Libertas Academica
FREEDOM TO RESEARCH

Open Access: Full open access to this and thousands of other papers at http://www.la-press.com.

Evolutionary Bioinformatics

Netmes: Assessing Gene Network Inference Algorithms by Network-Based Measures

Gökmen Altay[1], Zeyneb Kurt[2], Matthias Dehmer[3] and Frank Emmert-Streib[4]

[1]Biomedical Engineering, Bahçeşehir University, Beşiktaş, Istanbul, Turkey. [2]Department of Computer Engineering, Yildiz Technical University, Davutpasa Campus, 34220, Esenler, Istanbul, Turkey. [3]Institute for Bioinformatics and Translational Research, UMIT- The Health and Life Sciences University, Eduard Wallnoefer Zentrum 1, 6060 Hall in Tyrol, Austria. [4]Computational Biology and Machine Learning Laboratory, Center for Cancer Research and Cell Biology, School of Medicine, Dentistry and Biomedical Sciences, Queen's University Belfast, Belfast, UK.

ABSTRACT: Gene regulatory network inference (GRNI) algorithms are essential for efficiently utilizing large-scale microarray datasets to elucidate biochemical interactions among molecules in a cell. Recently, the combination of network-based error measures complemented with an ensemble approach became popular for assessing the inference performance of the GRNI algorithms. For this reason, we developed a software package to facilitate the usage of such metrics. In this paper, we present *netmes*, an *R* software package that allows the assessment of GRNI algorithms. The software package *netmes* is available from the *R*-Forge web site https://r-forge.r-project.org/projects/netmes/.

KEYWORDS: global network-based measures, local network-based measures, metrics for assessing ensemble datasets, R package for the network-based measures, gene regulatory networks

CITATION: Altay et al. Netmes: Assessing Gene Network Inference Algorithms by Ensemble Network-Based Measures. *Evolutionary Bioinformatics* 2014:10 1–9 doi: 10.4137/EBO.S13481.

RECEIVED: October 24, 2013. **RESUBMITTED:** December 1, 2013. **ACCEPTED FOR PUBLICATION:** December 7, 2013.

ACADEMIC EDITOR: Jike Cui, Associate Editor

TYPE: Original Research

FUNDING: Matthias Dehmer and Frank Emmert-Streib thank the Austrian Science Fund for supporting this work (project P22029-N13). This project is supported by the Department for Employment and Learning through its "Strengthening the All-Island Research Base" initiative.

COMPETING INTERESTS: Author(s) disclose no potential conflicts of interest.

CORRESPONDENCE: gokmen.altay@bahcesehir.edu.tr; v@bio-complexity.com

Introduction

Gene regulatory network inference (GRNI) algorithms are essential in order to interpret large-scale microarray datasets, which allows an understanding of the working mechanism of an organism genome wide. To characterize these GRNI algorithms in detail by evaluating their network inference performances over different aspects, various network-based measures were recently introduced.[1,2] Using these new ensemble-based measures, the differences in the abilities of various GRNI algorithms have been revealed from the highest resolution at the edge level to the lowest resolution at the global level.[1] These analyses helped to gain a better understanding of the available inference algorithms. Assessing GRNI algorithms with respect to network-based metrics became recently popular, and is introduced in Refs. 1 and 2. Therefore, a software package providing an easy-to-use implementation of these network-based measures would be helpful for further promoting their application. For this reason, we present an *R* software package implementation of the network-based measures introduced in Refs. 1 and 2. The software package is developed in the software platform most commonly used by the bioinformatics community, *R*, which is freely available at the web site http://www.R-project.org. We named our package *netmes* as an abbreviation of "**net**work **mes**ures". The package can be downloaded from the *R*-Forge web site http://netmes.r-forge.r-project.org/.

Local network-based measures provide researchers with means to investigate the robustness and weakness of an inference algorithm. There are several studies that investigated the most appropriate approaches to infer gene networks. Some of them propose new inference methods, whereas others are comparative studies.[3–5] In these studies, one method is observed as preferable to others in particular cases and conditions. However, so far, there is no commonly accepted inference method showing the best performance among all the conditions. As the underlying networks of the large-scale real biological datasets are mostly unknown, ie we have restricted information about them, the evaluations of the inference approaches are commonly performed using small-scale real datasets or moderate-scale synthetic datasets.[6,7] Moreover, as the datasets involve pathological conditions, evaluations of the methods become more difficult.

We propose to investigate local characteristics of network structures, which affect the correlation patterns between genes, to overcome the above problems and to facilitate the evaluation of existing inference methods. Hence, existing inference methods could be investigated in a more fair and network-specific manner. In the literature, global measures such as the *F*-score[8] and the area under the receiver operating characteristic (AUROC) curve[9] are utilized to evaluate the inference performances of the methods. As mentioned above, we propose to use several local network-based measures in addition to a global metric, eg an *F*-score.

