Exploring the operational characteristics of inference algorithms for transcriptional networks by means of synthetic data

Koenraad Van Leemput*, Tim Van den Bulcke+, Thomas Dhollander+, Bart De Moor+, Kathleen Marchal&+1 and Piet van Remortel*

* ISLab (Intelligent Systems Lab), Universiteit Antwerpen, Antwerpen, Belgium.

+32 3 265.33.57, {koen.vanleemput | piet.vanremortel}@ua.ac.be

+ ESAT-SCD, K.U.Leuven, Heverlee, Belgium.

+32 16 32.17.09, {tim.vandenbulcke | thomas.dhollander | bart.demoor}@esat.kuleuven.be

& CMPG, Dept. Microbial and Molecular Systems, K.U.Leuven, Heverlee, Belgium.

+32 16 32.96.85, kathleen.marchal@biw.kuleuven.be

Abstract

The development of structure learning algorithms for gene regulatory networks depends heavily on the availability of synthetic datasets that contain both the original network and associated expression data. This paper reports the application of SynTReN, an existing network generator which samples topologies from existing biological networks and uses Michaelis-Menten and Hill enzyme kinetics to simulate gene interactions. We illustrate the impact of different aspects of the expression data on the quality of the inferred network. The tested expression data parameters are network size, network topology, type and degree of noise, quantity of expression data and interaction types between genes. This is done by subjecting three well-known inference algorithms to SynTReN datasets. The results show the power of synthetic data in revealing operational characteristics of inference algorithms which are unlikely to be discovered by means of biological micro-array data only.

Keywords: Gene regulatory network, simulated data, network inference, gene expression data.

[1]corresponding author

1 Introduction

The development of algorithms to infer the structure of gene regulatory networks based on expression data is an important subject in Systems Biology research and machine learning. Validation of these algorithms requires benchmark datasets for which the underlying network is known. Since experimental datasets of the appropriate size and design are usually not available, there is a need to generate well-characterized synthetic datasets that allow thorough testing of learning algorithms in a fast and reproducible manner. As a first step towards simulating the complete cellular behavior, computer models of gene interaction networks can be created. In such models it is possible to take the topology of a gene interaction network as a starting point, and use computer simulation of the gene interactions and transcription to extract the corresponding expression data under various experimental settings.

In this paper the application of *SynTReN* [18] will be illustrated. SynTReN is an existing generator of transcription regulatory network models and of the associated expression data. Instead of using random graph models, SynTReN uses topologies that are generated based on previously described source networks, allowing better approximation of the statistical properties of biological networks. We make use of the different parameters of SynTReN to illustrate the power of synthetic datasets in exploring the application scope of a number of published inference algorithms. In this way this paper aims at illustrating both the added value of biologically plausible synthetic expression data in general, and the flexibility of SynTReN in particular. This also sheds light on the question to which extent the knowledge of the statistical properties of real interaction networks influences the confidence in inference algorithms for transcriptional networks. The reported experiments support the need for accurate and high-performance computer models of living systems in order to leverage Systems Biology research.

2 Experimental approach

The main goal of the presented analyses is to demonstrate how the use of synthetic data can provide substantial insight into the performance of a network inference algorithm and its relationship to properties of the input data. The experimental setup that will be used, is shown in Figure 1: in

a first step, a synthetic gene interaction network is generated based on a chosen network topology and interaction type. Next, expression data is generated that corresponds to the gene interactions dictated by the network. This involves setting various levels of noise and structuring the resulting dataset in experiments and samples per experiment. An *experiment* in this context involves the selection of a set of external conditions which are subsequently perturbed and fed into the transcriptional gene network. For each experiment, SynTReN produces a number of micro-array datasets, referred to as *samples*. The resulting dataset is then used as input to a number of different network inference algorithms, which produce a candidate network of genes or gene modules. In a final step, both the original network topology and the inferred candidate are compared through a derived adjacency matrix. The calculated performance metrics are summarized by means of plots and discussed in Section 6.

