Extending PSICOV: Incorporating solvent accessibility information for structural contact prediction

Roxana Rădulescu and Timothy Verstraeten

Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium
{rradules, tiverstr}@vub.ac.be

Abstract

PSICOV is a statistical method that predicts residue-residue contacts from primary structure information. The algorithm successfully deals with indirect coupling effects among residue pairs. However, additional information, such as solvent accessibility, might improve the contact predictions. We attempt here to validate the PSICOV algorithm and propose an extension upon it, by incorporating solvent accessibility as a measure to allocate contacts in the residue-residue space. The obtained results show slight improvement over the original PSICOV method.

Introduction

Residue-residue contacts present important information for modelling the three-dimensional protein structures. Inter-residue interaction is known to have a critical role in the protein folding process, as well as in the stability of the structure (Gromiha and Selvaraj, 2004). Therefore contact prediction represents an active research area. Specifically *ab initio* procedures can benefit from progress in this field. These approaches are common in practice, as data is abundantly available compared to alternatives such as homology modelling, which require time-consuming homologue constructions (Lee et al., 2009).

However, it proves to be difficult to extract contacts from primary structure information, as there is no relation between the proximity of two amino acids in the sequence and their proximity in the protein fold. We define contacts to be residue pairs with C-β atoms (C-α for Glycine) that are within 8 Å of each other.

Many approaches have been proposed, most of them based on extracting co-evolutionary information to infer couplings between position pairs. This information can be derived from a multiple sequence alignment (MSA), which is an arrangement that groups together corresponding residues among a set of homologous sequences. The most common method is to compute the mutual information (MI) between two columns in the MSA (Dunn et al., 2008). Another approach makes use of a Bayesian network model to detect dependencies among residue pairs (Burger and Van Nimwegen, 2010). The efficiency of these methods comes from the preservation of structure and function over a protein family, despite the occurrence of mutations. In the case of two functionally related residues, if a mutation occurs for one of the residues, then it is more likely that the second one also evolves in such a way that would allow the preservation of the chemical and functional properties of the protein. Detecting these co-evolving pairs translates into finding residue contacts and will be referred to as 'correlated mutations'.

Although these approaches have gotten a lot of attention, the accuracy of the predictions remains rather low. A potential problem for all these statistical approaches is represented by the additional captured information regarding phylogenetic relationships and entropic noise that bias the final results. To some extent, a method for approximating and eliminating this additional background noise has been developed. However, indirect coupling, i.e., dependencies detected when chains of directly coupled residues form, remains a problem, as this introduces an apparent coupling between unrelated residues.

In addition to the number of contacts, solvent accessibility (SA) is also a key factor in characterizing protein structure (Pollastri et al., 2002). The SA of a residue describes the degree to which this residue is exposed to solvents, indicating its hydrophobicity level. It is common to assume that the SA of a residue is the inverse of its number of contacts, as buried residues tend to have more contacts (Pollastri et al., 2002). However, Fariselli and Casadio (2000) have shown that the distributions of these two measures are different, thus making it a theoretically unsound assumption.

This paper seeks to validate and extend the work of Jones et al. (2012). They introduce PSICOV (Protein Sparse Inverse COVariance), a method that attempts to separate indirect from direct coupling, thus providing more accurate predictions for residue-residue contact. In addition to the original method, we attempt to incorporate SA knowledge in order to investigate if the limited correlation between the measures can either bring a significant improvement in the results or even be detrimental to the entire procedure.

In the following sections we present a detailed description of the PSICOV algorithms, explaining the procedure of extracting the correlated sites within the MSA using a covariance matrix, followed by the estimation of the sparse inverse in order to remove the indirect coupling effects. We then present our approach on incorporating SA information in the procedure and, lastly, the computation of the final contact scores with the normalization method to account for the phylogenetic and entropic bias. Finally, we compare the results obtained for various settings of the system and analyse if our extension provides any significant improvements.

Method

PSICOV is a statistical method for predicting residue-residue contacts from primary structure information. More precisely, it extracts a covariance matrix from an MSA and inverts it in order to obtain correlations between residues. As these correlations also capture indirect coupling, a sparsity constraint on the inverse is necessary. Optionally, SA information can be provided to further model this constraint. A pipeline representation of the method is displayed in Figure 1.