Global metrics integrate the overall inference performance and provide one output to the researchers. As such, they do not provide precise information of how the global metric is affected by a structural component, ie an individual edge or an individual motif (eg, three-node motif), in the network. Instead, local measures can provide such details. Hence, they allow the assessment of the characteristics of the inference method and are more problem oriented than the global ones.

In the following, ensemble datasets are used to illustrate the results of local network-based measures. For this, the underlying networks (true network) of the ensemble datasets are the same, but their expression values are different. Each synthetic expression dataset is obtained using different kinetic equation parameters in SynTReN.[10]

The network specific measures utilized in this study are number of three-node (gene) motifs; mean true reconstruction rate of each three-node motif; mean true positive rates (TPRs) for each edge; and *Ds*, the degrees of edges around each gene (node). Information regarding these measures is given in detail in the following section.

A detailed definition of this package with examples for each function can be found in the following sections of this study. We developed *netmes* as a command line package that allows great flexibility and also adaptability for integration with different related software packages such as *minet*.[8,11] Most of the figures in Refs. 1 and 2 can be generated by similar datasets using *netmes*.

The organization of the paper is as follows: the measures and their implemented methods included in the software package are explained in Section 2, an example of the usage of the methods is given in Section 3, and finally, a conclusion is given in Section 4.

Methods

An example of the usage of the software package *netmes* is given in the following section. We briefly define the local network-based measures and mention important points and functions of the package.

As it is the nature of the network-based measures, *netmes* requires an ensemble of gene expression datasets instead of just a single dataset, belonging to the same underlying structure of a gene network. Although small datasets can be used for generating the results of metrics of the *netmes* package in principle, larger datasets allow for a more robust assessment. For instance, in Refs. 1 and 2, the number of datasets of each network under consideration was 300. This number corresponds to the size of the ensemble (*E*). To use the real biological datasets that belong to the Dialogue for Reverse Engineering Assessments and Methods (DREAM) project,[12,13] these datasets should be resampled for one underlying network. Hence, there should be an ensemble of the gene expression datasets that have the same underlying network. It is not applicable to use the *netmes* with only one expression dataset for one true network.

We provided five datasets of the network shown in Figure 1 to demonstrate all the network measures of *netmes*; this allows us to obtain similar figures as in Refs. 1 and 2. The example datasets including biological noise were generated by SynTReN.[10] As an exemplary GRNI algorithm, we used relevance networks (RelNet)[14] and stored the inferred adjacency matrix in the same folder as the example datasets as *mim.exp* in each folder.

The ensemble inference performance approach to evaluate GRNI algorithms and also the details of the network-based metrics can be found in Refs. 1 and 2, and for reasons of brevity, we shortly define them in this study.

The first network-based measure evaluates the inferability of the three-gene motifs that involve the different directed edge types. Inferring and examining the three-gene motifs is important because the three-gene motifs are the most crucial fundamental components of large networks.[15] Moreover, mean true reconstruction rate of each motif using TPR and true negative rate (TNR) of each gene in that motif can also be extracted using *netmes* as given in the study.[2]

Another metric, namely mean TPR of an edge, is obtained by dividing the number of times that edge was truly inferred among the datasets by the number of datasets (*E*) in the ensemble.

The next metric, namely *Ds* (degrees of the edges), provides the categorization of the edges and investigates the impacts of activator (ac) (positive effect) and repressor (re) (negative effect) edges on the TPRs. We demonstrate the impacts of the ac (red bars) and re (blue bars) edges on the TPR of the edges in Figure 3(a) and (b) for synthetic and real biological datasets, respectively.

In addition to the above local, network-based metrics, a global metric, namely *F*-score, is also implemented in *netmes*. *F*-score metric is obtained by using precision (p) and recall (r) measures as given in the following equation:

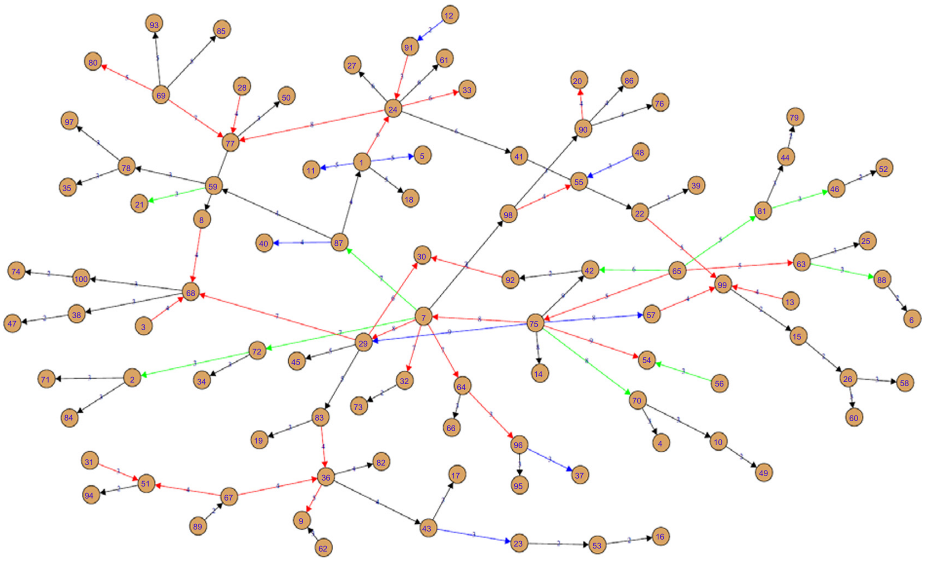


Figure 1. An example of synthetic network we provide together with ensemble datasets for demonstrating the usage of *netmes*.

