A multi-method approach for proteomic network inference in 11 human cancers

Yasin Şenbabaoğlu, Selçuk Onur Sümer, Giovanni Ciriello, Nikolaus Schultz, Chris Sander

Computational Biology Program, Memorial Sloan Kettering Cancer Center, New York, NY, USA.

SUPPLEMENTARY MATERIALS

- **Text S1**: Major caveats in performance evaluation of computational proteomic network predictions.
  - The pros and cons of using RPPA data in the inference of cellular networks
  - The community-wide underperformance of network inference methods
- **Text S2**: Different tumor types are not equally amenable to network discovery with RPPA data. *(Figure S1, Figure S2)*
- **Text S3**: The relative merit of TOP6 and ALL13 consensus calls. *(Figure S3)*
- **Text S4**: Choosing a cutoff for consensus edge-ranks in tumor type clustering. *(Figure S4)*
- **Text S5**: Steps involved in computational network inference.
  1. Running the methods *(Table S1)*
  2. Plotting precision-recall curves for individual parameter configurations
  3. Computing AUPR and determining the optimal parameter values *(Table S2)*
  4. Interpretation of edge weights and normalization for CLR
  5. The principal component analysis (PCA) of the 13 network inference methods *(Table S3)*
  6. Computing the consensus edge-ranks and consensus edge-weights across methods
  7. PCA and hierarchical clustering on the 11 tumor types *(Table S4)*
- **Text S6**: Mathematical descriptions of the 13 tested methods
  - Correlation-based methods
    - Pearson correlation
    - Spearman correlation
  - Methods based on partial correlation
    - Formulation 1: Inversion of the covariance matrix
      - Simple partial correlation
      - GeneNet
      - Graphical lasso
    - Formulation 2: Least squares regression
      - Partial least squares regression
      - Ridge regression
      - Lasso regression
      - Elastic net regression
  - Methods based on mutual information
    - ARACNE with additive penalty
    - ARACNE with multiplicative penalty
    - Context-likelihood of relatedness (CLR)
    - MRNET
Major caveats in performance evaluation of computational proteomic network predictions

- The pros and cons of using RPPA data in the inference of cellular networks

The RPPA technology has previously been validated to allow a practical screening of potential biomarkers in certain cancers \(^1\), \(^2\). However, using RPPA datasets for cellular network inference has both **pros** and **cons** when compared with using mRNA-measuring platforms such as RNAseq or microarrays. The pros are primarily related to cellular physiology and include facts such as following: (1) Proteins are closer to the organism’s **phenotype** than mRNAs; (2) mRNAs are subject to **degradation**, so correlation between mRNA and protein abundance is low; (3) mRNAs cannot be used to measure activity levels of phosphorylated proteins which are key components of signal transduction, an important mechanism for the flow of influence/information in the cell. Collectively, these factors suggest that RPPA levels are a better proxy for both protein abundance and pathway activity levels compared with mRNA data.

The **cons** of using RPPA experiments for network inference primarily involve technology- and platform-related shortcomings. For instance, (1) the RPPA platform may have higher technical noise than transcriptomic profiling platforms such as RNAseq and mRNA microarrays. (2) Currently, there are antibodies for only a limited number of proteins (less than 300) while it is possible to perform genome-wide mRNA expression profiling using RNAseq or microarrays. (3) Some of the widely-used antibodies are known to lack specificity for their conjugate proteins (for instance, antibodies for phospho-EGFR and phospho-HER2 are likely to bind to both proteins), and some others may have yet under-appreciated specificity problems. (4) The quality of antibodies has to be checked right before an RPPA experiment because the specificity of antibodies can decay over time. Also, (5) quantifying the abundance of the non-phosphorylated state of a protein currently remains a challenge in RPPA experiments due to the lack of an antibody that specifically targets the non-phosphorylated state.

- The community-wide underperformance of network inference methods

High-throughput genomic and proteomic platforms generate high-dimensional datasets with a strong correlation structure. An accurate classification of these correlations into the direct and transitive relationships is an unsolved challenge.

The precision of computational network inference methods remains less than satisfactory not only for PPI networks but also for other applications such as gene regulatory networks \(^3\). One important reason could be that, the statistical dependence paradigms involving measures such as partial correlation and mutual information are not sufficient to model the biochemical reactions in the cell. Moreover, the fact that many cellular events involve interactions within multi-protein complexes introduces a level of complexity higher than can be modeled by pairwise interactions alone. However, the ansatz is that predominant pathway activities can potentially be captured by the inference of direct pairwise interactions.

Further, to what extent one can empirically characterize direct and transitive relationships depends substantially on the inclusion of all potentially confounding variables in the model. A regulatory mechanism such as proteolysis, epigenetic silencing, microRNA regulation, etc. that concurrently acts on two proteins, but is not accounted for in the dataset could easily lead to **spurious direct-association calls** between the proteins. On the contrary, it is possible that true direct associations could be missed due to tight regulation among a group of proteins: for instance, high correlations between a transcription factor and multiple proteins it upregulates could hinder the inference of one or
more of the direct interactions between the transcription factor and each one of the regulated proteins.

Such lack of signal resolution in tracking information/influence flow in biological networks can be moderated with information from time-series and perturbation experiments, i.e. temporal and causal domain respectively. However, it is important to note that strong correlations found in the temporal domain alone, or found in the causal domain alone are not necessarily reproducible in the other domain. For instance, it was shown that, in experiments with receptor tyrosine kinases, some phosphosite pairs were strongly correlated in the temporal domain (i.e., having similarly shaped time courses) but not the causal domain (i.e. not responding similarly to shRNA perturbations); whereas other phosphosite pairs were not well correlated in the temporal domain (i.e. their time course shapes look different), but were correlated in the causal domain (i.e. they respond similarly to perturbations). These two phenomena are exemplified by the [phospho-PKC-beta & phospho-MARCKS], and the [phospho-PKC-beta & phospho-ERK] pairs respectively. These results suggest that even time-course data, let alone static profiles, may not be sufficient to infer causal relationships among proteins in the cell. However, static computational PPI-network predictions are still attractive in that they can reduce the hypothesis space of interactions and prioritize them so that interactions with higher statistical significance can be functionally studied in the laboratory before others.

