



Fast pseudolikelihood maximization for direct-coupling analysis of protein structure from many homologous amino-acid sequences



Magnus Ekeberg^{a,b,*,1}, Tuomo Hartonen^{c,d,1}, Erik Aurell^{b,c,e}

^a Engineering Physics Program, KTH Royal Institute of Technology, SE-100 77 Stockholm, Sweden

^b Department of Computational Biology, AlbaNova University Centre, 106 91 Stockholm, Sweden

^c Department of Information and Computer Science, Aalto University, PO Box 15400, FI-00076 Aalto, Finland

^d The Master's Degree Programme in Translational Medicine, Biomedicum Helsinki, FI-00014 University of Helsinki, Finland

^e Aalto Science Institute, PO Box 15600, FI-00076 Aalto, Finland

ARTICLE INFO

Article history:

Received 24 January 2014

Received in revised form 23 June 2014

Accepted 15 July 2014

Available online 25 July 2014

Keywords:

Protein structure prediction

Contact map

Direct-coupling analysis

Potts model

Pseudolikelihood

Inference

ABSTRACT

Direct-coupling analysis is a group of methods to harvest information about coevolving residues in a protein family by learning a generative model in an exponential family from data. In protein families of realistic size, this learning can only be done approximately, and there is a trade-off between inference precision and computational speed. We here show that an earlier introduced l_2 -regularized pseudolikelihood maximization method called plmDCA can be modified as to be easily parallelizable, as well as inherently faster on a single processor, at negligible difference in accuracy. We test the new incarnation of the method on 143 protein family/structure-pairs from the Protein Families database (PFAM), one of the larger tests of this class of algorithms to date.

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

A momentous challenge for research, companies, and society at large is how to use better and in novel ways vast swathes of accrued information, often referred to as “Big Data”. Such data can be collected and catalogued in many different ways, and then analyzed by different actors, potentially in new fashion to pursue very different objectives than for which the data was originally gathered. In this paper, we report on progress on one important example where data on homologous proteins,² collected by many research groups around the world, can be decoded to reveal amino-acid contacts within protein structures to very good accuracy. An existing pseudolikelihood maximization approach currently delivers higher accuracy

Abbreviations: PSP, Protein Structure Prediction; CASP, Critical Assessment of protein Structure Prediction; DCA, Direct-Coupling Analysis; PFAM, Protein Families database; SCOP, Structural Classification of Proteins; mFDCA, mean-field Direct-Coupling Analysis; plmDCA, pseudolikelihood maximization Direct-Coupling Analysis; MSA, Multiple Sequence Alignment; FN, Frobenius Norm; APC, Average Product Correction; CFN, Corrected Frobenius Norm; HMM, Hidden Markov-Model; PDB, Protein Data Bank; UNIPROT, Universal Protein Resource; NMR, Nuclear Magnetic Resonance; SIFTS, Structure Integration with Function, Taxonomy and Sequence; TPR, True-Positive Rate.

* Corresponding author at: Department of Computational Biology, AlbaNova University Centre, 106 91 Stockholm, Sweden.

E-mail address: ekeb@kth.se (M. Ekeberg).

¹ Joint first authors.

² In this paper, we use “protein” interchangeably with “protein domain”.

than other methods, but at the cost of longer running time. We here introduce a new version of this earlier method, and show that it yields predictions with practically identical precision, but with a large computational speed-up.

Protein Structure Prediction (PSP) aims to reap information about the three-dimensional structure of a protein from any suitable data, but in particular from its amino-acid sequence. Advances are regularly evaluated in the framework of CASP (The Critical Assessment of protein Structure Prediction) [1]. Although much progress has been made, the consensus opinion has become that *ab initio* PSP, i.e. predicting the three-dimensional structure of a protein from its amino-acid sequence only, is not feasible. On the other hand, homology PSP, i.e. predictions taking cues from known structures of proteins that are homologous, is often possible, although in many respects remaining an art.

Direct-Coupling Analysis (DCA) belongs to an intermediate level of PSP where predictions are made not from a single amino-acid sequence, but from the set of amino-acid sequences of a family of homologous proteins. The interest of this approach is at least twofold. First, the number of known protein structures grows at a much slower rate than the number of known amino-acid sequences – their ratio today being about 1 : 550 – and this can be expected to remain the case for the foreseeable future. Therefore, while today if a protein is a member of a family containing many homologues then very often at least one of the homologues has a known structure, this may be less and less likely to be true in the future. Second, it is of interest to know if the information contained not just in one amino-acid sequence, but in a whole family of sequences – usually evolutionary related and hence subject to the same evolutionary constraints – is sufficient to determine the three-dimensional structure. In fact, it has been known for almost 20 years that the evolutionary history leaves a trace in the correlations between amino acids at different positions along a protein which contains nontrivial information, see e.g. [2–4], but before DCA this information was not fully exploitable. PSP by DCA is thus, apart from its intrinsic scientific interest, also a showcase for Big Data and how it can be exploited to arrive at new useful knowledge checkpoints. For a broader review of coevolution analysis for elucidating protein structures, see e.g. [5].

This paper is organized as follows: in Section 2 we introduce DCA and review and summarize the main approaches used up to now. In Section 3 we then present the pseudolikelihood maximization approach in more detail, first the previous version presented in [6], and then the faster parallel version introduced here. In Section 4 we present the data (and extraction thereof) on which our analysis is based, and in Section 5 we compare the speed and accuracy of the two versions of the pseudolikelihood maximization, followed by extensive experiments on the new version. Finally, in Section 6 we discuss our results. Supplementary Information to this paper gives additional data on proteins used and a family-per-family view of performance.

2. A primer on direct-coupling analysis

Let us represent the amino-acid sequence of a protein as $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_N)$. We assume that we have a Multiple Sequence Alignment (MSA), which is a table $\{\sigma^{(b)}\}_{b=1}^B$ of such amino-acid sequences of B proteins that have been aligned to have a common length N . In this work we will limit ourselves to using MSAs obtained from the PFAM database [7,8]. We will discuss how such tables look in Section 4 below and here just observe that each row in the table will represent a protein, and each column a position in the sequence. At row b and position i we hence have a symbol $\sigma_i^{(b)}$ which can be one of the 20 naturally occurring amino acids or a “–”, representing a gap in the alignment. For a list of amino acids and the symbols and abbreviations representing them, see Table S1.

The essence of DCA is then to assume that the rows, i.e. our aligned homologous proteins, are independent events drawn from a Potts-model probability distribution,

$$P(\sigma) = \frac{1}{Z} \exp \left(\sum_{i=1}^N h_i(\sigma_i) + \frac{1}{2} \sum_{i,j=1}^N J_{ij}(\sigma_i, \sigma_j) \right), \quad (1)$$

and to use the interaction parameters J_{ij} as predictions of spatial proximity among amino-acid pairs in the protein structure. Interpreting the J_{ij} this way can be biologically justified as follows: it is well-known that the detrimental effects of a single-site mutation, that alone would impair the function of the protein, can be countered by a compensatory mutation at a nearby site. Consequently, short intra-domain position–position distances can, and do, show up as pairwise couplings among the columns in the table $\{\sigma^{(b)}\}_{b=1}^B$.

To avoid trivial overparameterization we will define $J_{ij}(k, l) = J_{ji}(l, k)$ if i and j are different and $J_{ij} = \mathbf{0}$ if $i = j$. The double sum in (1) hence goes over all unordered pairs of distinct positions along the columns in the table, i.e.

$$P(\sigma) = \frac{1}{Z} \exp \left(\sum_{i=1}^N h_i(\sigma_i) + \sum_{1 \leq i < j \leq N} J_{ij}(\sigma_i, \sigma_j) \right). \quad (2)$$

Throughout the paper, we will, unless otherwise specified, assume single position-indexes to run across $1 \leq i \leq N$, pairwise position-indexes to run as $1 \leq i < j \leq N$, and amino-acid indexes to span $1 \leq k \leq q$, where $q = 21$ (20 amino acids and one additional state for the alignment gap). Determining the J_{ij} from the observations $\{\sigma^{(b)}\}_{b=1}^B$ is a nontrivial inference problem, since for N large enough the normalization constant Z , the number of terms of which (q^N) grows exponentially with the protein length, cannot be computed efficiently and exactly. Let us note that if we would have a multidimensional Gaussian

model $P \sim \exp(-\frac{1}{2}\mathbf{x} \cdot \mathbf{M}\mathbf{x})$, then it is natural to consider the matrix elements M_{ij} as “causes” or “direct couplings”, in contrast to correlations which are given by the inverse matrix $(M^{-1})_{ij}$; two elements may be strongly correlated although not directly coupled if instead indirectly coupled through intermediaries. Analogous but computationally less elementary considerations should also pertain to the model in (2).

A bedrock principle of model learning in the frequentist’s interpretation of statistics is maximum likelihood, which means to minimize over a set of parameters θ the negative log-likelihood function:

$$L(\theta; \sigma) = -\log P(\sigma|\theta), \quad (3)$$

where σ are the observations which enter as parameters in the function on the left-hand side, and where we have defined L as minus the logarithm of P . If we have B independent observations from the same model it is customary to divide the negative log-likelihood function by B and work with $l = \frac{1}{B}L$. In our case, where P is given by (2), we have

$$\begin{aligned} l(\mathbf{h}, \mathbf{J}) &= -\frac{1}{B} \sum_{b=1}^B \ln \left[\frac{1}{Z} \exp \left(\sum_{i=1}^N h_i(\sigma_i^{(b)}) + \sum_{1 \leq i < j \leq N} J_{ij}(\sigma_i^{(b)}, \sigma_j^{(b)}) \right) \right] \\ &= \ln Z - \sum_{i=1}^N \sum_{k=1}^q f_i(k) h_i(k) - \sum_{1 \leq i < j \leq N} \sum_{k,l=1}^q f_{ij}(k, l) J_{ij}(k, l), \end{aligned} \quad (4)$$

where we have introduced the empirical one-point and two-point correlation functions

$$f_i(k) = \frac{1}{B} \sum_{b=1}^B \delta(\sigma_i^{(b)}, k), \quad (5)$$

$$f_{ij}(k, l) = \frac{1}{B} \sum_{b=1}^B \delta(\sigma_i^{(b)}, k) \delta(\sigma_j^{(b)}, l). \quad (6)$$

$\delta(a, b)$ is the Kronecker symbol taking value 1 if both arguments are equal, and 0 otherwise. Since (2) is of the form of a Gibbs–Boltzmann distribution of equilibrium statistical-mechanics, it maximizes the entropy under the constraints that the expectation values of all its “energy” terms are given. Learning the parameters $\{\mathbf{h}, \mathbf{J}\}$ (exactly) from minimizing l above is therefore equivalent to learning them by maximizing (exactly) the entropy given the observed $f_i(k)$ and $f_{ij}(k, l)$. This is a special case of a classical fact concerning sufficient statistics in exponential families of probability distributions [9–12]. As mentioned above, the problem with (4) is that for large systems Z is not efficiently and exactly computable, and exact maximum likelihood learning is hence not feasible. One solution to this dilemma is to keep the form of (4) but approximating Z ; the mean-field method of [13] and the message-passing method of [14], both discussed below, are in this class, as well as other and more sophisticated methods which have so far not been tested on the PSP problem [15–18].