$$F = \frac{2pr}{p + r} \tag{1}$$

Precision (p) and recall (r) values are obtained by

$$p = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad \text{and} \quad r = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

where TP, FP, and FN, respectively, denote true positive, false positive, and false negative. TP is the number of edges that are inferred by the inference algorithm and actually exist in the underlying network. FP is the number of edges inferred by the inference method, but do not actually exist in the underlying network. Finally, the FN edges actually existed in the underlying network, but they could not be inferred by the inference algorithm.

The definitions of the functions that allow us to obtain the mentioned network-based measures are given below.

- The function *TPRall* provides one of the network-based metrics that assesses the influence of ac and re edges on the inference performance. TPR means true positive rate and TP denotes number of ac and re edges that are truly inferred by the GRNI algorithm.
- The function *plotDs* provides a measure that assesses the influence of the in-degree and the out-degree (*Ds*) of an edge on the inferability.
- The functions *motifcount*3 and *plotMotifs* find motifs and provide a measure that evaluates the influence of three-node motif types on the inferability.
- The function *visnet* allows the visualization of networks with respect to their inferability of each edge by classifying the edges into four colors, namely black, blue, green, and red, showing the best to the worst inferability of each edge, respectively. In-degree and out-degree values can also be labeled on the edges of this network.

In addition to the functions given above, there are two main functions, namely *evalnetworks* and *evalnetworksNames*. They involve all the measures' implemented functions in which all the calculations are performed. If the network under consideration is a synthetic network and nodes have numbers as names, then *evalnetworks* is used as it is designed to read from the *.sif* file of the output of SynTReN.[10] If the network under consideration is a real biological network and nodes have real gene names, then *evalnetworksNames* should be used, as it reads from the *.sif* file in a different manner.

Examples and a Demonstration of the Package's Usage

To make it easier to understand *netmes*, we provide a detailed example in this section. Moreover, a detailed explanation of the datasets is also available along with a step-by-step usage of each function including their outputs. To learn the usage of *netmes* and employ it efficiently, one should read the previous section and also have the two main papers[1,2] in hand to see the real applications of each function of the package.

In the folder of the source package (with the extension *tar.gz*) *netmes*, we provide an *EXAMPLE.txt* file that can be copied and pasted to run all the important functions of *netmes*. It will give a quick run of all the functions. To be able to use the *netmes*, the dependent package *minet* 3.1.1 must be installed. For earlier versions of *minet*, one needs to follow the instructions in *EXAMPLE.txt*. We placed example datasets and their true networks in the *data* folder of *netmes* for running the example simulations. The outputs of the functions can be seen step-by-step in Figures 2–5, respectively, for a quick demonstration of the abilities of *netmes* for a given synthetic network.