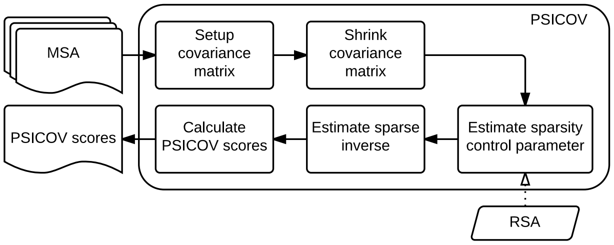

Figure 1: A pipeline diagram describing the PSICOV method. The input is an MSA for, and optionally RSA information about, a target sequence, while the output consists of the PSICOV scores.

Covariance Matrix

Given an MSA of n homologous sequences of length l, we set up an empirical $21l \times 21l$ covariance matrix, in order to indicate associations between any two amino acids at any position pairs over all sequences. Note that we introduce the gap as an additional element in the sequences, resulting in 21 amino acid types. We use the following general covariance formula for an entry:

$$S_{ij}^{ab} = \mathbb{E}[x_i^a \wedge x_j^b] - \mathbb{E}[x_i^a]\mathbb{E}[x_j^b] = f(a_i b_j) - f(a_i)f(b_j) \quad (1)$$

where x_i^a is a binary variable denoting whether amino acid a is present at position i and $f(a_i)$ represents the relative observed frequency of amino acid a at position i.

The frequencies can be extracted directly from the given alignment. Moreover, instead of using raw frequencies, one can also introduce weights for each sequence, to favour diversification in the evolutionary tree. Given a sequence identity threshold, the weight of a sequence is the inverse of the number of the sequences identical to that sequence. Additionally, we use add-one smoothing to account for unseen data, as we are handling a limited set of sequences.

Sparse Inverse Covariance Estimation

The 21×21 submatrices of our covariance matrix give an indication of coupling between position pairs. To get the actual correlation between those pairs, the inverse covariance matrix has to be calculated. However, the indirect coupling between two positions will still be captured in the exact inverse. To account for this, we apply sparse inverse covariance estimation. We can use the sparsity constraint induced by the estimation, since $\pm 3\%$ of the residue pairs are in direct contact.

The estimation procedure employed is the Graphical Lasso (glasso) method as presented by Banerjee et al. (2008). This statistical approach aims to minimize, w.r.t. the inverse covariance matrix, a linear combination between the negative log-likelihood and a regularization term which penalizes dense solutions:

$$tr(S\Theta) - \log\det\Theta + \sum_{ij} P_{ij}|\Theta_{ij}| \quad (2)$$

where S is the covariance matrix, Θ its inverse and P a matrix with positive entries that controls the sparsity level of the inverse.

A simple method of choosing P is to assign a fixed value ρ to each of its entries and keep adapting ρ until the desired sparsity in the inverse is achieved. This procedure, however, requires the application of glasso multiple times, thus can be time consuming. Regardless, an initial value for ρ can be picked such that the covariance matrix itself has the desired sparsity level, since the non-zero elements in the inverse are approximately the same as the non-sparse elements (i.e., entries $> \rho$) in the covariance matrix. This is based on the fact that the indices of non-sparse entries of both the covariance matrix and its inverse are the same for each of their respective connected components (Mazumder and Hastie, 2012).

By optimizing the negative log-likelihood equation, correlation information will be allocated at a limited set of entries, such that the estimated inverse best defines the correlations in the exact inverse. Therefore, the non-zero entries emerging from indirect coupling will be set to zero in the estimation, as indirect coupling can be explained by a chain of direct couplings.

We minimize this function by maximizing its dual representation (Banerjee et al., 2008). This is more convenient as it allows us to optimize w.r.t. to the dual variable, the covariance matrix itself, which means we can apply an iterative technique to make the covariance matrix converge. During the iterations, we use block coordinate descent (Friedman et al., 2008), in order to obtain row-wise optimized solutions.

This estimation technique represents the bulk of the computation, as solving this problem requires $O(21l)^3$ operations for the $21l \times 21l$ covariance matrix. The efficiency of the algorithm can be increased by grouping highly co-varying disjoint components in the covariance matrix, and applying glasso to each of these components (Witten et al., 2011). This reduces the complexity to $O((21l)^2 + \sum_{k=1}^{K} |C_k|^3)$, where C_k is one of the K components.