The first attempt to predict spatial proximity by inferred interaction parameters was by Lapedes et al. [19] (unpublished) in 1999 using an iterative method where the normalizing constant Z was estimated by Monte Carlo. The calculations involved were very time-consuming and required supercomputing resources, and since at that time the number of known amino-acid sequences was much lower than today the wider implications were not noted. The same procedure was used in 2005 by Russ and collaborators as a way to conceive new protein sequences [20]. The next contribution was by Weigt et al. [14] in which a message-passing scheme was used, effectively computing Z in a Bethe–Peierls approximation. These calculations are still somewhat cumbersome and in practice only proteins of moderate size (N less than about 80) could be addressed, but very impressive results were nonetheless attained on the important example of two-step signal transduction pathways in bacteria. Slightly later, Burger and van Nimwegen [21] applied a Bayesian network model to the problem of predicting contact residues, followed by Balakrishnan and coworkers whose method GREMLIN was the first to utilize (l_1 -regularized) pseudolikelihood maximization for DCA [22].

The field then really took off from the 2011 paper [13], where Z was approximated by the lowest-order mean-field expansion, which means using the same formula as for learning a Gaussian model. This approach allowed for drastically shorter running times, since the central computation only amounts to inverting the correlation matrix between which amino acid is present at some position i and which amino acid is present at some other position j along the chain ($c_{ij}(k, l) = f_{ij}(k, l) - f_i(k)f_j(l)$), and eventually led to the first successful DCA-based algorithms for predicting whole 3D-structures of proteins [23,24]. Since the number of parameters in the model (2) is large (around $400N^2$), typically much greater than the number of examples B learnt from, some kind of regularization is necessary to avoid overfitting. In [13], the regularization is performed implicitly by asserting that correlations are computed combining real counts in a table of aligned sequences and added pseudocounts, which then renders the correlation matrices invertible. In the PSICOV routine of Jones et al. [25], the regularization is also performed by applying an l_1 penalty forcing the inverse correlation-matrix to be sparse. A recent further development modifies (2) to a Hopfield–Potts model where the independent interaction parameters are much fewer in number [26,27].

In [6] two of us introduced a different procedure which relies on l_2 -regularized pseudolikelihood maximization and a new and efficient score S_{ij}^{CFN} for ranking pairwise couplings within the protein structure. This method will here be referred

to as plmDCA (pseudolikelihood maximization Direct-Coupling Analysis). We will review the basis of this approach in Section 3 below. The GREMLIN method of [22] uses an l_1 -regularized pseudolikelihood objective, and does not utilize a score akin to S_{ij}^{CFN} for ranking couplings. In a recent contribution, however, Kamisetty et al. in [28] presented a new version of GREMLIN which also uses an l_2 -regularized pseudolikelihood objective and the interaction score S_{ij}^{CFN} , and which then goes further and expands the model to incorporate prior data (such as structural context information). In a parallel development, Skwark, Abdel-Rehim and Elofsson in [29] have combined plmDCA, PSICOV and protein alignments from multiple sources using random forests to a meta-predictor termed PconsC.

Several methods now integrate plmDCA into their computational frameworks, some mentioned above (see also EVfold³ [24] and EVcomplex [30]), so a reduction in execution time is highly desirable. The goal of this paper is to present a new version of plmDCA which achieves close to identical prediction accuracy as the original plmDCA, at a much lower computational cost. An evaluation of all the different DCA approaches is out of scope of the present paper, but to guide the reader and perchance newcomer to the field, the current consensus seems to be that the message-passing approach of [14] and the Bayesian network model of [21] are the weakest, and are both outperformed by the simpler mean-field method of [13]. The l_1 -regularized pseudolikelihood approach of [22] has, to our knowledge, not yet been matched against other methods. plmDCA [6] and PSICOV [25] on the other hand both outperform the mean-field method, and out of the two plmDCA has been reported to have the higher accuracy [29]. Both the meta-predictor of [29] and the integration of prior information in [28] improve upon the performance of plmDCA, the latter particularly in the important regime of small B , i.e. when few sequence homologues are available. The Hopfield–Potts inference of [26] has, as far as we are aware, only been performed using the mean-field method, and then works from less well to equally well as the method of [13] (but with many fewer parameters). The method of Lapedes et al. [19] has not been evaluated again using modern data and modern computer resources, and its relative performance as to prediction accuracy is hence unknown.

Numerous freshly conceived methods expand the concepts and applicability of DCA in various directions [31–41]. The field is growing rapidly, and other approaches are likely to appear in the near future.

3. Symmetric and asymmetric pseudolikelihood maximization

Pseudolikelihood maximization [42] starts from a different learning criterion than minimizing l , which in principle should give less accurate predictions than (3), but which is instead efficiently computable without further approximations (such as mean-field). The alternative learning criterion is to maximize the conditional probability of observing one variable given all the others, i.e. $P(\sigma_r = \sigma_r^{(b)} | \sigma_{\setminus r} = \sigma_{\setminus r}^{(b)})$, which for the model (2) comes out as

$$P(\sigma_r = \sigma_r^{(b)} | \sigma_{\setminus r} = \sigma_{\setminus r}^{(b)}) = \frac{\exp(h_r(\sigma_r^{(b)}) + \sum_{i=1, i \neq r}^N J_{ri}(\sigma_r^{(b)}, \sigma_i^{(b)}))}{\sum_{l=1}^q \exp(h_r(l) + \sum_{i=1, i \neq r}^N J_{ri}(l, \sigma_i^{(b)}))}, \quad (7)$$

where, to simplify the notation, we assume $J_{ri}(l, k)$ to mean $J_{ir}(k, l)$ when $i < r$. Given B observations we can hence define a negative pseudo-log-likelihood function

$$g_r(\mathbf{h}_r, \mathbf{J}_r) = -\frac{1}{B} \sum_{b=1}^B \ln[P(\sigma_r = \sigma_r^{(b)} | \sigma_{\setminus r} = \sigma_{\setminus r}^{(b)})], \quad (8)$$

for each amino-acid position $r = 1, \dots, N$. Here, \mathbf{J}_r denotes $\{J_{ir}\}_{i \neq r}$. Similarly to (4), this can be rewritten as

$$\begin{aligned} g_r(\mathbf{h}_r, \mathbf{J}_r) &= -\frac{1}{B} \sum_{b=1}^B \left\{ h_r(\sigma_r^{(b)}) + \sum_{i=1, i \neq r}^N J_{ri}(\sigma_r^{(b)}, \sigma_i^{(b)}) - \ln \left[\sum_{l=1}^q \exp \left(h_r(l) + \sum_{i=1, i \neq r}^N J_{ri}(l, \sigma_i^{(b)}) \right) \right] \right\} \\ &= z_r - \sum_{k=1}^q f_r(k) h_r(k) - \sum_{i=1, i \neq r}^N \sum_{k,l=1}^q f_{ri}(k, l) J_{ri}(k, l), \end{aligned} \quad (9)$$

where z_r is a position-specific normalization constant,

$$z_r = \frac{1}{B} \sum_{b=1}^B \ln \left[\sum_{l=1}^q \exp \left(h_r(l) + \sum_{i=1, i \neq r}^N J_{ri}(l, \sigma_i^{(b)}) \right) \right]. \quad (10)$$

³ <http://evfold.org/evfold-web/evfold.do>.

When data is abundant, maximizing conditional likelihood (exactly) is apt to give the same result as maximizing full likelihood (exactly). In the terminology of statistics, pseudolikelihood maximization is hence a consistent estimator, which is an important theoretical advantage of this approach to infer the interaction coefficients in (2).

Yet, given finite data maximizing conditional likelihood will deviate from maximizing full likelihood, and is in addition not in itself a fully specified method. Suppose we minimize g_i in (9) over the parameters $\{\mathbf{h}_i, \mathbf{J}_i\}$, and at the same time minimize for another node j the corresponding g_j in (9) over the parameters $\{\mathbf{h}_j, \mathbf{J}_j\}$. This will give us two inferred values of the matrix \mathbf{J}_{ij} , one from g_i and one from g_j . We shall denote these \mathbf{J}_{ij}^{*i} and \mathbf{J}_{ij}^{*j} respectively. These two will, in general, be different, while in the model (2) they have to be the same. Several ways can be imagined to resolve this inconvenience. The most straight-forward is to combine the N negative pseudo-log-likelihood functions into one overall score function, and then minimize this with the constraints that \mathbf{J}_{ij} is the same in both g_i and g_j (for all pairs of different i and j):

$$\{\mathbf{h}^*, \mathbf{J}^*\} = \arg \min_{\mathbf{h}, \mathbf{J}} [l_{pseudo}(\mathbf{h}, \mathbf{J})], \quad (11)$$

$$l_{pseudo}(\mathbf{h}, \mathbf{J}) = \sum_{r=1}^N g_r(\mathbf{h}_r, \mathbf{J}_r). \quad (12)$$

This approach was used in [6] and will here be referred to as *symmetric pseudolikelihood maximization* (symmetric as in $\mathbf{J}_{ij}^{*i} = \mathbf{J}_{ij}^{*j}$). While this has proved to be an accurate method to predict amino-acid contacts, it has the drawback of being somewhat slow, as it depends on a high-dimensional optimization.