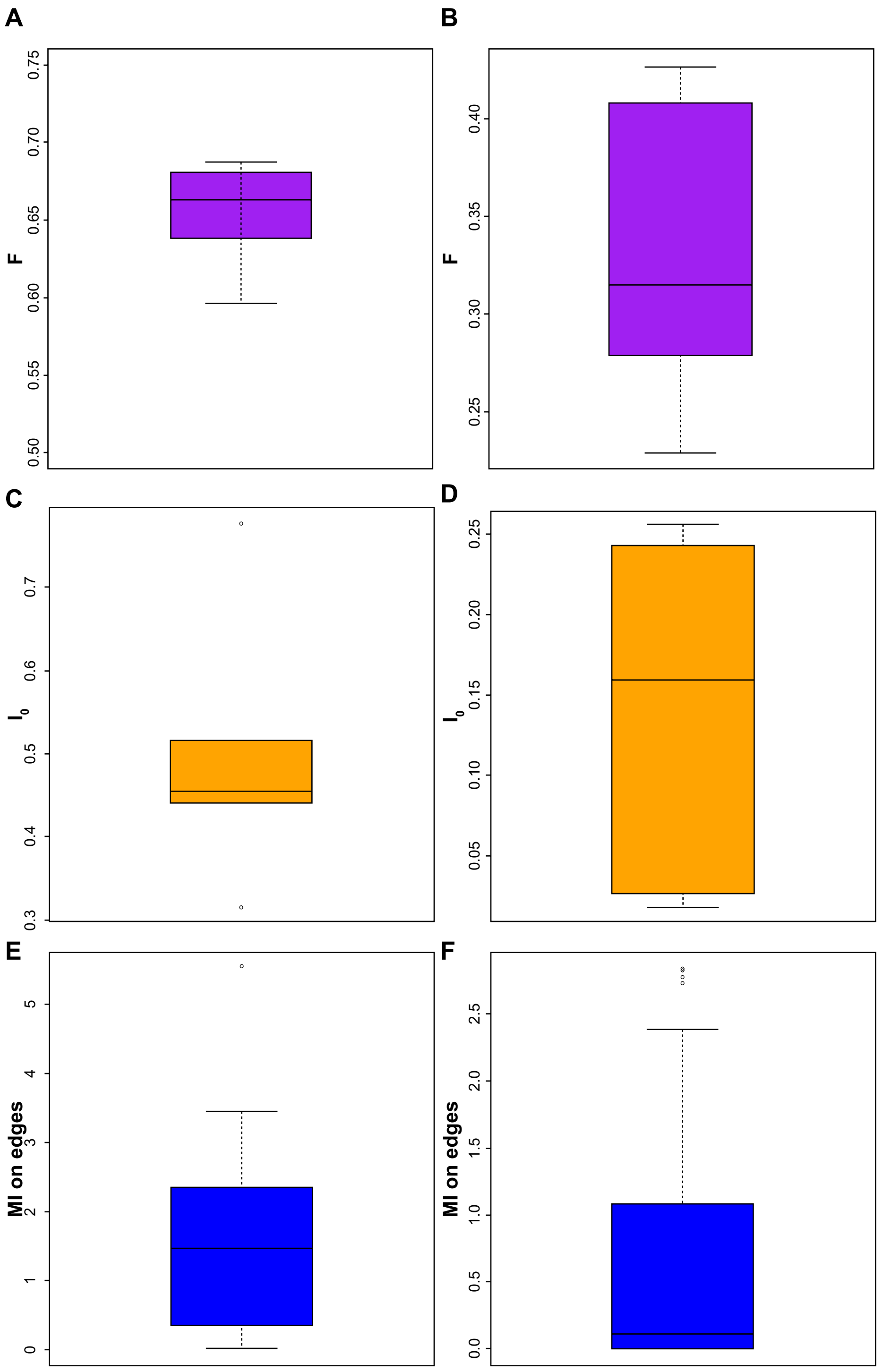


Figure 2. (**A**) *F*-scores obtained for the synthetic network. (**B**) *F*-scores obtained for a real biological sub-network of *E. coli*. (**C**) I_0 threshold values of MI values of the edges for the synthetic network. (**D**) I_0 threshold values of MI values of the edges for the real biological network. (**E**) Average MI values of the edges in the synthetic network. (**F**) Average MI values of the edges in the real biological network.

The true network of these datasets is a directed acyclic graph (DAG)-like network consisting of 100 nodes (genes). SynTReN was used to generate the datasets from the network, which generated expression data in steady-state conditions using dynamic equations in Michaelis–Menten and Hill enzyme kinetics with 100 samples. For using different datasets, formats and names of the datasets and files should be adjusted, exactly in the same way as the ones in the folder *data*. Or alternatively, names of the files and datasets can be changed appropriately in the source files in the folder of the *R* package. Therefore, one needs to be thorough in the naming and formatting of the dataset so that *netmes* can find the desired data files.

Section 3.1 involves the definitions of the datasets. Section 3.2 explains the usage of the functions with examples.

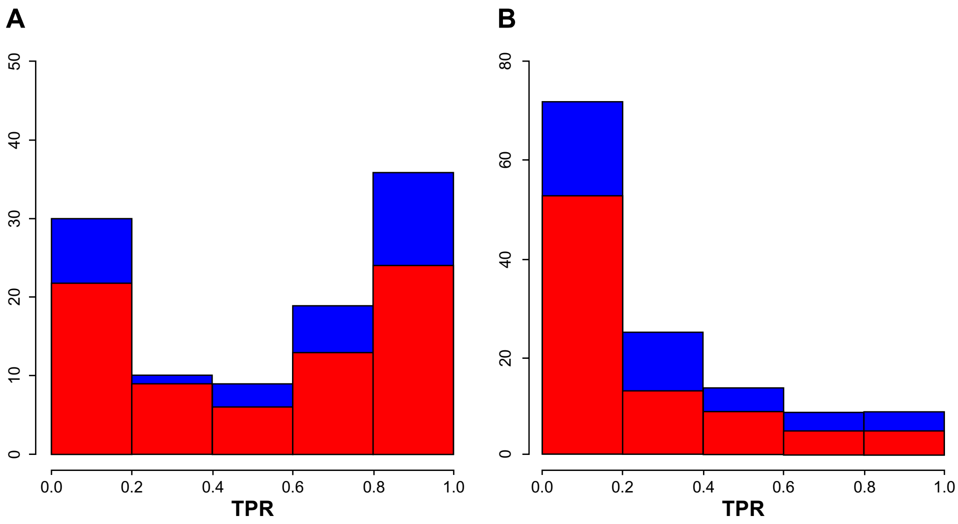


Figure 3. TPR values of the edges in the (**A**) synthetic and (**B**) real biological networks.