**Text S2**

**Different tumor types are not equally amenable to network discovery with RPPA data.**

We see in Figure 3a that none of the methods reach an AUPR level of 0.02 for five of the eleven tumor types (KIRC, OV, COAD, READ, and BLCA). For three other tumor types, namely LUAD, HNSC, and LUSC; mutual-information- and correlation-family methods ARACNE, CLR, SPEARMANCOR, and PEARSONCOR achieve some of the highest AUPRs in this study. On the other hand, regularized methods such as RIDGENET, GLASSO, and ELASTICNET have some of the worst performances for these tumor types. This is a surprising result because SPEARMANCOR and PEARSONCOR do not remove any transitive edges from the system, and RIDGENET is shown to be the best-overall performing method in Figure 3b. There is an inherent difference in LUAD, LUSC, and HNSC datasets that enables non-regularized methods to perform better. We asked whether a difference between regularized and non-regularized methods might be leading to differences in the topological characteristics of predicted networks, that in turn could partially explain the higher performance of non-regularized methods in the LUAD, LUSC, and HNSC datasets. One such difference is the number of conditioned nodes when the independence of two nodes is examined. Regularization-based methods condition on the other N-2 nodes when inspecting the existence of an edge in a system with N nodes. In contrast, non-regularized methods either do not condition on any nodes at all (i.e. correlation) or condition on only a third node (i.e. ARACNE).

We investigated the topological characteristics of the predicted networks and the gold-standard benchmark by computing network statistics such as average-node-degree, network density, optimal number of modules, and modularity index. The *average-node-degree* for an undirected network is the mean value of all node degrees where the node degree is simply the number of edges incident on a node. The *density* of a network is the ratio of the number of edges in the network to the maximum possible number of edges. The *optimal number of modules* and the *modularity indices* are computed using the edge-betweenness statistic in the Girvan-Newman algorithm.

We observe in Figure S1b and S1d that the average-node-degree and network-density in predicted networks have a high negative correlation with AUPR (Spearman r = −0.626 and −0.453 respectively). Indeed, the methods with the highest values for these statistics in Figure S1a and S1c, namely SIMPLEPARCOR and PLSNET, are among the poor-overall performing methods according
In contrast, the best-overall performing methods such as RIDGENET and ARACNE are not the ones with the lowest density or the lowest average-node-degree network predictions. The high-performers tend to yield networks with density and average-node-degree values that have not substantially deviated from the same statistics derived from the gold-standard network (Figure S1a-S1d).

We note that regularized and non-regularized methods do not have a marked difference in terms of network density and average-node-degree. However, we observe in Figure S2c and S2d that (1) low optimal module number and (2) high modularity score are both discriminating and also pan-cancer characteristics of non-regularized methods such as SPEARMANCOR, PEARSONCOR, ARACNE, and CLR. It can be speculated that these methods yield more modular networks because the number of conditioned nodes is far fewer (or zero), thus they preserve to a greater extent the original protein-protein correlation structure in the data. Preserving the original correlation structure is not always a desirable property since the goal in network construction is to eliminate transitive edges, i.e. spurious correlations. However, the correlation structure in the LUSC, LUAD, and HNSC datasets could be so strong that (1) it leads to better performance for non-regularized methods, and (2) it causes regularized methods to erroneously eliminate true edges due to conditioning on N-2 nodes when inspecting the existence of an edge. Considering the shared tissue or cell type of origin among LUSC, LUAD, and HNSC, the inherent difference in these datasets compared with the others in PANCAN11 potentially stem from a biological difference rather than a systematic bias.
**Figure S1**: Statistics of predicted networks for each method and each tumor type. Tumor types are on the x-axis and different methods are denoted with colors. (a) average node degree, (c) network density, (e) optimal number of modules, and (g) highest modularity score. (b,d,f,h) Scatter plots of these network statistics with AUPR.
Figure S2: Statistics of predicted networks for each method and each tumor type. In contrast with Figure S1, methods are on the x-axis and different tumor types are denoted with colors. (a) average node degree, (c) network density, (e) optimal number of modules, and (g) highest modularity score.

Text S3

The relative merit of TOP6 and ALL13 consensus calls

Each edge has a separate consensus rank from the TOP6 and the ALL13 groups of methods, computed as the average of the ranks from the relevant methods. Since this consensus rank is different in each tumor type, a pan-cancer consensus can be obtained by taking the average of consensus ranks from all 11 tumor types. It is possible to perform a pan-cancer comparison of TOP6 and ALL13 using these single pan-cancer edge ranks. We took the same number of top edges from the pan-cancer-ranked edge lists of TOP6 and ALL13, and computed (a) the overlap and difference percentages between the two lists, and (b) the precision score of each list for predicting the interactions in the gold-standard network. We then varied the number of edges from 1 to 3000 in order to assess the trends in these variables.
**Figure S3a** shows that there is a high level of intersection between these two sets: higher than 80% for the most restrictive edge-rank thresholds, dropping to 60% around 250 edges, and eventually plateauing around 70%. The TOP6 and ALL13 lists may have modest differences, but we see in **Figure S3b** that TOP6 edges achieve a slightly superior AUPR value compared with ALL13 edges when compared against the benchmark. The superiority in performance is more pronounced for the first 1000 edges after which the performances of the two groups become virtually equivalent.

**Figure S3**: The comparison of TOP6 and ALL13 edge lists. First, a pan-cancer rank for each edge is computed separately from the TOP6 and the ALL13 methods. Edge lists ordered according to these pan-cancer ranks are then subjected to comparison. (a) Overlap and difference percentage between the significant ALL13 and TOP6 edges as the significance threshold is varied from 1 to 3000. (b) Precision of the significant ALL13 and TOP6 edges in the gold-standard network as the significance threshold is varied from 1 to 3000.

**Text S4**

**Choosing a cutoff for consensus edge-ranks in tumor type clustering**

We would like to perform the unsupervised clustering of tumor types using only the high-confidence edges that well represent both the similarities and the differences between tumor types. The union set of edges from limited-recall (**Table S3**) has 13193 edges, resulting in a network density of 76% (in the total of 17391 possible edges). Considering that the gold-standard network has a density of ~9% (**Figure S1c**), it is clear that a network with 76% density will have an extremely low true positive rate. Therefore, we sought to adopt a more stringent criterion in choosing the edge-rank cutoff for the edges to be used in tumor-type clustering.