In this paper we investigate the more radical approach – previously studied by two of us in [43] on synthetic data in the special case of binary variables ($q = 2$) – where all g_r are separately minimized, and the predictor of \mathbf{J}_{ij} is taken as the combination

$$\mathbf{J}_{ij}^* = \frac{1}{2}(\mathbf{J}_{ij}^{*i} + \mathbf{J}_{ij}^{*j}). \quad (13)$$

We will refer to this approach as *asymmetric pseudolikelihood maximization*. Due to the much lower dimensionality of each subproblem, minimizing all the g_r separately is a lighter task than minimizing l_{pseudo} . Furthermore, because the N minimizations (which in statistics language are multiclass logistic regression problems) are completely independent, the asymmetric variant easily lends itself to execution in parallel across many cores.

Although the engine of plmDCA is the maximization of pseudolikelihoods, various add-on techniques, tailored for the particular application to PSP, have been shown crucial for optimal performance; in fact, the increase in accuracy in [6] over [13] was shown to stem as much from a change in the score used to rank amino-acid interactions (discussed below) as from the choice of pseudolikelihood over mean-field. We therefore now turn to describing in particular the sequence reweighting, regularization and scoring used for the asymmetric plmDCA. Most current versions of DCA include one variant or another of each of these three, and new tactics for tackling these tasks are likely to appear. For instance, a Bayesian approach using priors may be assimilated to a regularizing penalty on the parameters, and it is now known from [28] that this improves prediction performance when B is small. It is also quite conceivable that more appropriate reweighting procedures can be found, perhaps including phylogenetic information, and similarly for the scoring.

3.1. Reweighting

Protein sequences in databases are very unevenly distributed, and there can be many rows in the data table which are closely similar. For instance, some types of species (e.g. human pathogens) are likely to have been sequenced many times, and many variants of the same protein from different variants of one species, or from closely related species, can be (and are) found in a database. A common heuristic approach to correct for such a bias is *sequence reweighting*, which was used in [6]. Essentially it means that each sequence contribution is multiplied with a weight that is inversely related to the number of similar sequences in a given MSA. Two sequences are considered similar if more than a fraction of x ($0 \leq x \leq 1$) of the positions in their chains are in the same state (one of the amino acids or a gap). To state this explicitly, each sequence $\sigma^{(b)}$ is assigned a weight $w_b = 1/m_b$, where m_b is the number of sequences in the MSA that are similar to $\sigma^{(b)}$:

$$m_b = |\{a \in \{1, \dots, B\} : \text{similarity}(\sigma^{(a)}, \sigma^{(b)}) \geq x\}|. \quad (14)$$

Using this technique, the frequencies and normalization in (9) are adjusted as

$$\begin{aligned} f_i(k) &= \frac{1}{B_{eff}} \sum_{b=1}^B w_b \delta(\sigma_i^{(b)}, k), & f_{ij}(k, l) &= \frac{1}{B_{eff}} \sum_{b=1}^B w_b \delta(\sigma_i^{(b)}, k) \delta(\sigma_j^{(b)}, l), \\ z_r &= \frac{1}{B_{eff}} \sum_{b=1}^B w_b \ln \left[\sum_{l=1}^q \exp \left(h_r(l) + \sum_{\substack{i=1 \\ i \neq r}}^N J_{ri}(l, \sigma_i^{(b)}) \right) \right], \end{aligned} \quad (15)$$

where $B_{\text{eff}} = \sum_{b=1}^B w_b$ is the effective number of sequences. Appropriate values for x were in [13] found to be in the range 0.7–0.9. In this work we use $x = 0.8$.

3.2. Gauge invariance and regularization

Although the convention $J_{ij}(k, l) = J_{ji}(l, k)$ removes most of the overparameterization in (1), there remains in (2) a more subtle redundancy: any constant c_i can be added to all elements in \mathbf{h}_i without changing any probabilities, since such a change will be compensated by a change of Z in (2), or by z_r in (9). Also, any function $u_i(k)$ can be added to $J_{ij}(k, l)$ and simultaneously subtracted from $h_i(k)$. Hence, a probability distribution of the form (2) is not uniquely represented; many distinct parameter sets correspond to the same distribution. Eq. (2) has $Nq + \frac{N(N-1)}{2}q^2$ parameters, but it is easy to show that the number of nonredundant parameters is $N(q-1) + \frac{N(N-1)}{2}(q-1)^2$. This overparameterization is in the statistical-physics literature referred to as a *gauge invariance*, and eliminating it as a *gauge choice* [13,14]. For example, the message-passing equations in [14] were derived under the *Ising gauge*,

$$\begin{cases} \sum_{s=1}^q J_{ij}(k, s) = 0, \\ \sum_{s=1}^q J_{ij}(s, l) = 0, \\ \sum_{s=1}^q h_i(s) = 0. \end{cases} \quad (16)$$

Including a regularization term typically removes this gauge freedom. In [6], for example, l_2 regularization was used, where instead of minimizing l_{pseudo} one minimizes $[l_{\text{pseudo}} + R_{l_2}]$ with

$$R_{l_2}(\mathbf{h}, \mathbf{J}) = \lambda_h \sum_{i=1}^N \|\mathbf{h}_i\|_2^2 + \lambda_J \sum_{1 \leq i < j \leq N} \|\mathbf{J}_{ij}\|_2^2, \quad \|\mathbf{h}_i\|_2^2 = \sum_{k=1}^q h_i(k)^2, \quad \|\mathbf{J}_{ij}\|_2^2 = \sum_{k,l=1}^q J_{ij}(k, l)^2. \quad (17)$$

λ_h and λ_J are regularization strengths to be specified by the user. Suitable values were in [6] found to be $\lambda_h = \lambda_J = 0.01$ (which, unless said otherwise, is also used in this work), and it was observed that this type of regularization implies the gauge

$$\begin{cases} \lambda_J \sum_{s=1}^q J_{ij}(k, s) = \lambda_h h_i(k), \\ \lambda_J \sum_{s=1}^q J_{ij}(s, l) = \lambda_h h_j(l), \\ \sum_{s=1}^q h_i(s) = 0. \end{cases} \quad (18)$$

For the asymmetric plmDCA, we shall demonstrate how regularization eliminates the need to fix a gauge. We will also use an l_2 penalty, added separately to each of the N objective functions; instead of minimizing g_r , we minimize

$$g_r^{(\text{reg})}(\mathbf{h}_r, \mathbf{J}_r) = g_r(\mathbf{h}_r, \mathbf{J}_r) + \lambda_h \|\mathbf{h}_r\|_2^2 + \lambda'_J \sum_{\substack{i=1 \\ i \neq r}}^N \|\mathbf{J}_{ri}\|_2^2. \quad (19)$$

We denote the coupling-regularization parameter λ'_J instead of λ_J to highlight the fact that it is not equivalent to λ_J in (17). Indeed, the correct relationship is $\lambda'_J \sim 0.5\lambda_J$, since in the asymmetric plmDCA each \mathbf{J}_{ij} is regularized twice, once in $g_i^{(\text{reg})}$ and once in $g_j^{(\text{reg})}$ (note that adding all $g_r^{(\text{reg})}$ gives $l_{\text{pseudo}} + 2R_{l_2}$ and not $l_{\text{pseudo}} + R_{l_2}$). Thus, following [6], proper input values⁴ to the asymmetric plmDCA are $\lambda_h = 0.01$ and $\lambda'_J = 0.005$. We now proceed to show that this regularization choice enforces a particular gauge. We first write $g_r^{(\text{reg})}$ out explicitly:

⁴ To promote backward compatibility of the asymmetric plmDCA with the symmetric, the distributable code (as well as the full algorithm description in Section 3.4) still uses λ_h and λ_J as input, and as a first step takes $\lambda'_J = 0.5\lambda_J$. This way, recommended input remains as $\lambda_h = \lambda_J = 0.01$.

$$\begin{aligned}
g_r^{(reg)}(\mathbf{h}_r, \mathbf{J}_r) &= -\frac{1}{B_{eff}} \sum_{b=1}^B w_b \log[P(\sigma_r = \sigma_r^{(b)} | \sigma_{\setminus r} = \sigma_{\setminus r}^{(b)})] + \lambda_h \|\mathbf{h}_r\|_2^2 + \lambda'_J \sum_{\substack{i=1 \\ i \neq r}}^N \|\mathbf{J}_{ri}\|_2^2 \\
&= -\frac{1}{B_{eff}} \sum_{b=1}^B w_b \left\{ h_r(\sigma_r^{(b)}) + \sum_{\substack{i=1 \\ i \neq r}}^N J_{ri}(\sigma_r^{(b)}, \sigma_i^{(b)}) - \log \left[\sum_{l=1}^q \exp \left(h_r(l) + \sum_{\substack{i=1 \\ i \neq r}}^N J_{ri}(l, \sigma_i^{(b)}) \right) \right] \right\} \\
&\quad + \lambda_h \|\mathbf{h}_r\|_2^2 + \lambda'_J \sum_{\substack{i=1 \\ i \neq r}}^N \|\mathbf{J}_{ri}\|_2^2.
\end{aligned} \tag{20}$$

From this, we can compute its partial derivatives:

$$\frac{\partial g_r^{(reg)}}{\partial h_r(s)} = -\frac{1}{B_{eff}} \sum_{b=1}^B w_b (I[\sigma_r^{(b)} = s] - P(\sigma_r = s | \sigma_{\setminus r} = \sigma_{\setminus r}^{(b)})) + 2\lambda_h h_r(s), \tag{21}$$

$$\frac{\partial g_r^{(reg)}}{\partial J_{ri}(s, k)} = -\frac{1}{B_{eff}} \sum_{b=1}^B w_b I[\sigma_i^{(b)} = k] (I[\sigma_r^{(b)} = s] - P(\sigma_r = s | \sigma_{\setminus r} = \sigma_{\setminus r}^{(b)})) + 2\lambda'_J J_{ri}(s, k). \tag{22}$$

$g_r^{(reg)}$ is smooth, so minimizing it means looking for point at which these derivatives are all zero. Setting (21) to zero and summing over s gives $2\lambda_h \sum_{s=1}^q h_r(s) = 0$ (since the sum across b vanishes). Similarly, setting (22) to zero and summing over s shows that $2\lambda'_J \sum_{s=1}^q J_{ri}(s, k) = 0$, while summing instead over k gives $\lambda'_J \sum_{k=1}^q J_{ri}(s, k) = \lambda_h h_r(s)$. Thus, the estimates coming from $g_r^{(reg)}$ are going to satisfy the gauge

$$\begin{cases} \lambda'_J \sum_{s=1}^q J_{ri}(k, s) = \lambda_h h_r(k), \\ \sum_{s=1}^q J_{ri}(s, l) = 0, \\ \sum_{s=1}^q h_r(s) = 0. \end{cases} \tag{23}$$