Usage of the datasets. *netmes* is designed to run with the datasets in the format of the data files that are automatically generated as a result of the expression data generator software application SynTReN.[10] Also, each of the ensemble datasets must be saved in a different folder, and the names of these folders are given as integer numbers from 1 to *N*, where *N* is the number of datasets that will be used for the ensemble analysis. This means that *N* is the size of the ensemble. It is noteworthy to point out that each of the datasets has the same underlying network and consequently, allows an ensemble analysis of inference performance of a GRNI algorithm on this network. The name of each dataset in the numbered folders must be *nn*100_*nbgr*0_*hop*0.3_*bionoise*0.05_*expnoise*0.0_*corrnois*0.0_*neighAdd_maxExpr*1_*dataset.txt*. This is the standard output file name by SynTReN. If desired, this name can be changed by modifying the functions in *netmes*; however, we designed our package to work conveniently with SynTReN because it can be considered as a gold standard data generator package for simulation purposes. The resulting data files are tab delimited and their structure is matrix like, where the columns correspond to samples and the rows to genes. The first row of the matrix from column 2 to $S + 1$, where S is the sample size, contains the list of names of each sample. The first column of the matrix from row 2 to $G + 1$, where G is the number of genes, contains the list of names of each gene. If the underlying network of the dataset is a biological network (no synthetic network), then only the gene names are in the first column. If the network is a synthetic network, such as a DAG, then the gene names become numbers of the form *gene_number* (eg gene_098).

The second file that must be in the numbered data folders along with the datasets is the corresponding mutual

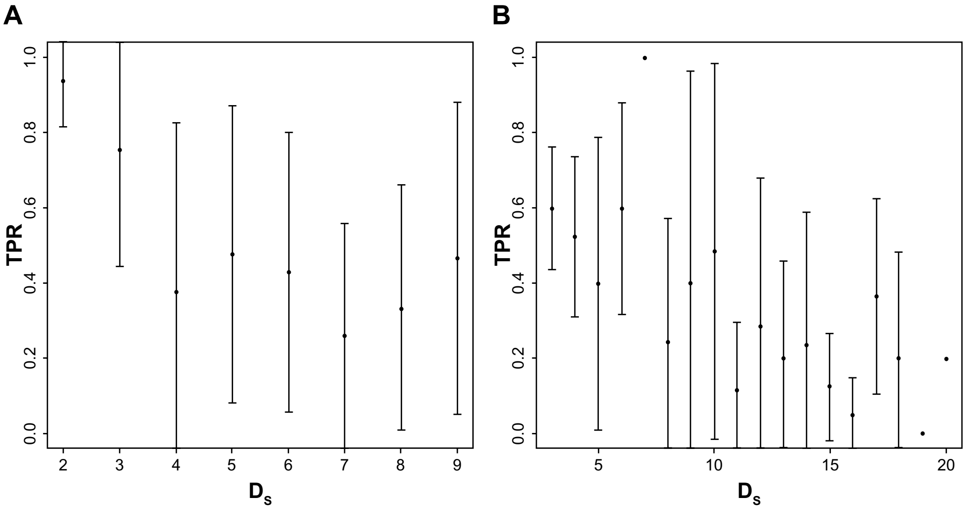
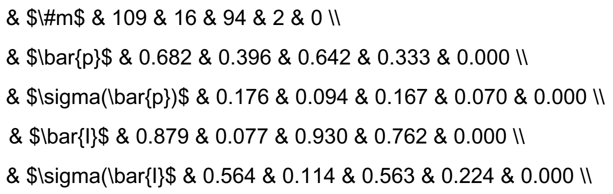


Figure 4. Total in-degree and out-degree of edges (*Ds*) in the (**A**) synthetic and (**B**) real biological networks.



```
& $\#m$ & 109 & 16 & 94 & 2 & 0 \\
& $\bar{p}$ & 0.682 & 0.396 & 0.642 & 0.333 & 0.000 \\
& $\sigma(\bar{p})$ & 0.176 & 0.094 & 0.167 & 0.070 & 0.000 \\
& $\bar{I}$ & 0.879 & 0.077 & 0.930 & 0.762 & 0.000 \\
& $\sigma(\bar{I}$ & 0.564 & 0.114 & 0.563 & 0.224 & 0.000 \\
```

Figure 5. Automatically generated latex table for three-node motifs.

information (MI) matrix, which is also a tab-delimited file. For the example datasets, we obtained it by using the *R* package *minet*[11] and application of the RelNet[14] GRNI algorithm. The name of the MI matrix is *mim.exp*, and the first row of the file contains the names of the genes. The first column goes from the second row till the end and also contains the names of the genes, and the rest are the MI values among the genes, where the MI values lead to a square matrix. The diagonal of this matrix is set to zero.