Each edge in a tumor type has a consensus rank, computed as the average of ranks from the TOP6 methods. The consensus ranks of these edges are thresholded, separately in each tumor type, with a certain cutoff value to determine the edges that will be used in the unsupervised clustering of tumor types. When the cutoff value is 25, the overlap of edge sets from different tumors is about 35% (computed as the average of the overlap percentages between each pair of tumors) (**Figure S4a**). This high overlap percentage indicates that the most significant edges in individual tumor types are largely shared across all tested tumor types. Indeed, **Figure 6** shows that the majority of these edges are interactions involved in signal transduction pathways that result in the growth and proliferation of neoplastic cells.
For a given rank-cutoff value, the union of edges from the 11 tumor types was obtained; and the consensus edge weights for these edges were used to cluster the tumor types. A principal component analysis along the tumor-type dimension was performed to infer the projections of the data with greatest separation power. This analysis revealed that the variance explained by the first 3 PCs started as high as 75% when the rank-cutoff value was low, but gradually declined as the cutoff was lowered (Figure S2b). In particular, the sum of the first three principal components made two local maxima at cutoff values 375 and 425 before showing a steadily declining trend. Therefore, we decided to perform unsupervised clustering of tumor types using the union of of edges with consensus ranks lower than 425 in the 11 tumor types. The number of edges in each tumor type that pass this criterion is given in Table S4 with the union set having 1008 edges (i.e. the “discovery set” as referred to in the main text).

![Graph](image)

**Figure S4: Determining the edge-rank cutoff for tumor-type clustering.** (a) The left y-axis shows the mean of the percentages of shared edges across all tumors (red dotted line), and between tumors and a randomly ordered edge list (black dotted line). The right y-axis shows the number of edges in the union set from all tumors (blue line). The x-axis shows the cutoff value for the TOP6 consensus-edge-ranks. (b) The left y-axis shows the percentage of variance explained by PC1, PC2 and PC3 (black, red, and orange dotted lines respectively). The right y-axis indicates the sum of the variance-percentages explained by the first three principal components (blue line). The x-axis shows the cutoff value for the TOP6 consensus-edge-ranks.

**Text S5**

**Steps involved in computational network inference**

**Step 1: Running the methods (by varying parameter values if applicable)**

The 13 network inference methods used in this study are shown in Table S1. Nine of these methods require the user to specify at least one parameter value. The parameters and the tested values for the former nine methods are also listed in Table S1. These parameter values were chosen to achieve a good representation of the parameter space, and when the parameter concerns sparsity, to cover the full spectrum from the most sparse networks to fully-connected ones.

Of the remaining four methods, three do not have any user-specified parameters (pearson correlation, spearman correlation, and simple partial correlation). The fourth one, GeneNet, has a shrinkage parameter that is internally optimized in a data-dependent manner, hence not user-specified.
Table S1: The parameters and tested values for each network inference method. R expressions are given for the tested values if the number of values is greater than 5.

<table>
<thead>
<tr>
<th>Family</th>
<th>Method</th>
<th>Parameter type</th>
<th>Tested values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlation</td>
<td>PEARSONCOR</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>SPEARMANCOR</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Partial correlation with inverse covariance</td>
<td>SIMPLEPARCOR</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>GENENET</td>
<td>shrinkage ((\lambda))</td>
<td>(\lambda) internally optimized</td>
</tr>
<tr>
<td></td>
<td>GLASSO</td>
<td>sparsity ((\lambda))</td>
<td>seq(0,1,length=150) {0}</td>
</tr>
<tr>
<td></td>
<td>GLASSO</td>
<td>cross-validation (K), maximum number of components to consider (Ncomp)</td>
<td>K = {3,4,5,10,20}, Ncomp = 30</td>
</tr>
<tr>
<td></td>
<td>RIDGENET</td>
<td>cross-validation (K), shrinkage ((\lambda))</td>
<td>K = {3,4,5,10,20}, (\lambda) internally optimized</td>
</tr>
<tr>
<td></td>
<td>LASSONET</td>
<td>sparsity ((\lambda))</td>
<td>seq(0,1,length=150) {0}</td>
</tr>
<tr>
<td></td>
<td>ELASTICNET</td>
<td>cross-validation (K), weight for (\ell_1) penalty ((\alpha)), sparsity ((\lambda))</td>
<td>K = {3,4,5,10,20}, (\alpha) = seq(0.01,0.99, length=10), (\lambda) internally optimized</td>
</tr>
<tr>
<td>Mutual information</td>
<td>ARACNE-A</td>
<td>nearest neighbor (K), additive penalty ((\varepsilon))</td>
<td>K = {2,3,4,5,6}, (\varepsilon) = seq(0.01,0.99, length=100), seq(0.2,0.4,length=49) {0}</td>
</tr>
<tr>
<td></td>
<td>ARACNE-M</td>
<td>nearest neighbor (K), multiplicative penalty ((\tau))</td>
<td>K = {2,3,4,5,6}, (\tau) = seq(0.1,0.5, length=100), seq(0.5,1, length=49) {0}</td>
</tr>
<tr>
<td></td>
<td>CLR</td>
<td>nearest neighbor (K)</td>
<td>K = {2,3,4,5,6}</td>
</tr>
<tr>
<td></td>
<td>MRNET</td>
<td>nearest neighbor (K)</td>
<td>K = {2,3,4,5,6}</td>
</tr>
</tbody>
</table>

Technical details in running the algorithms include the following:

- For GLASSO and LASSONET, \(\lambda = 0\) runs were removed because the algorithms did not converge.
- For PLSNET, the maximum number of components for each regression model (Ncomp) was fixed at 30 because we empirically observed that the internally-optimized number of components for all \((187 \times 11) 2057\) regression models (number of proteins * number of tumor types) was less than 30 in more than 99% of the cases.
- For RIDGENET, the shrinkage parameter \(\lambda\) was not varied in our tests because it is internally optimized and the value that minimizes the overall mean squared error across all cross-validation folds is used.
- For ELASTICNET, we applied a grid search in each one of the regression models to find the \([K,\alpha]\) pair that optimized the sparsity parameter \(\lambda\). The number of regression models is the same as the number of proteins because a predictive model is built for each protein separately. Thus, there is not a \([K,\alpha]\) pair optimal for all regression models in ELASTICNET; the optimal pair is different for each model. Since it is not possible to report all optimal \([K,\alpha]\) pairs, we report in Table S2 only the median across all regression models.
- The optimal \(\lambda\) in ELASTICNET was determined as the largest \(\lambda\) in the \(\lambda\) set whose cross-validation error was still within 1 standard error of the minimum cross-validation error (i.e. maximum predictive power).
- For ARACNE, CLR, and MRNET, the mutual information between each pair of proteins was computed using entropy estimates from the K-nearest neighbor distances according to
Thus, we optimized $K$, the number of nearest neighbors, for the mutual-information-based methods.

