 090601 (2011).
- S. Cocco and R. Monasson, *Adaptive cluster expansion for the inverse Ising problem: convergence, algorithm and tests*, arXiv:1110.5416 (2011).
- G. Cross and A. Jain, *Markov random field texture models*, IEEE Transactions on Pattern Analysis and Machine Intelligence **5**, 25 (1983).
- S.D. Dunn, L.M. Wahl, and G.B. Gloor, *Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction*, Bioinformatics **24**, 333 (2008).
- S. Geman and D. Geman, *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*, IEEE Transactions on Pattern

Analysis and Machine Intelligence **6**, 721 (1984).

U. Göbel, C. Sander, R. Schneider, and A. Valencia, *Correlated mutations and residue contacts in proteins*, Proteins: Structure, Function, and Genetics **18**, 309 (1994).

A. Hyvärinen, *Connections between score matching, contrastive divergence, and pseudolikelihood for continuous-valued variables*, IEEE Transactions on Neural Networks **18**, 1529 (2007).

H. Höfling and R. Tibshirani, *Estimation of sparse binary pairwise Markov networks using pseudo-likelihoods*, Journal of Machine Learning Research **10**, 883 (2009).

E. Ising, *Beitrag zur theorie des ferromagnetismus*, Zeitschrift für Physik **31**, 253 (1925).

A. Jalali, P. Ravikumar, V. Vasuki, and S. Sanghavi, *On learning discrete graphical models using group-sparse regularization*, Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2011), Fort Lauderdale, FL, USA (2011).

D.T. Jones, D.W.A. Buchan, D. Cozzetto, and M. Pontil, *PSICOV: Precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments*, Bioinformatics **28**, 184 (2012).

R.P. Kindermann and J.L. Snell, *On the relation between Markov random fields and social networks*, The Journal of Mathematical Sociology **7**, 1 (1980).

A.S. Lapedes, B.G. Giraud, L.C. Liu, and G.D. Stormo, *Correlated mutations in protein sequences: phylogenetic and structural effects*, Proceedings of the AMS/SIAM Conference on Statistics in Molecular Biology, Seattle, WA, USA (1997).

T.R. Lezon, J.R. Banavar, M. Cieplak, A. Maritan, and N.V. Fedoroff, *Using the principle of entropy maximization to infer genetic interaction networks from gene expression patterns*, Proceedings of the National Academy of Sciences of the United States of America **103**, 19033 (2006).

C.D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*, MIT Press (1999).

D.S. Marks, L.J. Colwell, R. Sheridan, T.A. Hopf, A. Pagnani, R. Zecchina, and C. Sander, *3D protein structure predicted from sequence*, arXiv:1110.5091 (2011).

M. Mezard and T. Mora, *Constraint satisfaction problems and neural networks: a statistical physics perspective*, Journal of Physiology - Paris **103**, 107 (2009).

- F. Morcos, A. Pagnani, B. Lunt, A. Bertolino, D.S. Marks, C. Sander, R. Zecchina, J.N. Onuchic, T. Hwa, and M. Weigt, *Direct-coupling analysis of residue coevolution captures native contacts across many protein families*, Proceedings of the National Academy of Sciences of the United States of America **108**, E1293 (2011).
- D.W. Mount, *Bioinformatics: Sequence and Genome Analysis* 2nd edition, Cold Spring Harbor Laboratory Press (2004).
- H.C. Nguyen and J. Berg, *Bethe-Peierls approximation and the inverse Ising problem*, Journal of Statistical Mechanics: Theory and Experiment **2012**, P03004 (2012).
- R.B. Potts, *Some generalized order-disorder transformations*, Mathematical Proceedings of the Cambridge Philosophical Society **48**, 106 (1952).
- M. Punta, P.C. Coggill, R.Y. Eberhardt, J. Mistry, J. Tate, C. Boursnell, N. Pang, K. Forslund, G. Ceric, J. Clements, A. Heger, L. Holm, E.L.L. Sonnhammer, S.R. Eddy, A. Bateman, and R.D. Finn, *The Pfam protein families database*, Nucleic Acids Research **40**, D290 (2012).
- P. Ravikumar, M.J. Wainwright, and J.D. Lafferty, *High-dimensional Ising model selection using l1-regularized logistic regression*, Annals of Statistics **38**, 1287 (2010).
- F. Ricci-Tersenghi, *On mean-field approximations for estimating correlations and solving the inverse Ising problem*, arXiv:1112.4814 (2011).
- Y. Roudi, E. Aurell, and J.A. Hertz, *Statistical physics of pairwise probability models*, Frontiers in Computational Neuroscience **3** (2009).
- M. Schmidt, K. Murphy, G. Fung, and R. Rosales, *Structure learning in random fields for heart motion abnormality detection*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), Anchorage, AK, USA (2008).
- M. Schmidt, *Graphical model structure learning with l1-regularization*, PhD thesis, The University of British Columbia, Vancouver (2010).
- E. Schneidman, M.J. Berry, R. Segev, and W. Bialek, *Weak pairwise correlations imply strongly correlated network states in a neural population*, Nature (London) **440**, 1007 (2006).
- V. Sessak and R. Monasson, *Small-correlation expansions for the inverse Ising problem*, Journal of Physics A **42**, 055001 (2009).
- J. Sohl-Dickstein, P.B. Battaglino, and M.R. DeWeese, *New method for parameter estimation in probabilistic models: minimum probability flow*, Physical Review Letters **107**, 220601 (2011).

D.J. Thouless, P.W. Anderson, and R.G. Palmer, *Solution of 'Solvable model of a spin glass'*, Philosophical Magazine **35**, 593 (1977).

M.J. Wainwright and M.I. Jordan, *Graphical models, exponential families, and variational inference*, Foundations and Trends in Machine Learning **1**, 1 (2008).

M. Weigt, R.A. White, H. Szurmant, J.A. Hoch, and T. Hwa, *Identification of direct residue contacts in protein-protein interaction by message passing*, Proceedings of the National Academy of Sciences of the United States of America **106**, 67 (2009).

D.J.A. Welsh, *Complexity: Knots, Colourings and Counting*, Cambridge University Press (1993).