The third file is for representing the true network connections, and it is saved with the extension *.sif* as a standard input file format to SynTReN. Its name and folder can be named arbitrarily as it can be referred to by a variable, ie *infile* = "*./netmes/data/networkDAG.sif*." However, the format of the file must be such that each row has three elements: the first one is the gene name, the second one is the interaction type (ac for activator and re for repressor), and the third one is again a gene name. "*cpxAR ac rotA*" or "*gene_*006 *re gene_*097" is an example. All pairwise interactions in a network are written as rows in this file. We direct the reader to see the example files in the data folder of *netmes*. To run simulations, we provided five different datasets for a quick but informative overview. In Refs. 1 and 2, the number of datasets was 300 for a thorough analysis of the inference performances of GRNI algorithms. Moreover, the biological network is a sub-network from the transcriptional regulatory network of *Escherichia coli*[16,17] and no synthetic network in our demonstration.

Usage of the methods. We run all the functions of *netmes* with the example datasets in the folder data of the package. Practicing the examples and supporting them with a real application will help for a better understanding and usage of *netmes*.

To indicate an *R* command line, we place symbol *R*> at the beginning of a line and change the background of each command line to gray. We start by demonstrating the functionality of *netmes* for a synthetic network. For this reason, one needs to use the function *evalnetworks*, whereas *evalnetworksNames* is used for biological networks, as mentioned above.

We assign the number of different datasets for the synthetic network:

R > E = 5

The path of the *.sif* file that includes the true network is pointed to a variable:

R > infile = "./netmes/data/networkDAG.sif"

The folder path of the datasets, which are in the numbered folders, is assigned to a variable:

R > infilepath = "./netmes/data/syn/"

The true network matrix is formed from the *.sif* interaction map file and set to a variable:

R > net = readNet(infile, infilepath)

Now the initialization process is finished, and we can run the core function *evalnetworks* to estimate the *F*-scores for each dataset and also other related metrics that can be seen in the value of this function in *R* or in the vignette file under the *inst* folder in *netmes*:

R > res = evalnetworks(E, net, infilepath)

Here care must be taken to make sure that the dependent package *minet* is the version 3.1.1 as mentioned above, because otherwise an error message appears. If an earlier version of *minet* is installed, then it is needed to change expression *fs*[*ind*] to *fs*[*ind*,] in the functions *evalnetworks* and *evalnetworksNames* in the *R* folder of the *netmes* package. This will also require you to repack the *netmes* folder by the command "*R CMD build netmes*" in the command line of a terminal.

We now have the ensemble values in the variable *res* and can plot these. Figure 2(a) shows the boxplot for the *F*-scores obtained from the example datasets of the synthetic DAG-like network. Accordingly, Figure 2(b) shows the *F*-scores for a small real biological network. Figure 2(a) is an example that is similar to the real application in Figure 1.B in Ref. 1 and the middle Figure 2 in Ref. 2. In Refs. 1 and 2, real application means that real biological gene expression datasets are used in the experiments.

Figure 2(c) shows the boxplot for the optimal threshold MI values, I_0, which were obtained from maximizing the *F*-score for each of the dataset. Real applications of this example are Figure 1.A in Ref. 1 and the top Figure 2 in Ref. 2. Also, the boxplot showing the optimal I_0 threshold value for the real biological datasets is given in Figure 2(d).

Figure 2(e) shows a boxplot of the average MI values in the network per edge as described in Equation (3). This is also shown in Equation (2) in the study:[2]

$$\begin{aligned}&\mathrm{MI}_{ij}^{\mathrm{avg}}\left(i \to j \quad \text{in NETWORK}\right)\\&= \frac{1}{E}\sum_{k}^{E}\mathrm{MI}_{ij}\left(k \,|\, i \to j \quad \text{in NETWORK}\right)\end{aligned} \tag{3}$$

where *E* is the number of expression datasets that have the same underlying network in the ensemble of the datasets, as mentioned above.

Figure 2(e) corresponds to the results shown in the bottom Figure 2 in Ref. 2 and Figure 1.C in Ref. 1. Average MI values of the edges obtained from the real biological datasets are illustrated in Figure 2(f).

Boxplots in Figure 2(a)–(f) are obtained by using the default boxplot function (ie the modified boxplot) in *R*. The smallest datum above the "1st quartile − 1.5 × IQR" corresponds to the lower whisker; the greatest datum below the "3rd quartile + 1.5 × IQR" denotes the upper whisker where IQR is the interquartile range that is obtained by subtracting the 1st quartile from the 3rd quartile. Any data points under the 1st quartile − 1.5 × IQR and above the 3rd quartile + 1.5 × IQR are denoted by circles in Figure 2.

So far, we have performed some global ensemble analysis of the network. We continue with local network-based measures. To plot the inferability of each edge, we created the function *visnet*, where each edge is classified into four colors, black, blue, green, and red, showing the best to the worst inferable edges, respectively:

```
R > Sm = sumdeg(net)
```

```
R > visnet(net, c("no"), res$rmat, Sm)
```

The visualization format can be changed in the output GUI, and we observed the best representation by using the Fruchterman and Reingold graphical layout. The color and the size of the nodes and edges can also be changed in the output figure. These changes can be saved as an *.eps* file. An example of this function is illustrated in Figure 1 in this paper, and figures of real applications can be found in Figure 5 in Ref. 1 or Figures 5–7 in Ref. 2.