The output of the methods is a matrix of edge weights that we turn into a ranked edge list for subsequent analysis. With 187 proteins, the number of edges for a fully-connected network would be $C(187,2) = 17391$ edges. Sparse methods did not infer a weight for all 17391 edges, in which case the missing edges were assigned a weight of zero.

**Step 2: Plotting precision-recall curves for individual parameter configurations**

The precision-recall curve for a given parameter configuration was constructed by taking the ranked edge list from Step 1 and iterating over the edges from the most significant to the least significant. At each iteration, we obtained a cumulative edge set that included all the edges seen up to and including that iteration, and computed the precision-recall values for these edges. The precision-recall curve consisted of the results from all iterations. In the methods with user-specified parameters, a separate precision-recall curve was generated for each parameter configuration.

Sparse methods required an extra step in order to achieve a ranking of all 17391 edges. The ranks for the zero weight edges had to be artificially created, and were determined with a random permutation.

**Step 3: Computing AUPR and determining the optimal parameter values**

The area under a precision-recall curve (AUPR) was computed with the formula:

$$AUPR(p, r) = \sum_{i=1}^{n-1} \frac{(p_{i+1} + p_i)(r_{i+1} - r_i)}{2}$$

where $n$ indicates the number of edges in the edge list, $p$ and $r$ indicate precision and recall respectively, and the bold font $p$ and $r$ indicate the vector forms.

**Table S2**: Optimal parameter values in the limited-recall case for each method and tumor type.
For the methods with user-specified parameters, the optimal parameter configuration was determined as the one that yielded the largest AUPR. In cases where a method had more than one parameter that needed optimization (ARACNE_A, ARACNE_M and ELASTICNET), we applied a grid search by computing the AUPR for each parameter combination, and then choosing the combination that had the largest AUPR.

As shown in Figure 2a and Figure 2b, we had two different sets of results based on (1) a full-recall case where we measured AUPR in the entire recall range from 0 to 1, and (2) a limited-recall case where we measured AUPR only in the [0,0.1] recall range. We report here (Table S2) the optimal values of the parameters only for the limited-recall case because of its superior performance as discussed in the main text and Figure 2. All subsequent analysis was performed with edges in the [0,0.1] recall range from the runs that used the optimal parameters in the limited-recall case.

**Step 4: Interpretation of edge weights and normalization for CLR**

The range of edge weights from methods in the correlation and partial correlation family is [-1,1]. The magnitude of the weight indicates the strength of the interaction while the sign shows the direction. In contrast, the mutual-information family of methods do not infer the direction of edges, and output weights that are in the range [0,1]. The only exception to the [0,1] range is CLR (context-likelihood of relatedness), which has a lower bound of 0 but no upper bound as the test statistic (edge weight) in this method is essentially a z-score for mutual information values.

To ensure comparability between CLR edge weights and those of other methods, we normalized the CLR weights by their sum and multiplied by 100. This procedure converted the range to approximately [0,1.2]. Even though a [0,1] range was not achieved, we had the desirable property that the distribution of the original CLR weights was maintained. We observe in Figure 3d that CLR clusters together with other mutual-information-based methods. This shows that the normalization procedure for CLR weights rendered CLR results comparable with other mutual-information-based methods.

**Step 5: The principal component analysis (PCA) of the 13 network inference methods**

The optimal parameter configuration determined in Step 3 enabled us to determine the optimal edge weights from each method and for each tumor type. Using these optimal edge weights, we performed principal component analysis for the 13 network inference methods in the study.

To achieve this, we created a matrix $M$ where the methods were placed in the columns and the edge weights were placed in the rows. Edges from the tested tumor types were vertically stacked in order to preclude a tumor-type bias in the PCA results. The stack of each tumor type consisted of the union of limited-recall edges from all methods in that particular tumor type. For instance, the BLCA stack had 6959 edges as shown in the ALL13 union row in Table S3. The edges that were included in $M$ but not found in the limited-recall set of an individual method were assigned a zero weight for that method, indicating that the limited-recall criterion imposed sparsity in the network. Constructed in this fashion, $M$ consisted of 60202 rows (the sum of all columns in the ALL13 union row of Table S3) and 13 columns.

We then applied principal component analysis on $M$ to discover its predominant features in the high-dimensional space formed by the linear combination of method vectors.
Table S3: The number of edges at limited recall for each method and tumor type. The high-performing TOP6 group is shown in the top panel, and the other methods in the bottom panel. All 13 methods are used for the PCA of methods, but only the TOP6 methods are used for the PCA of tumor types.