Incorporating Solvent Accessibility

As an attempt to further improve our contact predictions, we introduce relative solvent accessibility (RSA) information. For this purpose we have defined the contact number of a residue as the inverse of its RSA value.

A first improvement is done by applying sparse inverse covariance estimation until the total number of contacts is reached in the inverse for the currently considered protein. In order to approximate this value, the contact numbers inferred by the RSA are added up.

A second, more complex adjustment is introduced by deriving a contact probability distribution for all residue pairs from the contact numbers. More specifically, we define the probability of contact and no contact between residues i and j (which we assume to be independent) respectively to be:

$$\begin{aligned} p_{ij}^{c} &= \frac{C_i C_j}{(\sum_{k=1}^{l} C_k)^2} \\ p_{ij}^{\neg c} &= 1 - p_{ij}^{c} \end{aligned} \tag{3}$$

where C_i is the contact number of residue i. Given this distribution, we can create a grid of weights over the residue-residue space, where entry ij is defined as:

$$w_{ij} = 1 - \alpha + 2\alpha \frac{p_{ij}^{\neg c} - min_{uv}(p_{uv}^{\neg c})}{max_{uv}(p_{uv}^{\neg c}) - min_{uv}(p_{uv}^{\neg c})} \tag{4}$$

where α characterizes the impact of the RSA knowledge, such that $w_{ij} \in [1-\alpha, 1+\alpha]$. This grid is then incorporated in the sparsity control parameter P from Equation 2, such that $P_{ij}^{ab} = w_{ij}\rho$ for all amino pairs ab. The scale ρ can then be adjusted to achieve the desired sparsity level.

Shrinking Covariance Matrix

The inverse estimation should work well for ill-conditioned and singular matrices. However, it takes the estimation procedure a long time to converge in this case. To speed up convergence, we can shrink the matrix using an unbiased estimator.

$$S' = \lambda F + (1 - \lambda) S \tag{5}$$

where F is a scalar shrinkage target matrix with a scale equal to the mean of the variances of S, i.e., $F = diag(\frac{tr(S)}{21l})$. The estimator is unbiased in the sense that ratios between pairs of covariance values will be preserved.

This linear shrinking approach is done in such a way that the covariance eigenvalues are drawn towards their mean, as it has been shown in Ledoit and Wolf (2003). This signifies that the estimated matrix will become well-conditioned.

The termination condition for the shrinking iteration is met when the matrix is positive definite, and thus not singular anymore. For this purpose we can check whether the Cholesky decomposition exists, as this is only the case when the matrix is positive definite.

PSICOV Scores

Finally, we calculate the PSICOV scores to get the actual degree of direct coupling between each pair of positions. As we have a matrix indicating the correlations for each pair of amino acids at each pair of positions, we compute the 1-norm for each of the 20x20 submatrices in the inverse covariance matrix:

$$S_{ij}^{contact} = ||\Theta_{ij}^{ab}||_1 \tag{6}$$

where ab are all the possible amino acid types, as we exclude gaps' contribution.

To reduce phylogenetic and entropic bias, we also normalize these scores to get the actual PSICOV scores. We use the average product correction (APC) (Dunn et al., 2008), which assumes that each score is a sum of the functional relationship between the two positions and another term that incorporates background noise and evolutionary information. The latter one can be approximated and then subtracted from the original score as follows:

$$PC_{ij} = S_{ij}^{contact} - \frac{\bar{S}_{i-}^{contact} \bar{S}_{-j}^{contact}}{\bar{S}^{contact}} \tag{7}$$

where $\bar{S}_{i-}^{contact}$ (or $\bar{S}_{-i}^{contact}$) is the mean score over all positions paired with position i and $\bar{S}^{contact}$ is the mean over all scores.

Experiments

The method is implemented in MATLAB (MATLAB, 2013) and R (R Core Team, 2014), from which we use the *glasso* library (Friedman et al., 2014). The entire dataset used for this work, the resulting predictions for each of the settings described below and additional statistical tests, along with the source code will be made available.