We will now plot the effect of ac and re edges on the TPR of edges in the network, as shown Figure 2 in Ref. 2. We used the color red for ac edges and blue for re edges. We used the function *TPRall* to compute the TPR of ac and re edges first:

```
R > TPRs = TPRall(res)
```

From this, we can obtain the boxplot of the results for a visual inspection of them. An example of it can also be seen in Figure 3(a) that is obtained by

```
R > postscript(file = "TPRall_syn.eps", horiz = FALSE,
paper = "letter")
```

```
R > hist(TPRs$TPReALL, 5, ylim = c(0,50), xlim = c(0,1),
col = "blue", main = "", ylab = "", xlab = "TPR", cex.lab = 1.7,
cex.axis = 1.6)
```

```
R > hist(TPRs$TPReALLac, 5, ylim = c(0,50), xlim = c(0,1),
add = TRUE, col = "red")
```

```
R > dev.off()
```

For the real biological network, the effect of ac and re edges on the TPR values is given in Figure 3(b). Again, red is used for *ac* edges and blue is used for *re* edges.

Another network-based measure calculates the variable *Ds*, which is the sum of the out-degree of node *i* plus the in-degree of node *j* that shows the effect of the neighbor edges on the inferability of each edge in the network, similar to Figure 3 in Ref. 1. An example of this can be seen in Figure 4(a), generated with the following function that results also in a saved figure with the name *Ds.eps*:

```
R > Ds = plotDs(TPRs$TPReALL, Sm, fig = TRUE)
```

Also, for the real biological network, TPR with respect to the sum of in-degree and out-degree neighbors of each edge is given in Figure 4(b).

We now use the function *motif_count*3 to search for the inferability of three-node motifs, which are given in Figure 1 of the supplementary file:[2]

```
R > res2 = motifcount3(net, res$rmat, res$aveMImat)
```

The obtained results can be automatically converted into a table in the latex format by

```
R > plotMotifs(res2)
```

The resulting latex text is shown in Figure 5.

If the network under consideration is a biological network, then it will have a slightly different *.sif* file and to perform a similar analysis, the function *evalnetworksNames* needs to be used instead of *evalnetworks*. Evaluation figures for a real biological dataset of *E. coli* bacterium, similar to the previous figures for the synthetic network, can be obtained by running the scripts in the *EXAMPLE.txt* file in the *netmes* folder of the source package (with the extension *tar.gz*). Figures 2(b), (d), (f), 3(b), and 4(b) illustrate the *F*-scores, optimal I_0 threshold values of MI scores, average MI values of the edges in the network, TPR values, and *Ds* measures of these small real biological ensemble datasets, respectively.

The inferred network obtained from RelNet for the real biological datasets is given in Figure 6.

Conclusion

In this study, we introduced the *R* software package *netmes*, aiming to facilitate the usage of network-based metrics for ensemble datasets.

A large variety of global and local network-based metrics can be used to evaluate the inference performances of GRNI algorithms when using an ensemble of gene expression datasets. In this paper, we show only some of these.

One of our measures evaluates three-gene motifs, which are the building blocks of many large-scale biological networks. Another metric, *Ds*, considers the direction of the edges in the

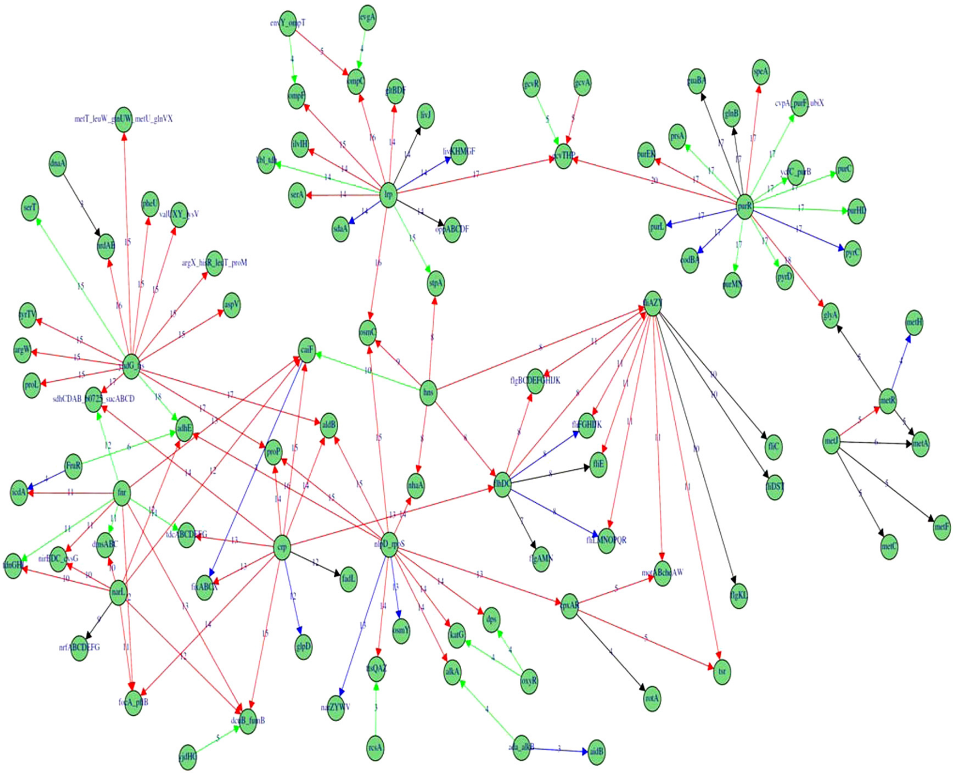


Figure 6. The inferred real biological network using the RelNet GRNI algorithm.