<table>
<thead>
<tr>
<th>TOP6 methods</th>
<th>BLCA</th>
<th>BRCA</th>
<th>COAD</th>
<th>GBM</th>
<th>HNSC</th>
<th>KIRC</th>
<th>LUAD</th>
<th>LUSC</th>
<th>OVCA</th>
<th>READ</th>
<th>UCEC</th>
<th>UNION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARACNE_A</td>
<td>1249</td>
<td>1276</td>
<td>1057</td>
<td>1108</td>
<td>1110</td>
<td>1038</td>
<td>770</td>
<td>789</td>
<td>1268</td>
<td>1073</td>
<td>1164</td>
<td>6747</td>
</tr>
<tr>
<td>ARACNE_M</td>
<td>1338</td>
<td>1261</td>
<td>1025</td>
<td>1094</td>
<td>1100</td>
<td>1091</td>
<td>769</td>
<td>792</td>
<td>1141</td>
<td>1059</td>
<td>1227</td>
<td>7129</td>
</tr>
<tr>
<td>CLR</td>
<td>1294</td>
<td>1161</td>
<td>1085</td>
<td>1202</td>
<td>1043</td>
<td>1162</td>
<td>930</td>
<td>1042</td>
<td>1099</td>
<td>1143</td>
<td>1173</td>
<td>7020</td>
</tr>
<tr>
<td>LASSONET</td>
<td>1107</td>
<td>1021</td>
<td>1334</td>
<td>1140</td>
<td>1172</td>
<td>1344</td>
<td>1317</td>
<td>1350</td>
<td>1156</td>
<td>1100</td>
<td>1108</td>
<td>8275</td>
</tr>
<tr>
<td>RIDGENET</td>
<td>1174</td>
<td>985</td>
<td>1336</td>
<td>954</td>
<td>1359</td>
<td>1209</td>
<td>1159</td>
<td>1338</td>
<td>1095</td>
<td>1328</td>
<td>994</td>
<td>7547</td>
</tr>
<tr>
<td>SPEARMANCOR</td>
<td>1341</td>
<td>1319</td>
<td>972</td>
<td>1099</td>
<td>1084</td>
<td>1151</td>
<td>842</td>
<td>801</td>
<td>1112</td>
<td>1018</td>
<td>1122</td>
<td>5826</td>
</tr>
<tr>
<td><strong>TOP6 UNION</strong></td>
<td>3141</td>
<td>2981</td>
<td>2604</td>
<td>2944</td>
<td>3042</td>
<td>3269</td>
<td>2780</td>
<td>2441</td>
<td>3162</td>
<td>2700</td>
<td>3014</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Other methods</th>
<th>BLCA</th>
<th>BRCA</th>
<th>COAD</th>
<th>GBM</th>
<th>HNSC</th>
<th>KIRC</th>
<th>LUAD</th>
<th>LUSC</th>
<th>OVCA</th>
<th>READ</th>
<th>UCEC</th>
<th>UNION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELASTICNET</td>
<td>1163</td>
<td>1026</td>
<td>1124</td>
<td>1176</td>
<td>1370</td>
<td>1146</td>
<td>1090</td>
<td>1280</td>
<td>1158</td>
<td>1335</td>
<td>1119</td>
<td>7277</td>
</tr>
<tr>
<td>GENENET</td>
<td>1318</td>
<td>1100</td>
<td>1488</td>
<td>1087</td>
<td>1474</td>
<td>1328</td>
<td>1319</td>
<td>1232</td>
<td>1312</td>
<td>1365</td>
<td>1048</td>
<td>8380</td>
</tr>
<tr>
<td>GLASSO</td>
<td>1242</td>
<td>1157</td>
<td>1148</td>
<td>1198</td>
<td>1303</td>
<td>1205</td>
<td>1328</td>
<td>1411</td>
<td>1150</td>
<td>1480</td>
<td>1087</td>
<td>7054</td>
</tr>
<tr>
<td>MRNET</td>
<td>1351</td>
<td>1186</td>
<td>1064</td>
<td>1287</td>
<td>1117</td>
<td>1193</td>
<td>1077</td>
<td>1131</td>
<td>1261</td>
<td>1244</td>
<td>1263</td>
<td>7923</td>
</tr>
<tr>
<td>PEARSONCOR</td>
<td>1214</td>
<td>1254</td>
<td>1125</td>
<td>1216</td>
<td>1250</td>
<td>1021</td>
<td>775</td>
<td>806</td>
<td>1149</td>
<td>1042</td>
<td>1231</td>
<td>5614</td>
</tr>
<tr>
<td>PLSNET</td>
<td>1301</td>
<td>1020</td>
<td>1222</td>
<td>954</td>
<td>1276</td>
<td>1111</td>
<td>1372</td>
<td>1462</td>
<td>1057</td>
<td>1757</td>
<td>1091</td>
<td>6499</td>
</tr>
<tr>
<td>SIMPLEPARCOR</td>
<td>2090</td>
<td>1139</td>
<td>1570</td>
<td>1496</td>
<td>1656</td>
<td>1364</td>
<td>1430</td>
<td>1780</td>
<td>1322</td>
<td>1411</td>
<td>988</td>
<td>10573</td>
</tr>
<tr>
<td><strong>ALL13 UNION</strong></td>
<td>6959</td>
<td>4534</td>
<td>5460</td>
<td>5890</td>
<td>5882</td>
<td>4793</td>
<td>5155</td>
<td>6247</td>
<td>4653</td>
<td>6376</td>
<td>4253</td>
<td></td>
</tr>
</tbody>
</table>

Step 6: Computing the consensus edge-ranks and consensus edge-weights across methods

The weights assigned to edges and their ranks can vary considerably among the different methods. In order to determine the overall significance and weight of an edge across the methods (but within a given tumor type), we defined the consensus edge-ranks and consensus edge-weights respectively. The consensus rank of an edge is computed as the mean of its ranks from different methods. If one or more of the methods do not have that particular edge in their limited-recall sets, the rank for the edge is assigned to “Total number of limited-recall edges in method + 1”. This rank is not assigned to NA in order to ensure that all methods participate in the consensus rank calculation for that edge. However, the consensus weight of an edge is computed as the mean of its weights from different methods where the missing weights are assigned to NA.

Ranks were accepted as a proxy for significance. Thus, only the magnitudes of edge weights, and not the signs, were taken into account in computing the consensus ranks. Also, the sign for a consensus weight could not be derived from the average of the weights because mutual-information-based methods always yield non-negative weights. Instead, the consensus sign for an edge was inferred from the sign of the Spearman correlation between the relevant antibodies.

The choice of the tie-breaking parameter in ranking was ‘minimum’, meaning that a set of equal weights in the ordered list received the same minimum rank and the next highest weight after this set received its index value (i.e. not the minimum rank of the previous set + 1).

Step 7: PCA and hierarchical clustering on the 11 tumor types

We used the consensus edge weights in the TOP6 methods to perform PCA and hierarchical clustering on the 11 tumor types. The design matrix, denoted with $T$, had tumor types in the columns and edge weights in the rows. However, we restricted the edges to the ones that had TOP6 consensus-rank smaller (more significant) than 425. The preference of the TOP6 methods over
ALL13, and the choice of the consensus edge-rank threshold of 425 were explained previously in Text S3 and Text S4 respectively. The number of edges that pass this threshold in each tumor type is shown in Table S4. The union set of these highly-significant edges has 1008 elements and is referred to as the discovery set.

Table S4: The number of edges that have a consensus-edge-rank value less than 425 for each tumor type. Tumor types are listed in increasing order of the number of edges from left to right.

<table>
<thead>
<tr>
<th>KIRC</th>
<th>LUAD</th>
<th>HNSC</th>
<th>BLCA</th>
<th>GBM</th>
<th>BRCA</th>
<th>UCEC</th>
<th>OV</th>
<th>LUSC</th>
<th>COAD</th>
<th>READ</th>
<th>Union</th>
</tr>
</thead>
<tbody>
<tr>
<td>96</td>
<td>135</td>
<td>138</td>
<td>150</td>
<td>157</td>
<td>161</td>
<td>168</td>
<td>171</td>
<td>243</td>
<td>246</td>
<td>250</td>
<td>1008</td>
</tr>
</tbody>
</table>

We applied principal component analysis on $T$ to discover its predominant features in the high-dimensional space formed by different tumor types. We also performed unsupervised clustering of the tumor types by applying hierarchical clustering on $T$ with Euclidean distance and Ward linkage.