This seemingly creates an issue: our intent is to combine \mathbf{J}_{ij}^{*i} and \mathbf{J}_{ij}^{*j} via a simple average, (13), but since the gauge (23) depends on the node r , \mathbf{J}_{ij}^{*i} and \mathbf{J}_{ij}^{*j} are going to be delivered to us satisfying different gauges. The way we address the issue is to first shift both matrices to the same gauge. For a set $\{\mathbf{h}, \mathbf{J}\}$ in an arbitrary gauge, we obtain the corresponding set $(\hat{\mathbf{h}}, \hat{\mathbf{J}})$ in the Ising gauge (16) using the transformation

$$\begin{cases} \hat{J}_{ij}(k, l) = J_{ij}(k, l) - J_{ij}(:, l) - J_{ij}(k, :) + J_{ij}(:, :), \\ \hat{h}_i(k) = h_i(k) - h_i(:) + \sum_{j=1, j \neq i}^N \{J_{ij}(k, :) - J_{ij}(:, :)\}, \end{cases} \tag{24}$$

where “ $:$ ” denotes average over the indicated variable. We hence first use (24) separately on \mathbf{J}_{ij}^{*i} and \mathbf{J}_{ij}^{*j} , and then average element-wise. We remark that since both this gauge change and the average are linear operations, the order in which they are performed does not matter, and hence the issue is only apparent. Would one, however, want to attempt a more sophisticated combination of \mathbf{J}_{ij}^{*i} and \mathbf{J}_{ij}^{*j} , converting them to the same gauge first would be appropriate.

3.3. Scoring

For each pair (i, j) , the inference procedure spawns an entire matrix $\hat{\mathbf{J}}_{ij}^*$. To tally pairwise interactions by strength S_{ij} , some score is needed to reduce $\hat{\mathbf{J}}_{ij}^*$ to a scalar. In this work, as in [6], we use the Frobenius Norm (FN)

$$FN_{ij} = \|\hat{\mathbf{J}}_{ij}^*\|_2 = \sqrt{\sum_{\substack{k, l=1 \\ k, l \neq gap}}^q \hat{J}_{ij}(k, l)^2}, \tag{25}$$

corrected by the Average Product Correction (APC) introduced (for mutual information rather than FN) in [44], giving our score

$$S_{ij}^{CFN} = FN_{ij} - \frac{FN_{:j}FN_{i:}}{FN_{::}}. \quad (26)$$

In [6], two of us introduced this Corrected Frobenius Norm (CFN) – the omission of the gap state in (25) is however a slight modification from the original score in [6] – and found it to perform significantly better than both the FN and the Direct Information score used in [13]. Why the particular form in (26) works so well for the FN is currently unknown.

Note that the parameters to be plugged into (25) are in the Ising gauge; this should be seen as part of the definition of the CFN score. Changing gauges allows shifting parts of the Hamiltonian from the couplings over to the fields (parts of \mathbf{J}_{ij} can be put into \mathbf{h}_i and \mathbf{h}_j) or vice versa. Since we use a large $\sum_{k,l=1}^q J_{ij}(k,l)^2$ to indicate spatial proximity between positions i and j , we do not want these $J_{ij}(k,l)$ to contain anything which could have been explained by the fields instead; the “field part” would have little to do with the pair-interaction we are trying to score. In other words, we want to shift as much as possible of the Hamiltonian into the fields. The Ising gauge takes this reasoning into account, as among all gauge choices it makes $\sum_{k,l=1}^q J_{ij}(k,l)^2$ as small as possible.

3.4. A rundown of the asymmetric plmDCA

For clarity, we now recap each step of the asymmetric plmDCA procedure. An implementation in C/MATLAB is available.⁵ The input is an MSA $\{\sigma^{(b)}\}_{b=1}^B$, a reweighting threshold x ($0 \leq x \leq 1$) and regularization parameters λ_h and λ_j . Typical values are $x = 0.8$ and $\lambda_h = \lambda_j = 0.01$. The steps are:

1. Set $\lambda'_j = 0.5\lambda_j$.
2. Calculate weights $\{w_b\}_{b=1}^B$ according to

$$w_b = \frac{1}{|\{a, 1 \leq a \leq B : \text{similarity}(\sigma^{(a)}, \sigma^{(b)}) \geq x\}|}, \quad (27)$$

where $\text{similarity}(\sigma^{(a)}, \sigma^{(b)})$ is the fraction of positions where $\sigma^{(a)}$ and $\sigma^{(b)}$ have the same amino acid. Set $B_{\text{eff}} = \sum_{b=1}^B w_b$.

3. Minimize separately for all positions $r = 1, \dots, N$ the function

$$\begin{aligned} g_r^{(\text{reg})}(\mathbf{h}_r, \mathbf{J}_r) = & -\frac{1}{B_{\text{eff}}} \sum_{b=1}^B w_b \left\{ h_r(\sigma_r^{(b)}) + \sum_{\substack{i=1 \\ i \neq r}}^N J_{ri}(\sigma_r^{(b)}, \sigma_i^{(b)}) - \log \left[\sum_{l=1}^q \exp \left(h_r(l) + \sum_{\substack{i=1 \\ i \neq r}}^N J_{ri}(l, \sigma_i^{(b)}) \right) \right] \right\} \\ & + \lambda_h \|\mathbf{h}_r\|_2^2 + \lambda'_j \sum_{\substack{i=1 \\ i \neq r}}^N \|\mathbf{J}_{ri}\|_2^2, \end{aligned} \quad (28)$$

with gradient

$$\frac{\partial g_r^{(\text{reg})}}{\partial h_r(s)} = -\frac{1}{B_{\text{eff}}} \sum_{b=1}^B w_b (I[\sigma_r^{(b)} = s] - P(\sigma_r = s | \sigma_{\setminus r} = \sigma_{\setminus r}^{(b)})) + 2\lambda_h h_r(s), \quad (29)$$

$$\frac{\partial g_r^{(\text{reg})}}{\partial J_{ri}(s, k)} = -\frac{1}{B_{\text{eff}}} \sum_{b=1}^B w_b I[\sigma_i^{(b)} = k] (I[\sigma_r^{(b)} = s] - P(\sigma_r = s | \sigma_{\setminus r} = \sigma_{\setminus r}^{(b)})) + 2\lambda'_j J_{ri}(s, k). \quad (30)$$

This generates two estimates for each coupling matrix \mathbf{J}_{ij} : \mathbf{J}_{ij}^{*i} from $g_i^{(\text{reg})}$ and \mathbf{J}_{ij}^{*j} from $g_j^{(\text{reg})}$.

4. Shift the $N(N-1)$ obtained coupling matrices into the Ising gauge using the formula

$$\hat{J}_{ij}(k, l) = J_{ij}(k, l) - J_{ij}(:, l) - J_{ij}(k, :) + J_{ij}(:, :), \quad (31)$$

where “:” means average over the respective indices (amino acids). Note that we do not need to compute the corresponding Ising-gauge fields $\hat{\mathbf{h}}^*$, since only the couplings are used in what follows.

5. Get the final coupling matrix estimates, unique to each pair (i, j) , by taking the averages

$$\hat{\mathbf{J}}_{ij}^* = \frac{1}{2} (\hat{\mathbf{J}}_{ij}^{*i} + \hat{\mathbf{J}}_{ij}^{*j}). \quad (32)$$

⁵ <http://plmdca.csc.kth.se/>.

6. Calculate pairwise interaction scores S_{ij}^{CFN} through the two steps

$$FN_{ij} = \sqrt{\sum_{\substack{k,l=1 \\ k,l \neq gap}}^q \hat{J}_{ij}^*(k,l)^2}, \quad (33)$$

and

$$S_{ij}^{CFN} = FN_{ij} - \frac{FN_{:j}FN_{i:}}{FN_{::}}, \quad (34)$$

where “:” means average over the respective indices (positions along the chain).

4. Data

As discussed earlier, plmDCA requires an MSA, i.e. a table of aligned evolutionary related amino-acid sequences, as an input for the inference. In these tables, each row is a string containing one amino-acid chain coded by the one-letter abbreviations of amino acids. An example of an MSA is shown in Fig. S1. The MSAs used in this work are downloaded from PFAM, a free-to-use online database of amino-acid sequences divided into almost 15,000 so called domain families based on their evolutionary relationship. Families consist of a varying number of amino-acid sequences, ranging from a couple of dozens to tens of thousands. For each family PFAM offers an MSA, making the database an easy-to-use benchmark tool for providing input to test the performance of a DCA algorithm. For each PFAM-family, the website also offers pointers to experimentally measured structures in the PDB-database (see below) which in turn can be used to verify contact predictions.

The profile Hidden Markov-Model (HMM) used to generate the alignments in PFAM is designed in such a way that it only aligns the matching states of sequences, and when they are not alignable, it denotes the position in the corresponding sequence with a gap (“-”) [8]. Insert mutations, on the other hand, are not aligned, and if an amino acid is recognized as an insert, the column is simply listed into the alignment as a lowercase letter, but does not affect the rest of the alignment in any way. Thus, an insert in one sequence introduces an additional gap to all other sequences, which would induce bias into the data if inserts would be kept while performing DCA. For this reason, inserts are removed from the PFAM-alignments before DCA, as was done also in [6,13,14,23,24].

Experimentally determined protein structures are collected into another online database, Protein Data Bank (PDB), accessible via its member organization's (PDBE, PDBJ and RCSB) websites [45]. It is a freely available, weekly updated database currently containing about 100,000 three-dimensional protein structures. The traditional, and by far most utilized technique for protein structure determination is X-ray crystallography, but also NMR-spectroscopy has been widely applied [46]. For consistency, we use only X-ray structures to benchmark plmDCA.