Data

The experiments were run on the same 150 sequence alignments used in the original work of Jones et al. (2012). The target sequences were selected such that they were part of large Pfam families (>1000 sequences), having highly resolved X-ray crystallographic structures (resolution ≤ 1.9Å). All the sequences were initially retrieved from the PDB records (`www.rcsb.org`, Berman et al. (2000)), while the alignments were generated using the jackhmmer program from the HMMER 3.0 package (`http://hmmer.org`), for three iterations against the UNIREF100 data bank (Magrane et al., 2011). Only target sequences with lengths between 50 and 275 were considered and additional processing in the final alignments was made in order to remove duplicate rows as well as columns that introduced gaps in the target sequence.

The RSA data was retrieved using the NetSurfP server, version 1.1 (`http://www.cbs.dtu.dk/services/NetSurfP`, Petersen et al. (2009)), for each of the 150 target sequences.

Setting

We set up our experiments using the following parameters defined throughout the Method section:

- Sequence identity threshold – 62%
- Shrinkage parameter λ – 0.02
- Maximum glasso iterations – 5 (or until the sparsity target is met within a range of 0.01)

We compare the original PSICOV method against RSA enhanced versions. Firstly, we evaluate the potential improvement of incorporating RSA knowledge solely for estimating the desired target sparsity level of the inverse (PSICOV+RSA_nd - no distribution). Secondly, we examine the contribution of using an RSA derived distribution mask over the sparsity parameter P from Equation 2. We have chosen three different values for the impact parameter α used in Equation 4, namely 0.2, 0.5 and 0.8 (PSICOV+RSA_α). Finally, we perform statistical analysis on the obtained results, in order to evaluate the significance of the extensions.

Results

We evaluate the performance of our algorithms in terms of average precision, which is the number of true positives divided by the total number of predictions, averaged over all sequences. Throughout this section, the results are considered for various minimum number of residues in between a contact pair (referred to as sequence separation) and also for either a fixed fraction of the total number of predicted contacts (top-n), or a value proportional to the length of each considered sequence (top-L/k, with L as the sequence length).

At first sight, the extended methods do not seem to yield too much enhancement over the original PSICOV method. When we examine Figure 2, we can see the overall trend of PSICOV's precision in terms of the top-n ranked predictions, with sequence separation > 4. PSICOV+RSA_0.8, the algorithm with the most frequent highest average precision, follows closely the same trend. However, the improvement on PSICOV is only within $\pm 2\%$.

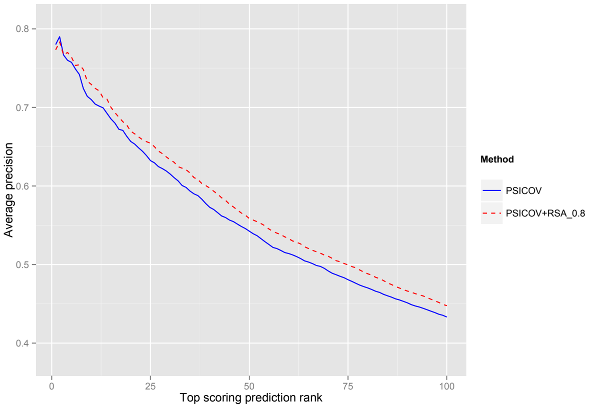

Figure 2: Average precision plotted against the top-n ranked predictions for PSICOV and the most consistently best performing algorithm PSICOV+RSA_0.8. The average precisions are between 0.4 and 0.8. Only contacts within a distance > 4 are considered.

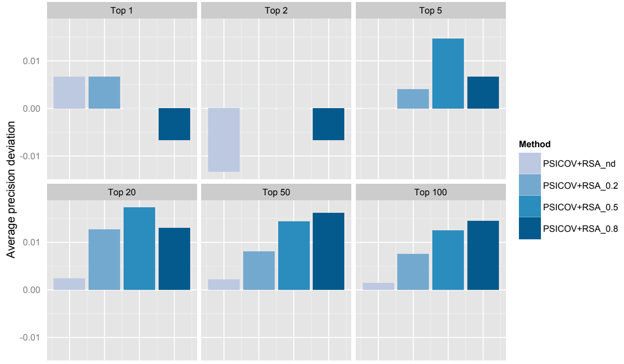

Figure 3: Average precision deviation over the top-n ranked predictions per method, w.r.t. the original PSICOV method. The precisions are calculated using a sequence separation of 4.