given network and categorizes the edges according to their types (ie whether they are ac or re edges). Hence, this measure allows us to observe the impacts of ac and re edges on TPR. Another metric illustrates the mean TPR of each edge among the gene pairs. Finally, an edge-level evaluation provides us with the highest resolution achievable.

By using the proposed local measures, inference algorithms can be evaluated in a network-specific manner.

Acknowledgments

We thank Shailesh Tripati for testing *netmes* and giving us valuable feedback.

Author Contributions

GA and FES conceived and designed the experiments. They also prepared the software package. ZK and MD checked the functions of the software package and corrected some of the functions appropriately. GA, ZK, MD, and FES wrote the manuscript. GA, ZK, MD, and FES agreed with the manuscript results and conclusions, and jointly developed the structure and arguments for the paper. All authors reviewed and approved of the final manuscript.

DISCLOSURES AND ETHICS

As a requirement of publication the authors have provided signed confirmation of their compliance with ethical and legal obligations including but not limited to compliance with ICMJE authorship and competing interests guidelines, that the article is neither under consideration for publication nor published elsewhere, of their compliance with legal and ethical guidelines concerning human and animal research participants (if applicable), and that permission has been obtained for reproduction of any copyrighted material. This article was subject to blind, independent, expert peer review. The reviewers reported no competing interests.

REFERENCES

1. Altay G, Emmert-Streib F. Revealing differences in gene network inference algorithms on the network-level by ensemble methods. *Bioinformatics*. 2010;26(14):1738–44.
2. Emmert-Streib F, Altay G. Local network-based measures to assess the inferability of different regulatory networks, *IET Syst Biol*. 2010;4(4):277–88.
3. Bansal M, Di Bernardo D. Inference of gene networks from temporal gene expression profiles. *IET Syst Biol*. 2007;1(5):306–12.
4. Hache H, Lehrach H, Herwig R. Reverse engineering of gene regulatory networks: a comparative study, *EURASIP J. Bioinform Syst Biol*. 2009: 617281.
5. Marbach D, Prill RJ, Schaffter T, Mattiussi C, Floreano D, Stolovitzky G. Revealing strengths and weaknesses of methods for gene network inference, *Proc Natl Acad Sci USA*. 2010;107(14):6286–91.
6. Olsen C, Meyer PE, Bontempi G. On the impact of entropy estimation on transcriptional regulatory network inference based on mutual information. *EUROSIP J Bioinf Syst Biol*. 2009:308959.
7. de Matos Simoes R, Emmert-Streib F. Influence of statistical estimators of mutual information and data heterogeneity on the inference of gene regulatory networks. *PLOS One*. 2011;6(12):e29279.
8. Meyer PE, Kontos K, Bontempi G. Information-theoretic inference of large transcriptional regulatory networks. *EUROSIP J Bioinf Syst Biol*. 2007:79879.
9. Husmeier D. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics*. 2003;19(17):2271–82.
10. Van den Bulcke T, Leemput KV, Naudts B, et al. SynTReN: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC Bioinformatics*. 2006;7(1):43.

11. Meyer PE, Lafitte F, Bontempi G. minet: A R/Bioconductor package for inferring large transcriptional networks using mutual information. *BMC Bioinformatics*. 2008;9(1):461.
12. Stolovitzky G, Califano A. *Reverse Engineering Biological Networks: Opportunities and Challenges in Computational Methods for Pathway Inference*. Vol 1115. Boston, MA: Wiley-Blackwell; 2007.
13. Stolovitzky G. The challenges of systems biology: community efforts to harness biological complexity. In: Kahlem P, Califano A, eds. Hoboken, NJ: Wiley-Blackwell; 2009.
14. Butte AJ, Kohane IS. Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. *Pac Symp Biocomput*. 2000;418–29.
15. Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D, Alon U. Network motifs: simple building blocks of complex networks. *Science*. 2002;298(5594):824–7.
16. Shenn-Orr S, Milo R, Mangan S, Alon U. Network motifs in the transcriptional regulatory network of *Escherichia coli*. *Nat Genet*. 2002;31:64–8.
17. Ma HW, Kumar B, Ditges U, Gunzer F, Buer J, Zeng AP. An extended transcriptional regulatory network of *Escherichia coli* and analysis of its hierarchical structure and network motifs. *Nucleic Acids Res*. 2004;32:6643–49.