Testing the accuracy of a DCA method is done by comparing contacts predicted from the MSA with contacts found from a corresponding X-ray structure from PDB. Distances between residues in the X-ray structure are measured from the α -carbons of the amino acids. A single PDB-structure is always just one realization from a given domain family, meaning it is usually not of the same length as the MSA obtained from PFAM. Position indexing between PFAM and PDB has to be matched via a third database, UNIPROT [47]. UNIPROT is a protein-sequence database whose entries are matched position by position to the entries in PDB, courtesy of the so called SIFTS-project [48]. This mapping allows linking PFAM-families to corresponding X-ray structures in PDB. To relate the indexing of PFAM-alignments and PDB-structures, we used the Backmapper software [49].

The PDB distance-files are essentially just lists of measured distances between each pair of amino acids, so how should one define a contact in the X-ray structures? In much of the current literature on DCA-like methods, amino acids are considered to be in contact if they are closer than 8 Å in space, a criterion used also in this work. It can, however, be argued that in potential applications one has significantly more use of contacts with separation less than about 5 Å than of those with 5 Å–8 Å. In our main analyses, we therefore also include results for the narrower cutoff 5 Å.

Moreover, we choose to restrict our evaluations to residue pairs further than five positions apart along the amino-acid backbone of the protein, i.e. $|j - i| \geq 6$ (see Fig. S19 for a view of how results depend on this choice). This essentially means disregarding the strong interactions among the neighboring residues and local secondary structure. In contrast to these, it is the pairs of amino acids that are close in space but *distant* in the sequence order that carry valuable information on the global spatial conformation of the chain.

Our set of protein structures for which contacts were predicted consists of 143 PDB-entries. The initial idea was to run the asymmetric plmDCA for all the 150 first PFAM families (PF00001–PF00150), but, due to for example the requirement of existence of at least one X-ray crystallography structure with resolution better than 3 Å, not all of the 150 first PFAM families were tested. The final set of family/structure-pairs also includes some PFAM-families outside of the 150 first entries, as some of the experimental structures include sequences from multiple domain families. The final list of PFAM-families and PDB-structures used is shown in Tables S2–S3, along with a list of rejected families (Table S4) and the reason for rejection.

Parts of the analysis will be presented separately for the fold types α , β , $\alpha + \beta$, α/β and membrane proteins, as categorized by the SCOP (Structural Classification of Proteins) database [50]. For proteins that did not have direct hits in SCOP,

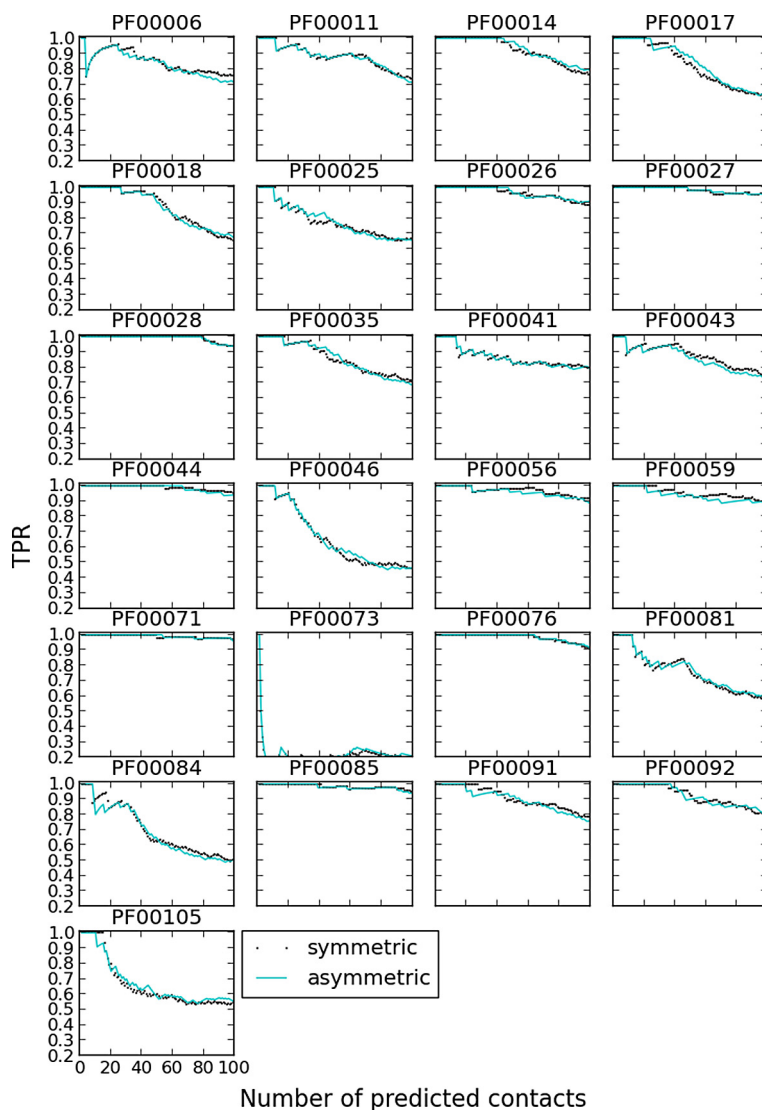


Fig. 1. X-axes show the number of predicted contacts (with $|j - i| \geq 6$) and y-axes show TPRs using the symmetric and asymmetric implementations of plmDCA for the families used in [6].

we queried the SUPFAM database [51] for matches based on the PFAM family, and we also submitted the sequences of such proteins to the Superfamily webserver [52] for SCOP classification. This way, 130 of our 143 proteins could be classified into one of the above categories (see Tables S2–S3).

5. Results

It is not immediately clear that the symmetric and asymmetric implementations of plmDCA should yield the same results. One might imagine that if Eqs. (9) have their minimas in very different parts of the parameter space for different positions, this could prevent our asymmetric plmDCA from reaching, or even coming close to, the minimum of Eq. (11). To assess the performance of the asymmetric plmDCA, we applied it to the collection of protein families used for the symmetric plmDCA in [6]. The predictions of the two methods are compared in Fig. 1 at different numbers of predicted contacts, using as an accuracy measure the True-Positive Rate (TPR), i.e. the fraction of predicted contacts that are actually contacts in the crystal structure. Unless stated otherwise, all tests in this section use regularization strengths $\lambda_h = \lambda_j = 0.01$ and reweighting threshold $\alpha = 0.8$. Fig. 1 clearly shows that the difference in accuracy between the two algorithms is negligible.

Fig. 2 shows the running durations for the domain families used in [6], using one CPU for the symmetric plmDCA and a varying number of CPUs for the asymmetric plmDCA. These times were attained on a computing cluster with the following hardware specifications:

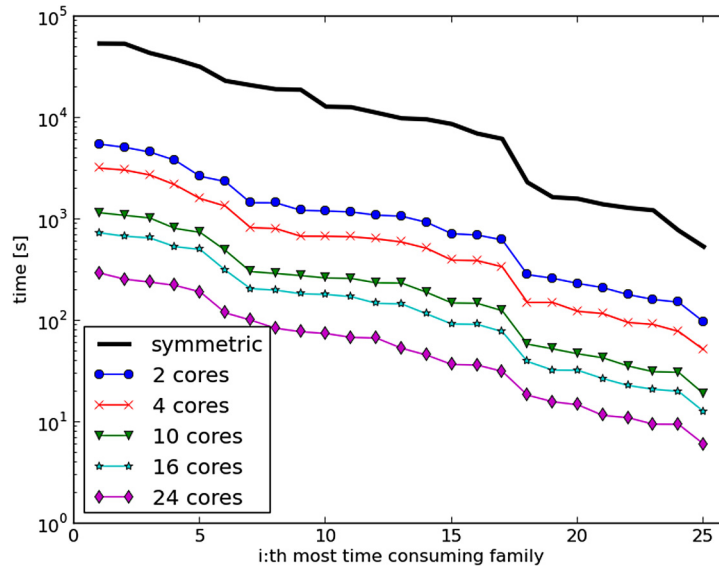


Fig. 2. Running times of the symmetric plmDCA using one CPU, and of the asymmetric plmDCA using various numbers of CPUs, for the families studied in [6].

– 107 nodes of type HP ProLiant BL465c G6, each equipped with 2x Six-Core AMD Opteron 2435 2.6 GHz processors. 80 of the nodes have 32 GB memory, while the remaining 27 have 64 GB memory.

– 118 nodes of type HP SL390s G7, each equipped with 2x Intel Xeon X5650 2.67 GHz (Westmere six-core each). Every SL390s G7 node has 48 GB of memory.

The minimizations, which are by far the most time-consuming part of plmDCA, were performed using a Limited-memory BFGS quasi-Newton descent scheme. The obvious overall take-away from Fig. 2 is that the asymmetric implementation can be performed much faster than the symmetric. Using the latter, some families need several hours, whereas they terminate within minutes using the new program, even employing as few as 10 CPUs. In fact, on just one CPU (on the same machine), the asymmetric variant still converges several times faster than the symmetric (data not shown). The drop in running time is, however, not linear with the number of cores, but is somewhat dependent on the architecture of the computing system used.

Due to the relatively long running times of the symmetric plmDCA algorithm, only a limited number of smaller families (both with respect to B and N) were used to assess its accuracy in [6]. With the faster asymmetric plmDCA, there are no such restrictions for sizes. Thus, the selection of families used in this study is more representative (see Tables S2–S3). In [6], it was demonstrated that the superior performance of plmDCA over the original mean-field based DCA (mfDCA) [13] stemmed partly from the use of the new score S_{ij}^{CFN} . Here, we therefore include results for mfDCA-CFN, an updated version of mfDCA that uses S_{ij}^{CFN} instead of Direct Information. For the regularization of mfDCA-CFN, we used the optimal pseudocount weight from [6], i.e. 0.7. Fig. 3 shows average TPRs over all 143 proteins for both methods. Clearly, plmDCA still keeps a modest but consistent advantage over mfDCA, for both cutoffs 5 Å and 8 Å. Figs. S2–S9 give individual TPRs plots for all our 143 proteins. It is evident from these figures that the differences in precision can be remarkable. For a large number of proteins almost all of the N top-scoring contacts actually exist in the crystal structure, while for a few the TPR is as low as 0.1–0.2 (e.g. PF00123, PF00143 and PF00236). Nevertheless, plmDCA predicts legitimate contacts with persistence across proteins, further reinforcing the rationale behind DCA.