Figure 3 offers a wider view of the results, through a comparison across all methods, having as benchmark the original PSICOV precisions with a sequence separation of 4. We can see that most of the time, all methods outperform PSICOV, even though slightly. Only PSICOV+RSA_nd is worse in the case of the top-2 ranked predictions, and PSICOV+RSA_0.8 when considering the top-1 and top-2 predictions. Overall, PSICOV+RSA_nd does not provide a large improvement to PSICOV, compared to the methods that use the sparsity distribution. We can also notice that, when considering more predictions, PSICOV+RSA_0.8 ranks first, while PSICOV+RSA_0.5 is the best when only considering 5 or 20 predictions.

	[i-j]>4			
	L	L/2	L/5	L/10
PSICOV	0.3788	0.5058	0.6378	0.6962
PSICOV+RSA_nd	0.3806	0.5083	0.6388	0.7024
PSICOV+RSA_0.2	0.3856	0.5146	0.6473	0.7084
PSICOV+RSA_0.5	0.3909	0.5219	0.6557	0.7168
PSICOV+RSA_0.8	0.3946	0.5239	0.6547	0.7187

Table 1: Mean precision values for the top-L/k contacts for a sequence separation equal to 4, where the $C - \beta - C - \beta$ distance < 8Å

	[i-j]>8			
	L	L/2	L/5	L/10
PSICOV	0.3500	0.4800	0.6240	0.6972
PSICOV+RSA_nd	0.3523	0.4822	0.6236	0.7000
PSICOV+RSA_0.2	0.3568	0.4887	0.6328	0.7054
PSICOV+RSA_0.5	0.3616	0.4967	0.6434	0.7108
PSICOV+RSA_0.8	0.3654	0.4992	0.6429	0.7129

Table 2: Mean precision values for the top-L/k contacts for a sequence separation equal to 8, where the $C - \beta - C - \beta$ distance < 8Å

	[i-j]>11			
	L	L/2	L/5	L/10
PSICOV	0.3333	0.4606	0.6118	0.6897
PSICOV+RSA_nd	0.3341	0.4627	0.6138	0.6914
PSICOV+RSA_0.2	0.3386	0.4690	0.6232	0.6950
PSICOV+RSA_0.5	0.3435	0.4756	0.6310	0.7028
PSICOV+RSA_0.8	0.3474	0.4791	0.6317	0.7016

Table 3: Mean precision values for the top-L/k contacts for a sequence separation equal to 11, where the $C - \beta - C - \beta$ distance < 8Å

	[i-j]>23			
	L	L/2	L/5	L/10
PSICOV	0.2754	0.3949	0.5516	0.6438
PSICOV+RSA_nd	0.2776	0.3964	0.5572	0.6478
PSICOV+RSA_0.2	0.2816	0.4016	0.5647	0.6552
PSICOV+RSA_0.5	0.2860	0.4073	0.5743	0.6666
PSICOV+RSA_0.8	0.2887	0.4109	0.5757	0.6633

Table 4: Mean precision values for the top-L/k contacts for a sequence separation equal to 23, where the $C - \beta - C - \beta$ distance < 8Å

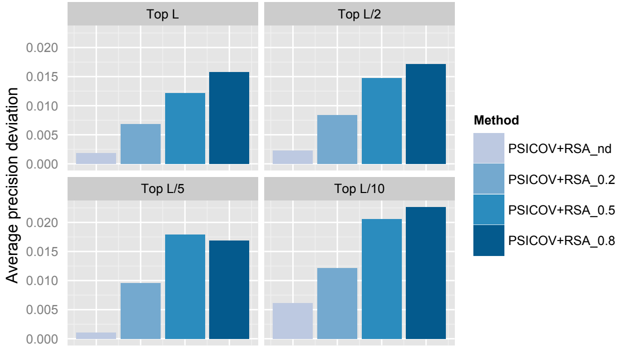

Figure 4: Average precision deviation over the top-L/k ranked predictions per method, w.r.t. the original PSICOV method. The precisions are calculated using a sequence separation of 4.