Text S6

Mathematical descriptions of the 13 tested methods

A) Correlation-based methods

A1) Pearson correlation
The Pearson correlation between $X$ and $Y$, denoted $r_{xy}$, is a measure of the strength of the linear relationship between $X$ and $Y$; and can be given as:

$$r_{xy} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y}$$

where $\bar{x}$ and $\bar{y}$ are sample means of $X$ and $Y$, and $s_x$ and $s_y$ are the sample standard deviations of $X$ and $Y$.

A2) Spearman correlation
The Spearman’s rank correlation coefficient is a non-parametric measure of statistical dependence between $X$ and $Y$ assessing the strength of the linear relationship between the ranks of the values in $X$ and $Y$.

Let $X^R$ and $Y^R$ be the ascending ranks of the values in $X$ and $Y$ respectively, where ties are assigned a rank equal to the average of their positions. The Spearman correlation is computed from $X^R$ and $Y^R$ in the same way as the Pearson correlation:

$$r_{xy} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y}$$

where, this time, $x_i$ and $y_i$ are individual components of $X^R$ and $Y^R$, $\bar{x}$ and $\bar{y}$ are sample means of $X^R$ and $Y^R$, and $s_x$ and $s_y$ are the sample standard deviations of $X^R$ and $Y^R$.

From here on, the data matrix is denoted with $X$. It contains $n$ observations for $p$ features, and the features are denoted with $X_1, X_2, \ldots, X_p$.

B) Two formulations for partial correlation

Formulation 1: Inversion of the covariance matrix
The $p \times p$ covariance (or correlation) matrix of the features can be inverted to obtain the $p \times p$ concentration (or precision) matrix denoted with $W$. Let $w_{ij}$ be the concentration value for row $i$ and column $j$. Partial correlations $p_{ij}$ are then the negative of the standardized concentrations where the
standardization is given as:

$$
\rho_{ij} = \frac{w_{ij}}{\sqrt{w_{ii}w_{jj}}}
$$

In graphical Gaussian models, the partial correlations represent the direct interactions between two variables, conditioned on all remaining variables.

Formulation 2: Least Squares Regression
Partial correlations can be obtained from any method based on a least squares regression. Following the exposé in Krämer et al.\textsuperscript{7}, we consider the p linear regression models:

$$X_i = \sum_{j \neq i} \beta^{(i)}_j X_j + \varepsilon$$

for $i = 1, \ldots, p$ where $\varepsilon$ stands for i.i.d noise. The least squares estimate for the regression coefficients in the $i$th model $\hat{\beta}^{(i)} = (\hat{\beta}^{(i)}_1, \ldots, \hat{\beta}^{(i)}_{i-1}, \hat{\beta}^{(i)}_{i+1}, \ldots, \hat{\beta}^{(i)}_p)$ can be found with the optimization:

$$\hat{\beta}^{(i)} = \arg\min_{\beta \in \mathbb{R}^p} \|X^{(i)} - X^{(i)} \beta\|^2$$

where $X^{(i)} \in \mathbb{R}^n$ is the $i$th column of $X$, and $X^{(i)} \in \mathbb{R}^{n \times (p-1)}$ is the matrix obtained from $X$ by deleting the $i$th column. The partial correlation between protein $i$ and $j$ is then estimated as:

$$\hat{\rho}_{ij} = \text{sign} \left( \hat{\beta}^{(i)}_j \right) \sqrt{\hat{\beta}^{(i)}_j \hat{\beta}^{(i)}_i}$$

It can be shown that Formulation 1 and Formulation 2 are equivalent in the sense that they always yield the same partial correlation estimates\textsuperscript{8}.

Methods based on formulation 1:

B1) Simple Partial Correlation

As described in Formulation 1, this method relies on the inversion of the covariance (or correlation) matrix to obtain the concentration (or precision) matrix denoted with $W$. In cases where the number of observations is less than the number of features ($n < p$), covariance matrices might be singular with some zero eigenvalues. Singular matrices can also be inverted with pseudo-inverse approaches to obtain partial correlations.

B2) GeneNet\textsuperscript{9}:

This method first shrinks the covariance matrix towards a low-dimensional target value so that a well-conditioned and positive definite matrix is obtained. The shrunken covariance matrix is then inverted to obtain partial correlations. The shrinkage parameter $\lambda$ is chosen automatically from the data. We describe the shrinkage estimation steps in more detail below.

a) The Ledoit-Wolf lemma: An analytic approach for determining the optimal shrinkage level

Ledoit and Wolf\textsuperscript{10} developed an analytic approach for determining the optimal shrinkage level in large-dimensional estimation problems.

Let $\Psi = (\psi_1, \ldots, \psi_p)$ denote the parameters of the unrestricted high-dimensional model of interest, and $\Theta = (\theta_1)$ the matching parameters of a lower dimensional restricted submodel.

For instance, $\Psi$ could be the mean vector of a p-dimensional multivariate normal, and $\Theta$ the vector of a corresponding constrained submodel where the means are all assumed to be equal (i.e. $\theta_1 = \theta_2 = \cdots = \theta_p$). By fitting each of the two different models to the observed data, we obtain the associated estimates $\bar{U} = \hat{\Psi}$ and $\bar{T} = \hat{\Theta}$. The unconstrained estimate $\bar{U}$ will exhibit a comparatively high variance due to the larger number of parameters that need to
be fitted, whereas its low-dimensional counterpart $T$ will have lower variance but potentially also considerable bias as an estimator of the true $\Psi$.

Instead of choosing between one of these two extremes, the linear shrinkage approach suggests to combine both estimators in a weighted average

$$U^* = \lambda T + (1 - \lambda) U$$

where $\lambda \in [0,1]$ denotes the shrinkage intensity. This is a convex combination of $T$ and $U$, and by setting $\lambda$ equal to 1 or 0, it is possible to recover the original estimates $T$ and $U$ respectively. However, the regularized estimate $U^*$ outperforms both $T$ and $U$ in terms of accuracy and statistical efficiency.