There are 17 families in the data set with two or more crystal structures. Of these, 16 do not exhibit considerable differences between the prediction accuracy for different proteins. The exception is PF00089; while the contacts in “human neutrophil elastase” (2z7f) are predicted almost 100% correctly for the first N top scoring pairs, the TPR for the other structure from the same family, the “Glu 18 variant of turkey ovomucoid inhibitor third domain complexed with streptomycesgriseus proteinase B at pH 6.5” (1sge), is below 0.7.

Previous work has indicated that DCA performance is lower for α and membrane proteins than for other classes [23,37]. However, these studies used a rather small set of 15 proteins, with less than five from each fold type. To investigate, we present in Fig. 4 the results of Fig. 3 separately for each of the main fold types α , β , $\alpha + \beta$, α/β and membrane proteins. TPRs for α proteins are indeed lower on average, though only slightly less than those of β proteins. The highest performance is observed for α/β proteins. We remark that α proteins tend to have a smaller ratio of contacts to protein length than other classes (as was also pointed out in [37]). In contrast to the poor accuracy observed for the single membrane protein in [23,37], the two membrane proteins in our collection allow for reasonably good predictions.

For a clearer view of how plmDCA scatters its predicted contacts over proteins with different fold types, we show in Figs. S10–S16 individual contact maps for all the 130 classified proteins. For all fold types, contacts are in general well

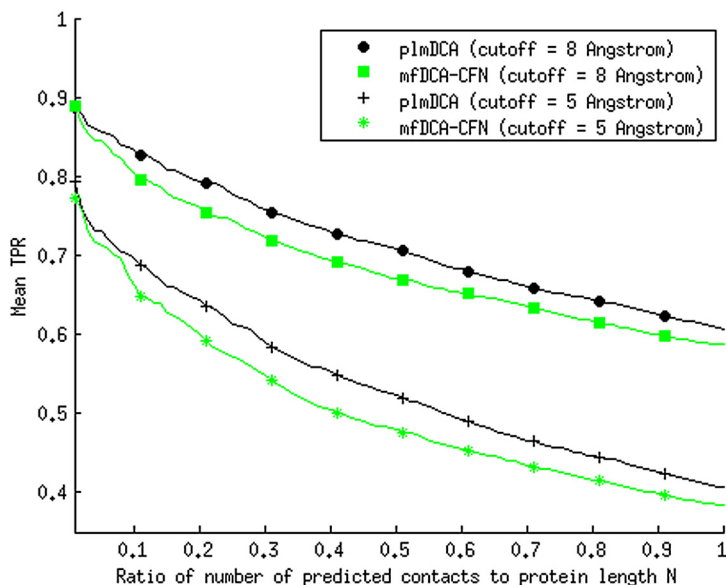


Fig. 3. Average TPRs over all 143 proteins in Tables S2–S3, as functions of p/N where p is the number of predicted contacts (with $|j-i| \geq 6$). Results are shown for two methods, mfDCA with S_{ij}^{CFN} score and asymmetric plmDCA, and for two different cutoffs for defining an amino-acid contact in the crystal structures.

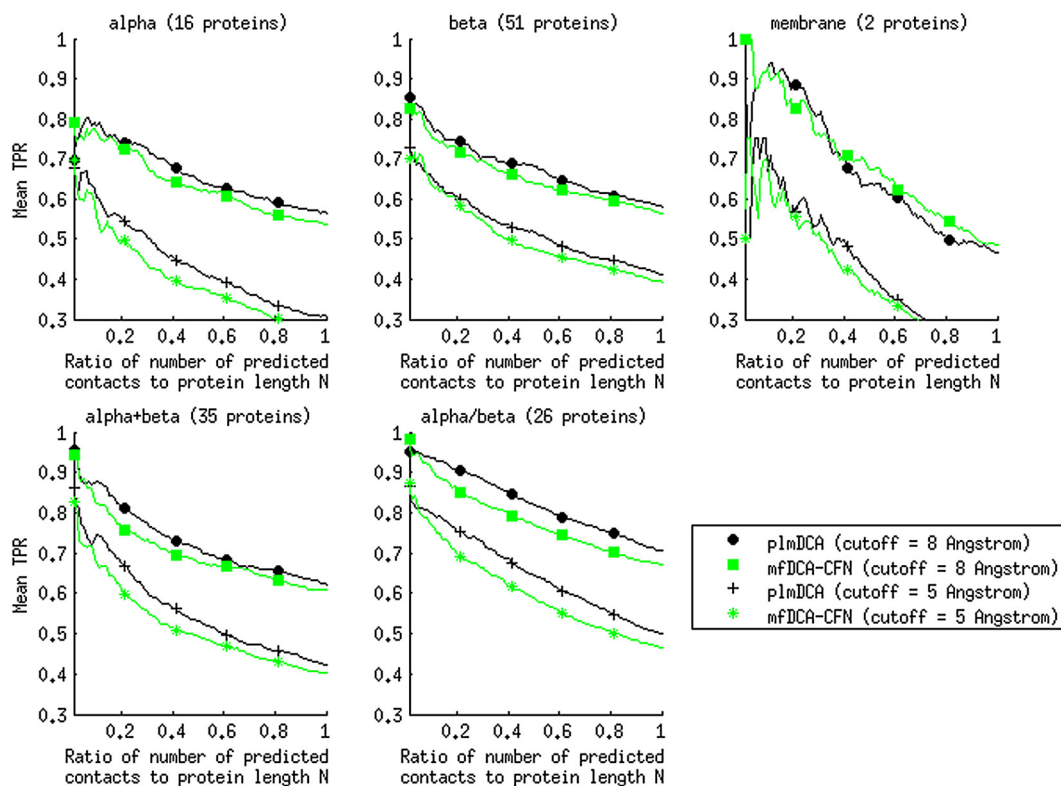


Fig. 4. Average TPRs for each of the fold classes α , β , $\alpha + \beta$, α/β and membrane proteins, as functions of p/N where p is the number of predicted contacts (with $|j-i| \geq 6$). Each plot is based on all the proteins with the designated fold class in Tables S2–S3. Results are shown for two methods, mfDCA with S_{ij}^{CFN} score and asymmetric plmDCA, and for two different cutoffs for defining an amino-acid contact in the crystal structures.

distributed over the map, i.e. there does not appear to be a strong tendency for plmDCA to detect coevolution in a particular kind of secondary structure element over another.

Fig. S17 shows accuracies of plmDCA as functions of N and B_{eff} . As expected, accuracy clearly correlates positively with B_{eff} , and no single fold type stands out as systematically deviating from the overall correlation. It is evident that α/β

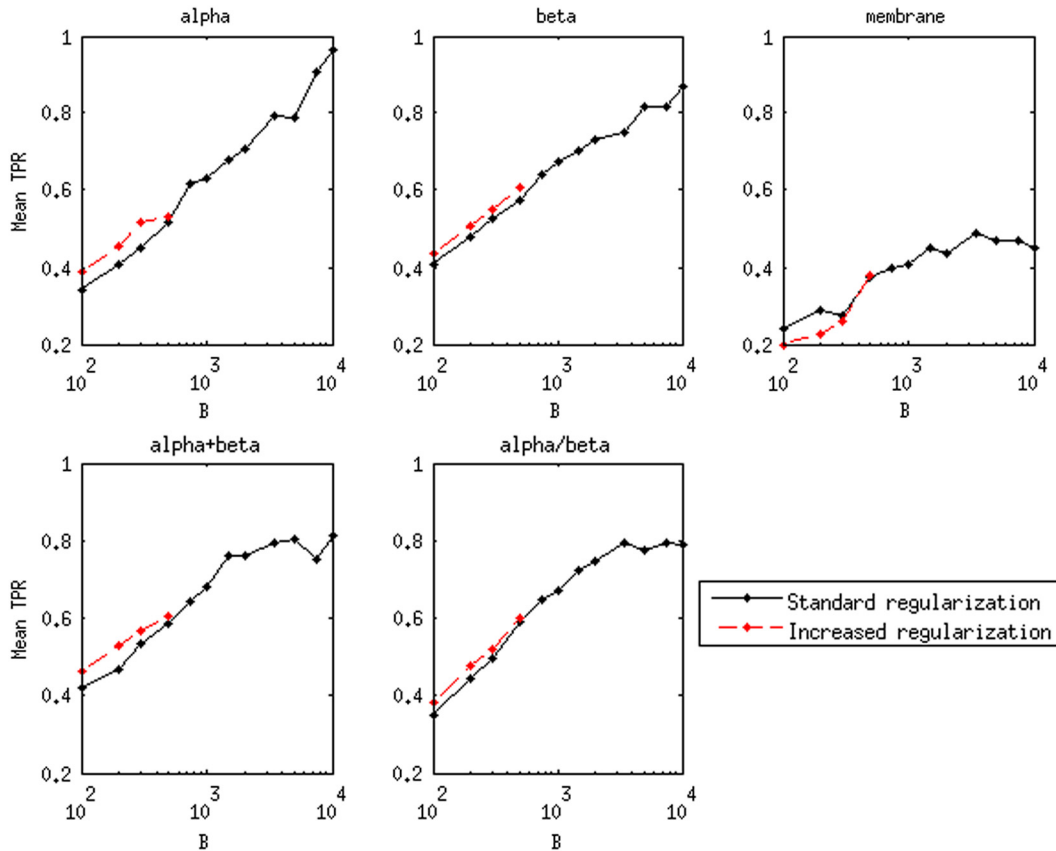


Fig. 5. Average TPRs as functions of the number of homologous sequences used for prediction. Results are based on the $N/2$ top scored pairs (with $|j - i| \geq 6$) from the asymmetric plmDCA, and are plotted separately for each of the fold classes α , β , $\alpha + \beta$, α/β and membrane proteins. Each dot represents the TPR averaged over all proteins in Tables S2–S3 (with the designated fold class) with sufficient sequences in the full alignment to sample at the indicated B . For $B < 500$, results are shown both for standard regularization $\lambda_h = \lambda_j = 0.01$ (solid lines) and for regularization tuned as $\lambda_h = \lambda_j = 0.01 \frac{B_{\text{eff}}}{500} + 0.1(1 - \frac{B_{\text{eff}}}{500})$ (dashed lines).

proteins on average have more sequence homologues available, possibly accounting for the better predictions noted in Fig. 4. There appears not to be an obvious dependency of TPR on protein length.