Examining Tables 1-4, we can see that for larger sequence separations, the precisions tend to decrease. The reason for this is that it is easier to predict residues in contact that are close to one another in sequence, than when they are far apart. This observation has already been made in the original PSICOV paper, but we can see that the other methods also follow this trend. Moreover, we encounter a nearly-consistent ranking among our algorithms, with the original PSICOV method placed last. For the others, it seems that the performance is directly proportional to the importance given to the SA knowledge, as it can be seen in Figure 4. Furthermore, there does not seem to be additional behaviour when considering the top-L/k contacts for different k.

In Figure 5, the average precisions over the top-L/2 predictions of the 150 target sequences for the PSICOV+RSA_0.8 method are plotted against the results of the PSICOV method, with a sequence separation of 4. In the majority of the cases PSICOV+RSA_0.8 performs equally good or better, even though there are a few sequences for which PSICOV outperforms it.

In order to further evaluate the results, we perform pairwise Wilcoxon tests with a confidence level of 95% between

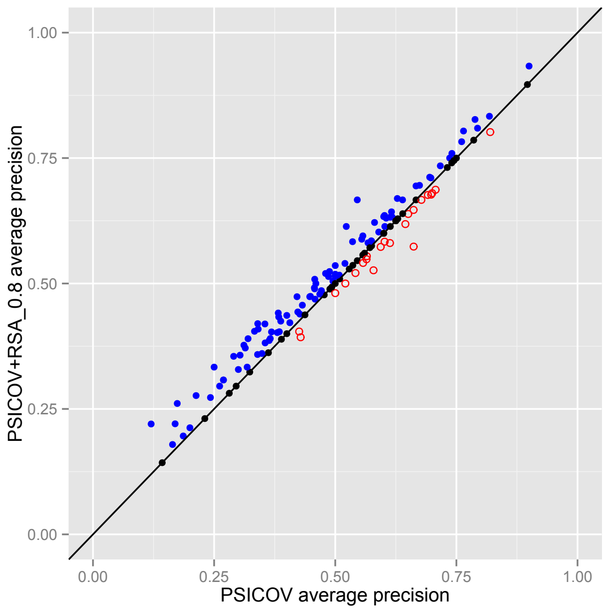

Figure 5: PSICOV average precision plotted against PSICOV+RSA_0.8 average precision for the 150 target sequences (line x = y taken as reference). The top-L/2 predictions are considered when computing the precisions, with a sequence separation of 4.

each of the methods. We use the average precisions of the top-L/2 ranked predictions and use a sequence separation of 4. The p-values and V-values of the tests are shown in Table 5. Even though the improvements are slim, most of them are highly significant ($p < 0.001$). The most interesting comparisons are the ones with the original PSICOV method. Given that the performances of the methods using the RSA sparsity distribution are significantly different from the performance of the PSICOV method, we can assume that including this distribution does improve our predictions. The only two insignificant results are those between the PSICOV and PSICOV+RSA_nd method, and between PSICOV+RSA_0.5 and PSICOV+RSA_0.8. Therefore, we cannot make any decisions regarding those results.

Discussion

We reimplemented the statistical contact prediction method PSICOV and validated the results from the original work of Jones et al. (2012). This method mainly focuses on eliminating the indirect coupling between residue pairs.

Additionally, we managed to slightly, but significantly, improve PSICOV, by incorporating information regarding the solvent accessibility of the composing residues for each sequence. The successful extension was made by computing a joint contact probability grid over the residue-residue space and introducing this in the sparsity control parameter for the glasso estimation algorithm. The attempt to infer the desired total number of contacts based on the RSA values of the residues does not seem to provide any significant contribution.

Method 1	**Method 2**	**p**	**V**
PSICOV	PSICOV+RSA_nd	0.11	1118
PSICOV	PSICOV+RSA_0.2	5.88e-5	1260
PSICOV	PSICOV+RSA_0.5	3.01e-10	994
PSICOV	PSICOV+RSA_0.8	1.52e-10	985
PSICOV+RSA_nd	PSICOV+RSA_0.2	8.06e-6	499
PSICOV+RSA_nd	PSICOV+RSA_0.5	5.20e-11	682
PSICOV+RSA_nd	PSICOV+RSA_0.8	9.71e-11	711
PSICOV+RSA_0.2	PSICOV+RSA_0.5	4.15e-6	729
PSICOV+RSA_0.2	PSICOV+RSA_0.8	2.50e-5	1236
PSICOV+RSA_0.5	PSICOV+RSA_0.8	0.11	1316

Table 5: p-and V-values of the pair-wise Wilcoxon tests between all pairs of methods. Their confidence levels are 95%. The precisions were computed for the top-L/2 contacts, with a sequence separation of 4

An interesting observation can be made by examining Figure 3. We notice that the larger the impact, the more contacts we need to evaluate in order to notice any improvement. This might indicate that the sparsity distribution contributes more overall, but at the cost of possibly polluting the correlations of the most ‘obviously’ directly coupled residue-residue pairs.