A key question in this procedure is how to select an optimal value for the shrinkage parameter $\lambda$. A systematic way is to choose it in a data-driven fashion by explicitly minimizing a risk function such as mean squared error (MSE):

$$R(\lambda) = E(L(\lambda)) = E\left(\sum_{i=1}^{p} (u_i^* - \psi_i)^2\right)$$

Ledoit and Wolf \cite{Ledoit2004} derived a closed-form solution for choosing $\lambda$ that guarantees minimal MSE without the need of having to specify the underlying distributions, and without requiring computationally expensive procedures such as MCMC, the bootstrap, or cross-validation.

$$\lambda^* = \frac{\sum_{i=1}^{p} Var(u_i) - Cov(u_i, t_i) - Bias(u_i)E(t_i - u_i)}{\sum_{i=1}^{p} E[(t_i - u_i)^2]}$$

which is (apart from some algebraic simplification) the expression given in Ledoit and Wolf \cite{Ledoit2004}.

b) Estimation of the optimal shrinkage intensity

Schafer and Strimmer \cite{Schafer2005} proposed to compute $\lambda^*$ by replacing all expectations, variances, and covariances in the above equation by their unbiased sample estimators. This leads to:

$$\hat{\lambda}^* = \frac{\sum_{i=1}^{p} \hat{Var}(u_i) - \hat{Cov}(u_i, t_i) - \hat{Bias}(u_i)E(t_i - u_i)}{\sum_{i=1}^{p} E[(t_i - u_i)^2]}$$

In finite samples, $\lambda^*$ may exceed the value one, and in some cases it may even become negative. In order to avoid over-shrinkage or negative shrinkage, we truncate the estimated intensity correspondingly, using $\lambda^{**} = \max(0, \min(1, \lambda^*))$ when constructing the shrinkage estimator.

c) Shrinkage estimation of the covariance matrix

In a matrix setting, the equivalent to the squared error loss function is the squared Frobenius norm. Thus,

$$L(\lambda) = \|S^* - \Sigma\|^2_F$$

$$= \|\lambda T + (1 - \lambda) S - \Sigma\|^2_F$$
\[
\lambda^* = \sum_{i=1}^p \sum_{j=1}^p (\lambda t_{ij} + (1 - \lambda)s_{ij} - \sigma_{ij})^2
\]

is a natural quadratic measure of distance between the true \( \Sigma \) and the inferred covariance matrix \( S^* \). In this formula, the unconstrained empirical covariance matrix \( S \) replaces the unconstrained estimate \( U \). In this setting, \( \lambda^* \) takes the form:

\[
\hat{\lambda}^* = \frac{\sum_{i=1}^p \sum_{j=1}^p Var(s_{ij}) - Cov(s_{ij}, t_{ij}) - Bias(s_{ij})(t_{ij} - s_{ij})}{\sum_{i=1}^p \sum_{j=1}^p (t_{ij} - s_{ij})^2}
\]

d) The choice of a target matrix \( T \)

In general, the choice of a target matrix should be guided by the presumed lower dimensional structure in the dataset as this determines the increase of efficiency over that of the empirical covariance. However, it is also a remarkable consequence of the above \( \lambda^* \) formulation that, in fact, any target will lead to a reduction in MSE, albeit only a minor reduction in case of a strongly mis-specified target (in which case \( S^* \) will simply reduce to the unconstrained estimate \( S \)).

In GeneNet, Schafer and Strimmer \(^9\) implemented a shrinkage target that shrinks off-diagonal entries to zero, but leaves diagonal entries intact. This is called the “diagonal, unequal variance” model, and has \( p \) estimated parameters (the variances \( s_{ii} \)).

\[
t_{ij} = \begin{cases} 
  s_{ii} & \text{if } i = j \\
  0 & \text{if } i \neq j
\end{cases}
\]

e) The formulae for the shrunk covariances and correlations

With the choice of \( T \) above, \( Cov(s_{ij}, t_{ij}) \) can take one of two values:

\[
Cov(s_{ij}, t_{ij}) = \begin{cases} 
  Var(s_{ii}) & \text{if } i = j \\
  0 & \text{if } i \neq j
\end{cases}
\]

\( Bias(s_{ij})(t_{ij} - s_{ij}) \), on the other hand, disappears from the equation if we use the unbiased sample estimates for \( s_{ij} \). So, for this choice of \( T \) and \( s_{ij} \), the closed-form solution of \( \lambda^* \) becomes:

\[
\hat{\lambda}^* = \frac{\sum_{i \neq j} Var(s_{ij})}{\sum_{i \neq j} s_{ij}^2}
\]

We can also choose to shrink the correlations instead of the variances. In this case, the formula takes the form:

\[
\hat{\lambda}^* = \frac{\sum_{i \neq j} Var(r_{ij})}{\sum_{i \neq j} r_{ij}^2}
\]

The shrunk correlation and covariance estimates \( r_{ij}^* \) and \( s_{ij}^* \) are then given by:
\[ r_{ij}^* = \begin{cases} s_{ii} & \text{if } i = j \\
 r_{ij}^* \max (0, \min(1, \lambda^*)) & \text{if } i \neq j \end{cases} \]

\[ s_{ij}^* = \begin{cases} s_{ii} & \text{if } i = j \\
 r_{ij}^* \sqrt{s_{ii}s_{jj}} & \text{if } i \neq j \end{cases} \]

**B3) Graphical lasso (GLASSO):**

The graphical lasso is an algorithm that learns the structure of an undirected Gaussian graphical model using an $\ell_1$ regularization to control the number of zeros in the precision matrix $\Theta = \Sigma^{-1}$.

In this algorithm, the $n$ observations in the design matrix $X$ are assumed to be multivariate normal with mean $\mu$ and covariance $\Sigma$. Let $S$ be the empirical covariance matrix, then the Gaussian log-likelihood of the data, partially maximized with respect to the mean parameter $\mu$ is

\[ \log \det \Theta - tr(S\Theta) - \rho\|\Theta\|_1 \]

Here $tr$ denotes the trace and $\|\Theta\|_1$ is the $\ell_1$ norm - the sum of the absolute values of the elements of $\Sigma^{-1}$. GLASSO uses a block-coordinate descent method to maximize this log-likelihood over non-negative definite matrices $\Theta$ and achieves fast convergence.

**Methods based on formulation 2:**

**B4) Partial least squares (PLS) regression:** PLS is a method for supervised dimensionality reduction. The main idea is to build a few orthogonal components from $X^{(i)}$ (all features but the $i$th one), and to use them as predictors in a least squares fit for $X^{(i)}$. In this fashion, a PLS-based regression model is computed for each protein in the dataset.