To examine in more detail how accuracy is affected by the number of available homologous sequences, we ran plmDCA using subalignments of varying sizes. These were generated by choosing sequences evenly spread out over the phylogenetic tree of the full alignment, thus increasing the chance that a subalignment of size B in fact has B fairly distinct sequences. The results are shown in Fig. 5. For all fold types, the accuracy increases steadily as B increases, but appears to level off when B reaches a few thousands. The sharp increase in average TPR at $B > 5000$ for α proteins is not a relevant exception, as it is due to the fact there are only two α proteins in our collection with large enough MSAs to create subalignments of sizes $B = 7500$ and $B = 10000$, while the average at the previous level $B = 5000$ is based on six proteins (see Fig. S18, which shows the same data as in Fig. 5 but for each protein individually).

Regularization is essentially a way to dampen negative effects of insufficient sampling, so one could suspect that predictions based on very few sequences may benefit from stronger regularization. For $B \leq 500$, we therefore ran additional experiments adjusting λ_h and λ_j as $\lambda_h = \lambda_j = 0.01 \frac{B_{\text{eff}}}{500} + R(1 - \frac{B_{\text{eff}}}{500})$, for various values of $R > 0.01$. We found the highest TPRs for R around 0.1. Fig. 5 shows that a slight boost in accuracy is indeed achievable by stricter penalization at low B .

6. Discussion

In this work, we have shown that an asymmetric implementation of a pseudolikelihood maximization approach to predict spatial amino-acid contacts from many homologous protein sequences, plmDCA, works equally well as a previously developed symmetric variant presented in [6], while drastically decreasing the running time of the algorithm. This allows plmDCA to be applied to more diverse sets of proteins than formerly possible, and to be competitive with e.g. mean-field based methods as to execution speed.

The difference between symmetric and asymmetric plmDCA lies in the output step when different predictions of an interaction matrix \mathbf{J}_{ij} , as seen from position i or as seen from position j , are harmonized. In the symmetric version one

tries to maximize a combined pseudolikelihood function over all the parameters at once, conceptually somewhat similar to conventional maximum likelihood. In the asymmetric version one instead separately makes two predictions \mathbf{J}_{ij}^{*i} and \mathbf{J}_{ij}^{*j} , and then combines them, here as $\mathbf{J}_{ij}^* = \frac{1}{2}(\mathbf{J}_{ij}^{*i} + \mathbf{J}_{ij}^{*j})$. An important theoretical point, which we discuss at some length, is how regularization fixes the gauges of \mathbf{J}_{ij}^{*i} and \mathbf{J}_{ij}^{*j} and that these gauges are generally different.

From the computational point of view, the symmetric plmDCA of [6] solves one optimization problem in $Nq(1 + (N - 1)q/2)$ parameters, while the asymmetric plmDCA solves N independent optimization problems each in $q(1 + (N - 1)q)$ parameters. We observe that significantly fewer descent steps are needed in these subproblems than in the high-dimensional single optimization, possibly accounting for why the asymmetric plmDCA is faster also when not utilizing parallel computing. Although one could imagine a parallel implementation also of the symmetric plmDCA – e.g. by carrying out the evaluation of $l_{\text{pseudo}} = \sum_{r=1}^N g_r$ and its gradient across several cores (although this would require significantly more cross-talk between the threads) – the asymmetric version is inherently parallel and can be trivially sped up using up to N CPUs virtually without overhead. For most protein families, the factor N is in the range 50–500. Such a steep increase in execution rate is well worth the insignificant precision change observed in Fig. 1, and we therefore propose the asymmetric plmDCA be preferred in the future.

Other ways to reduce execution time in plmDCA are conceivable, some of which were attempted during the course of this work. We experimented with various starting guesses for $\{\mathbf{h}, \mathbf{J}\}$ using mean-field estimates (regularized by pseudocounts as in [13]), but found these to reside too far from the pseudolikelihood maxima to offer substantial speed-up over cold-starting at the origin. We also tried constraining the entire minimization to the subspace of a gauge choice such as (16), but this merely increased the number of descent steps until termination.

Furthermore, several ways of further boosting the prediction accuracy were explored. We considered other combinations of \mathbf{J}_{ij}^{*i} and \mathbf{J}_{ij}^{*j} , such as $J_{ij}^*(k, l) = \min(J_{ij}^{*i}(k, l), J_{ij}^{*j}(k, l))$ and $J_{ij}^*(k, l) = \max(J_{ij}^{*i}(k, l), J_{ij}^{*j}(k, l))$, but found these to contain essentially the same information as the arithmetic average. We also probed several possible score alternatives, such as (i) an APC-corrected general l_p norm $\|\mathbf{J}_{ij}\|_p = (\sum_{k,l=1}^q J_{ij}(k, l)^p)^{1/p}$ for varying p , (ii) the score proposed in [31], i.e.

$$D_{ij} = \sum_{k,l=1}^q P_{ij}^D(k, l) \ln \frac{P_{ij}^D(k, l)}{f_i(k) f_j(l)}, \quad (35)$$

where

$$P_{ij}^D(k, l) \propto f_i(k) f_j(l) e^{J_{ij}(k, l)}, \quad (36)$$

or (iii) replacing the APC with an average *sum* correction,

$$S_{ij} = FN_{ij} - FN_{:,j} - FN_{i,:} + FN_{::}, \quad (37)$$

but on our dataset none of these replacements achieved accuracies as high as those of S_{ij}^{CFN} .

On our 143 proteins, plmDCA maintained the moderate accuracy advantage over mfDCA-CFN that was established in [6]. This is, however, probably not the most convincing reason for choosing plmDCA for future work. A more cogent argument is the following: other methods typically use a reduced version of the Potts model (e.g. a Gaussian approximation in the case of mfDCA and PSICOV), whereas plmDCA infers the full Potts model by utilizing an alternative learning criterion. This makes plmDCA well-suited for implementing extensions to the model, such as incorporation of prior information [28] or additional terms in the Hamiltonian [41]; the derivation of a pseudolikelihood objective is still straightforward. In contrast, it is not clear that, for example, mfDCA inference would remain fast and convenient after such modifications to the underlying model.

To conclude, plmDCA, the high accuracy of which no longer implies long waiting periods, should provide a natural choice for analysts interested in applying state-of-the-art PSP to their protein of interest, as well as for researchers looking to further extend the theory and practical applicability of DCA.

Acknowledgements

This work was supported by the COIN (Centre of Excellence in Computational Inference), Academy of Finland (grant number 251170), and through the Finland Distinguished Professorship program, project 129024/Aurell. We acknowledge the computational resources provided by Aalto Science-IT project, and thank Martin Weigt, Bryan Lunt, Cecilia Lövkvist and Nicolas Innocenti for help and discussions.

Appendix A. Supplementary material

Supplementary material related to this article can be found online at <http://dx.doi.org/10.1016/j.jcp.2014.07.024>.