Moreover, looking at Figure 4, the following performance ranking seems to be true in most cases:

$$\begin{aligned}&\text{PSICOV+RSA\_0.8} > \text{PSICOV+RSA\_0.5}\\&> \text{PSICOV+RSA\_0.2} > \text{PSICOV+RSA\_nd}\\&> \text{PSICOV}\end{aligned}$$

where $>$ is defined as $x > y$ if and only if x outperforms y.

Further improvements could be made regarding the setup of the sparsity mask in the sparse inverse covariance estimation, as this part has been incorporated simplistically. A more complex relation could be found between the two employed measures, the number of contacts of a residue and its RSA value. Additionally, the weights in Equation 4 could be calculated using a different type of transformation than our linear approach.

All in all, we have shown that, despite the limited correlation between relative solvent accessibility and the number of contacts of a residue, and their theoretical unsound inverse relationship, solvent accessibility can provide a small improvement over the original PSICOV algorithm.

Acknowledgements

The current work was done within the context of the *Algorithms in Computational Biology and Bioinformatics* course at the Vrije Universiteit Brussels and it is based on the PSICOV method developed by Jones et al. (2012).

References

Banerjee, O., El Ghaoui, L., and d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *The Journal of Machine Learning Research*, 9:485–516.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T., Weissig, H., Shindyalov, I. N., and Bourne, P. E. (2000). The protein data bank. *Nucleic acids research*, 28(1):235–242.

Burger, L. and Van Nimwegen, E. (2010). Disentangling direct from indirect co-evolution of residues in protein alignments. *PLoS computational biology*, 6(1).

Dunn, S. D., Wahl, L. M., and Gloor, G. B. (2008). Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction. *Bioinformatics*, 24(3):333–340.

Fariselli, P. and Casadio, R. (2000). Prediction of the number of residue contacts in proteins. In *Intelligent Systems for Molecular Biology (ISMB-2000)*, volume 8, pages 146–151.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.

Friedman, J., Hastie, T., and Tibshirani, R. (2014). *glasso: Graphical lasso - estimation of Gaussian graphical models*. R package version 1.8.

Gromiha, M. M. and Selvaraj, S. (2004). Inter-residue interactions in protein folding and stability. *Progress in biophysics and molecular biology*, 86(2):235–277.

Jones, D. T., Buchan, D. W., Cozzetto, D., and Pontil, M. (2012). PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics*, 28(2):184–190.

Ledoit, O. and Wolf, M. (2003). Honey, I shrunk the sample covariance matrix. *UPF Economics and Business Working Paper*, 691.

Lee, J., Wu, S., and Zhang, Y. (2009). Ab initio protein structure prediction. In *From protein structure to function with bioinformatics*, pages 3–25. Springer.

Magrane, M., Consortium, U., et al. (2011). Uniprot knowledgebase: a hub of integrated protein data. *Database*, 2011:bar009.

MATLAB (2013). *version 8.1 (R2013a)*. The MathWorks Inc., Natick, Massachusetts.

Mazumder, R. and Hastie, T. (2012). Exact covariance thresholding into connected components for large-scale graphical lasso. *The Journal of Machine Learning Research*, 13(1):781–794.

Petersen, B., Petersen, T. N., Andersen, P., Nielsen, M., and Lundegaard, C. (2009). A generic method for assignment of reliability scores applied to solvent accessibility predictions. *BMC Structural Biology*, 9(1):51.

Pollastri, G., Baldi, P., Fariselli, P., and Casadio, R. (2002). Prediction of coordination number and relative solvent accessibility in proteins. *Proteins: Structure, Function, and Bioinformatics*, 47(2):142–153.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Witten, D. M., Friedman, J. H., and Simon, N. (2011). New insights and faster computations for the graphical lasso. *Journal of Computational and Graphical Statistics*, 20(4):892–900.