A PLS component $t = X^{(i)}w$ is a linear combination of the original predictors that have maximal covariance with the response vector $X^{(i)}$ with the additional stipulation that the components are mutually orthogonal. The weights for the $k$th PLS component are found as:

\[ w_k = \arg \max_{\|w\|=1} \text{cov}(X^{(i)}w, X^{(i)})^2 \]

s.t. \[ w_k^T X^{(i)} w_l = 0 \text{ for } l < k \]

The maximum number of components that can be used is specified by the user. In this study, the optimal number for each regression model is determined via k-fold cross-validation.

PLS regression is particularly suited when the matrix of predictors has more features than observations ($p > n$), and when there is multicollinearity among the features. By contrast, standard regression will fail in these cases unless regularized.

**B5) Ridge regression:** This method involves adding an $\ell_2$ penalty term $\lambda \|\beta\|_2^2$ to the least squares criterion where $\lambda > 0$ is a regularization parameter.

\[ \hat{\beta}^{(i)} = \arg \min_{\beta \in \mathbb{R}^p} \|X^{(i)} - X^{(i)}\beta\|^2 + \lambda \|\beta\|_2^2 \]
A ridge regression model is computed for each protein in the dataset and partial correlation \( \hat{\rho}_{ij} \) is computed using regression coefficients \( \hat{\beta}_{ij}^{(i)} \) and \( \hat{\beta}_{ij}^{(j)} \) from the models for protein \( i \) and \( j \). In this study, the optimal value of \( \lambda \) is determined via k-fold cross-validation.

**B6) Lasso regression:** Similar to ridge regression, the estimated regression coefficients are chosen to optimize a penalized least squares criterion. Lasso regression is based on an \( \ell_1 \)-penalty \( \lambda \| \beta \|_1 \) where \( \lambda > 0 \) is the regularization parameter. The least squares criterion then takes the form

\[
\hat{\beta}^{(i)} = \arg \min_{\beta \in \mathbb{R}^p} \| X^{(i)} - X^{(\backslash i)} \beta \|^2 + \lambda \| \beta \|_1
\]

A matrix of partial correlations is estimated based on a lasso regression model for each of the columns (proteins) in the data matrix. The sparsity parameter \( \lambda \) can be estimated using k-fold cross-validation, which results in the oracle penalty for optimal prediction. However, in this study we used a \( \lambda \) value that is common for all regression models in a network and maximizes area under the precision-recall curve (AUPR).

Lasso is not the ideal method in cases where the number of features is much higher than the number of observations (\( p \gg n \)), or where there are highly correlated features. This is because lasso can only select at most \( n \) variables out of \( p \) candidates; and it tends to select one variable from a group of highly correlated features and ignore the others.

**B7) Elastic Net:** This method is a regularized regression method that linearly combines the \( \ell_1 \) and \( \ell_2 \) penalties of the lasso and ridge methods. The estimates from elastic net when fitting a model for \( \hat{\beta}^{(i)} \) are defined by:

\[
\hat{\beta}^{(i)} = \arg \min_{\beta \in \mathbb{R}^p} \| X^{(i)} - X^{(\backslash i)} \beta \|^2 + \lambda_1 \| \beta \|_1 + \lambda_2 \| \beta \|_2^2
\]

The naive version of elastic net finds an estimator in a two-stage procedure: it first finds ridge regression coefficients for each fixed \( \lambda_1 \), and then performs a lasso type shrinkage. This kind of estimation leads to increased bias and poor predictions due to the double amount of shrinkage. To alleviate this problem, Friedman et al. (2010) propose to fit the model using:

\[
\hat{\beta}^{(i)} = \arg \min_{\beta \in \mathbb{R}^p} \| X^{(i)} - X^{(\backslash i)} \beta \|^2 + \| \beta \|_1 + \frac{1}{2} \| \beta \|_2^2
\]

Elastic net encourages the grouping effect and thus can have improved prediction power compared with lasso. The grouping effect is the situation where the regression coefficients of a group of highly correlated variables tend to be equal (up to a sign change if negatively correlated) in a regression model. In particular, the regression method should assign identical coefficients to identical variables if there are any.

**C) Methods based on mutual information:**

The mutual information between protein \( i \) and protein \( j \) is denoted with \( MI(X_i; X_j) \) and abbreviated below with \( MI(i; j) \) if both components are a feature in \( X \). It can be efficiently computed using entropy estimates from K-nearest neighbor distances.

**C1) ARACNE additive (ARACNE-A):** ARACNE-A takes the mutual information matrix of proteins as input, considers each triple of edges independently, and removes the weakest edge \( i \leftrightarrow j \) if

\[
MI(i; j) < MI(j; k) - \varepsilon \quad \text{and} \quad MI(i; j) < MI(i; k) - \varepsilon
\]

are both true. The output is a weighted adjacency matrix of the inferred network. \( \varepsilon \) is a user-specified parameter.

**C2) ARACNE multiplicative (ARACNE-M):** This method works in a way similar to ARACNE-A, but with the difference that the decision criterion to remove the weakest edge involves a multiplicative threshold as opposed to an additive one. ARACNE-M removes the weakest edge \( i \leftrightarrow j \) if
\[ MI(i; j) < MI(j; k) \times (1 - \tau) \quad \text{and} \quad MI(i; j) < MI(i; k) \times (1 - \tau) \]
are both satisfied. \( \tau \) is a user-specified parameter.

**C3) Context-likelihood of relatedness (CLR):** This method computes the score \( \sqrt{z_i^2 + z_j^2} \) for each pair of variables \( i \) and \( j \) where
\[
z_i = \max (0, \frac{MI(i; j) - \mu_i}{\sigma_i})
\]
and, \( \mu_i \) and \( \sigma_i \) are the mean and standard deviation of the mutual information values for all \( MI(i; k) \) such that \( k = 1, \ldots, n \). Given the mutual information matrix, CLR computes the weighted adjacency matrix without a need for any internal parameters.

**C4) MRNET:** This method uses the mutual information between data objects (i.e. proteins) and a feature selection algorithm [minimum-redundancy-maximum-relevance (MRMR)] to infer interactions between objects. It first starts by selecting the variable \( X_i \) having the highest mutual information with the target \( Y \). Then it repeatedly enlarges the set of selected variables \( S \) by taking the \( X_k \) that maximizes \( MI(X_k; Y) - \text{mean}(MI(X_k; X_i)) \) for all \( X_i \) already in \( S \). The procedure stops when the score becomes negative.

**REFERENCES**