References

- [1] J. Moulton, J.T. Pedersen, R. Judson, K. Fidelis, A large-scale experiment to assess protein structure prediction methods, *Proteins, Struct. Funct. Bioinform.* 23 (3) (1995) R2–R4, <http://dx.doi.org/10.1002/prot.340230303>.
- [2] N.D. Clarke, Covariation of residues in the homeodomain sequence family, *Protein Sci.* 4 (11) (1995) 2269–2278.
- [3] U. Göbel, C. Sander, R. Schneider, A. Valencia, Correlated mutations and residue contacts in proteins, *Proteins, Struct. Funct. Genet.* 18 (4) (1994) 309, <http://dx.doi.org/10.1002/prot.340180402>.
- [4] E. Neher, How frequent are correlated changes in families of protein sequences? *Proc. Natl. Acad. Sci. USA* 91 (1) (1994) 98–102, <http://dx.doi.org/10.1073/pnas.91.1.98>.
- [5] D.S. Marks, T.A. Hopf, C. Sander, Protein structure prediction from sequence variation, *Nat. Biotechnol.* 30 (11) (2012) 1072–1080, <http://dx.doi.org/10.1038/nbt.2419>.
- [6] M. Ekeberg, C. Lövkvist, Y. Lan, M. Weigt, E. Aurell, Improved contact prediction in proteins: using pseudolikelihoods to infer Potts models, *Phys. Rev. E* 87 (1) (2013) 012707, <http://dx.doi.org/10.1103/PhysRevE.87.012707>, <http://link.aps.org/doi/10.1103/PhysRevE.87.012707>.
- [7] Protein families-database, <http://pfam.sanger.ac.uk/>, accessed: 2013-10-24.
- [8] M. Punta, P.C. Coghill, R.Y. Eberhardt, J. Mistry, J.G. Tate, C. Boursnell, N. Pang, K. Forslund, G. Ceric, J. Clements, A. Heger, L. Holm, E.L.L. Sonnhammer, S.R. Eddy, A. Bateman, R.D. Finn, The Pfam protein families database, *Nucleic Acids Res.* 40 (D1) (2012) D290, <http://dx.doi.org/10.1093/nar/gkr1065>.
- [9] M.J. Wainwright, M.I. Jordan, Graphical models, exponential families, and variational inference, *Found. Trends Mach. Learn.* 1 (1–2) (2008) 1–305, <http://dx.doi.org/10.1561/22000000001>.
- [10] E. Pitman, J. Wishart, Sufficient statistics and intrinsic accuracy, *Math. Proc. Camb. Philos. Soc.* 32 (1936) 567–579, <http://dx.doi.org/10.1017/S0305004100019307>.
- [11] G. Darrois, Sur les lois de probabilités à estimation exhaustive, *C. R. Acad. Sci. Paris* 200 (1935) 1265–1266 (in French).
- [12] B. Koopman, On distribution admitting a sufficient statistic, *Trans. Am. Math. Soc.* 39 (3) (1936) 399–409, <http://dx.doi.org/10.2307/1989758>.
- [13] F. Morcos, A. Pagnani, B. Lunt, A. Bertolino, D.S. Marks, C. Sander, R. Zecchina, J.N. Onuchic, T. Hwa, M. Weigt, Direct-coupling analysis of residue coevolution captures native contacts across many protein families, *Proc. Natl. Acad. Sci. USA* 108 (49) (2011) E1293, <http://dx.doi.org/10.1073/pnas.1111471108>.
- [14] M. Weigt, R.A. White, H. Szurmant, J.A. Hoch, T. Hwa, Identification of direct residue contacts in protein–protein interaction by message passing, *Proc. Natl. Acad. Sci. USA* 106 (1) (2009) 67, <http://dx.doi.org/10.1073/pnas.0805923106>, arXiv:0901.1248v1.
- [15] V. Sessak, R. Monasson, Small-correlation expansions for the inverse Ising problem, *J. Phys. A, Math. Theor.* 42 (5) (2009) 055001, <http://dx.doi.org/10.1088/1751-8113/42/5/055001>.
- [16] S. Cocco, R. Monasson, Adaptive cluster expansion for inferring Boltzmann machines with noisy data, *Phys. Rev. Lett.* 106 (9) (2011) 090601, <http://dx.doi.org/10.1103/PhysRevLett.106.090601>.
- [17] S. Cocco, R. Monasson, Adaptive cluster expansion for the inverse Ising problem: convergence, algorithm and tests, *J. Stat. Phys.* 147 (2) (2012) 252–314, <http://dx.doi.org/10.1007/s10955-012-0463-4>.
- [18] F. Ricci-Tersenghi, The Bethe approximation for solving the inverse Ising problem: a comparison with other inference methods, *J. Stat. Mech.* 2012 (2012) P08015, <http://dx.doi.org/10.1088/1742-5468/2012/08/P08015>.
- [19] A.S. Lapedes, B.G. Giraudo, L. Liu, G.D. Stormo, Correlated mutations in models of protein sequences: phylogenetic and structural effects, in: *Statistics in Molecular Biology and Genetics*, in: *Lect. Notes Monogr. Ser.*, vol. 33, 1999, pp. 236–256.
- [20] W.P. Russ, D.M. Lowery, P. Mishra, M.B. Yaffe, R. Ranganathan, Natural-like function in artificial WW domains, *Nature* 437 (7058) (2005) 579–583, <http://dx.doi.org/10.1038/nature03990>.
- [21] L. Burger, E. van Nimwegen, Disentangling direct from indirect co-evolution of residues in protein alignments, *PLoS Comput. Biol.* 6 (1) (2010) E1000633, <http://dx.doi.org/10.1371/journal.pcbi.1000633>.
- [22] S. Balakrishnan, H. Kamisetty, J.G. Carbonell, S.-I. Lee, C.J. Langmead, Learning generative models for protein fold families, *Proteins, Struct. Funct. Bioinform.* 79 (4) (2011) 1061, <http://dx.doi.org/10.1002/prot.22934>.
- [23] D.S. Marks, L.J. Colwell, T.A. Sheridan, P. Robert, A. Pagnani, R. Zecchina, C. Sander, Protein 3D structure computed from evolutionary sequence variation, *PLoS ONE* 6 (12) (2011) e28766, <http://dx.doi.org/10.1371/journal.pone.0028766>.
- [24] T.A. Hopf, L.J. Colwell, R. Sheridan, B. Rost, C. Sander, D.S. Marks, Three-dimensional structures of membrane proteins from genomic sequencing, *Cell* 149 (7) (2012) 1607–1621, <http://dx.doi.org/10.1016/j.cell.2012.04.012>.
- [25] D.T. Jones, D.W.A. Buchan, D. Cozzetto, M. Pontil, PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments, *Bioinformatics* 28 (2) (2012) 184, <http://dx.doi.org/10.1093/bioinformatics/btr638>.
- [26] S. Cocco, R. Monasson, M. Weigt, From principal component to direct coupling analysis of coevolution in proteins: low-eigenvalue modes are needed for structure prediction, *PLoS Comput. Biol.* 9 (8) (2013), <http://dx.doi.org/10.1371/journal.pcbi.1003176>.
- [27] S. Cocco, R. Monasson, M. Weigt, Inference of Hopfield–Potts patterns from covariation in protein families: calculation and statistical error bars, *J. Phys. Conf. Ser.* 473 (1) (2013), <http://dx.doi.org/10.1088/1742-6596/473/1/012010>.
- [28] H. Kamisetty, S. Ovchinnikov, D. Baker, Assessing the utility of coevolution-based residue–residue contact predictions in a sequence- and structure-rich era, *Proc. Natl. Acad. Sci. USA* 110 (39) (2013) 15674–15679, <http://dx.doi.org/10.1073/pnas.1314045110>.
- [29] M.J. Skwark, A. Abdel-Rehim, A. Elofsson, PconsC: combination of direct information methods and alignments improves contact prediction, *Bioinformatics* 29 (14) (2013) 1815–1816, <http://dx.doi.org/10.1093/bioinformatics/btt259>.
- [30] T. Hopf, C. Schärfe, J. Rodrigues, A. Green, C. Sander, A. Bonvin, D. Marks, Sequence co-evolution gives 3D contacts and structures of protein complexes, *bioRxiv* <http://dx.doi.org/10.1101/004762>.
- [31] N.S. Burkoff, C. Várnai, D.L. Wild, Predicting protein β -sheet contacts using a maximum entropy-based correlated mutation measure, *Bioinformatics* 29 (5) (2013) 580–587, <http://dx.doi.org/10.1093/bioinformatics/btt005>.
- [32] C. Savojardo, P. Fariselli, P.L. Martelli, R. Casadio, BCov: a method for predicting β -sheet topology using sparse inverse covariance estimation and integer programming, *Bioinformatics* 29 (24) (2013) 3151–3157, <http://dx.doi.org/10.1093/bioinformatics/btt555>.
- [33] S. Lui, G. Tian, The network of stabilizing contacts in proteins studied by coevolutionary data, *J. Chem. Phys.* 139 (15) (2013) 155103, <http://dx.doi.org/10.1063/1.4826096>.
- [34] O. Rivoire, Elements of coevolution in biological sequences, *Phys. Rev. Lett.* 110 (17) (2013) 178102, <http://dx.doi.org/10.1103/PhysRevLett.110.178102>.
- [35] M. Andreatta, S. Laplagne, S.C. Li, S. Smale, Prediction of residue–residue contacts from protein families using similarity kernels and least squares regularization, *arXiv:1311.1301*.
- [36] Z. Wang, J. Xu, Predicting protein contact map using evolutionary and physical constraints by integer programming, *Bioinformatics* 29 (13) (2013) i266–i273, <http://dx.doi.org/10.1093/bioinformatics/btt211>.
- [37] S. Miyazawa, Prediction of contact residue pairs based on co-substitution between sites in protein structures, *PLoS ONE* 8 (1) (2013) e54252, <http://dx.doi.org/10.1371/journal.pone.0054252>.
- [38] J. Ma, S. Wang, J. Xu, Protein contact prediction by joint evolutionary coupling analysis across multiple families, *arXiv:1312.2988*.
- [39] S. Feizi, D. Marbach, M. Medard, M. Kellis, Network deconvolution as a general method to distinguish direct dependencies in networks, *Nat. Biotechnol.* 31 (8) (2013) 726–733, <http://dx.doi.org/10.1038/nbt.2635>.

- [40] C. Baldassi, M. Zamparo, C. Feinauer, A. Procaccini, R. Zecchina, M. Weigt, A. Pagnani, Fast and accurate multivariate Gaussian modeling of protein families: predicting residue contacts and protein-interaction partners, *PLoS ONE* 9 (3) (2014) e92721, <http://dx.doi.org/10.1371/journal.pone.0092721>.
- [41] C. Feinauer, M.J. Skwark, A. Pagnani, E. Aurell, Improving contact prediction along three dimensions, arXiv:1403.0379.
- [42] J. Besag, Statistical analysis of non-lattice data, *Statistician* 24 (3) (1975) 179–195, <http://dx.doi.org/10.2307/2987782>.
- [43] E. Aurell, M. Ekeberg, Inverse Ising inference using all the data, *Phys. Rev. Lett.* 108 (9) (2012) 090201, <http://dx.doi.org/10.1103/PhysRevLett.108.090201>.
- [44] S.D. Dunn, L.M. Wahl, G.B. Gloor, Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction, *Bioinformatics* 24 (3) (2008) 333–340, <http://dx.doi.org/10.1093/bioinformatics/btm604>.
- [45] Protein data bank <http://www.wwpdb.org>, <http://www.pdbe.org>, <http://www.rcsb.org/pdb>, <http://www.pdbj.org>, accessed: 2013-10-24.
- [46] G. Petsko, D. Ringe, Protein Structure and Function, Primers in Biology, New Science Press, 2004, http://books.google.fi/books?id=bCI5u_19N_oC.
- [47] Uniprot-database <http://www.uniprot.org/>, accessed: 2013-10-24.
- [48] S. Velankar, P. McNeil, V. Mittard-Runte, A. Suarez, D. Barrell, R. Apweiler, K. Henrick, E-MSD: an integrated data resource for bioinformatics, *Nucleic Acids Res.* 33 (suppl. 1) (2005) D262–D265, <http://dx.doi.org/10.1093/nar/1/gki058>.
- [49] B. Lunt, H. Szurmant, A. Procaccini, J.A. Hoch, T. Hwa, M. Weigt, Chapter two: inference of direct residue contacts in two-component signaling, *Methods Enzymol.* 471 (2010) 17–41, [http://dx.doi.org/10.1016/S0076-6879\(10\)71002-8](http://dx.doi.org/10.1016/S0076-6879(10)71002-8).
- [50] A. Murzin, S. Brenner, T. Hubbard, C. Chothia, SCOP: a structural classification of proteins database for the investigation of sequences and structures, *J. Mol. Biol.* 247 (4) (1995) 536–540, [http://dx.doi.org/10.1016/S0022-2836\(05\)80134-2](http://dx.doi.org/10.1016/S0022-2836(05)80134-2).
- [51] S. Pandit, R. Bhadra, V. Gowri, S. Balaji, B. Anand, N. Srinivasan, SUPFAM: a database of sequence superfamilies of protein domains, *BMC Bioinform.* 5 (28) (2004), <http://dx.doi.org/10.1186/1471-2105-5-28>.
- [52] J. Gough, K. Karplus, R. Hughey, C. Chothia, Assignment of homology to genome sequences using a library of hidden Markov models that represent all proteins of known structure, *J. Mol. Biol.* 313 (4) (2001) 903–919, <http://dx.doi.org/10.1006/jmbi.2001.5080>.