The resulting performance metrics are used only as a relative score to differentiate across different experimental settings. For several reasons, they were not intended to quantitatively assess the performance of the inference algorithms or to compare algorithms with each other. First of all only default settings were used when running the inference algorithms. These settings are most likely not the most optimal parameter settings for every experiment. Second, the metrics only assess the presence and absence of edges in the inferred network. A more sophisticated or in-depth evaluation of the inferred networks could give more insight in the performance of an algorithm for a specific experiment, but it less suitable for the high-throughput nature of this study.

With the setup described above, we aimed at investigating the following questions related to several parameters of the expression data that will be supplied to the inference algorithms:

Network size: how trustworthy are inference results of ever larger interaction networks, given abundant expression data? This is important since it relates to the applicability of inference methods for large networks, and links to the discussion of the need of heterogeneous data sources to infer truly large networks.

Graph topology: to what extent does the quality of inferred networks depend on the statistical nature of the topology of interaction networks? Several previous studies[10, 11] have reported the use of synthetic expression data from random graph models to validate network inference. However, previous work has shown that these random graph models do not resemble real biological networks in every respect[18]. This raises the question of the performance of algorithms that have

been designed with only random network topologies as a benchmark. To what extent does good performance on a specific class of networks generalize to other classes? Can increasing knowledge of topological properties interaction networks substantially increase the ability to design better inference algorithms?

Noise type and amount: what is the effect of various types and amounts of noise in expression data on the quality of inference results? The experiments give an indication to the added value of a reduction of noise in high-throughput experiments. Other experiments try to answer the relation between the estimate of biological noise in datasets and the confidence expressed in inference results.

Amount of expression data: what is the marginal gain in inference quality by spending resources on obtaining extra datasets of micro-array experiments? Do inference algorithms reach a maximum inference quality, after which supplying more expression data becomes pointless? The reported experiments investigate this issue in relation to the amount of noise present in the expression data provided.

Interaction type: what is the impact of different interaction types between genes? More specifically: to what extent do highly non-linear interactions act as a buffer to mask the activity of downstream genes in an interaction cascade? Should such a buffering effect occur it can be expected that inference results downstream of such genes are of lower quality, which can be taken into account when validating results on real-world data.

3 Generating synthetic regulatory networks

For the generation of synthetic gene expression data, a software package *SynTReN* [18] was used. SynTReN produces synthetic transcriptional regulatory networks (TRNs) and corresponding simulated micro-array datasets. The SynTReN generator can be parameterized by the topology and size of the synthetic network, the gene interaction types and their parameters, the number of simulated experiments and the levels of biological, experimental and input noise.

Network topologies are either generated by selecting subnetworks from a previously described biological network (e.g. the *E. coli* network [16]) or by generating a topology using a random graph model. Two different strategies to select a connected subgraph from a source graph are

implemented in SynTReN: *neighbor addition* and *cluster addition* [18]. Different random graph models can also be generated using SynTReN. In this paper, four different random graph models were used: Erdös-Rényi (ER) [6], Watts-Strogatz (WS) [19], Albert-Barabási (AB) [1, 2] and Bollobás (DSF) [4, 5]. For a brief description of these models, we refer to the supplementary information available on the web.

After generating the topology, transition functions representing the regulatory interactions between the different genes are assigned to the edges in the network. Non-linear functions (based on Michaelis-Menten and Hill kinetics) model gene regulation in steady-state conditions [7, 9, 11].

Each interaction can be either activating or inhibiting; some examples of activating transition functions are given in Figure 2. By choosing appropriate parameter values, different types of transition functions can be obtained, such as *linear* (function 6 in Figure 2), *linearlike* (function 3 and 4), *steep* (function 1 and 8) or *sigmoidal* (function 7).

A gene expression dataset is obtained from this interaction network by simulating the synthetic network under different simulated experimental conditions and subsequently sampling data from it. Each sampled dataset represents a simulated micro-array dataset.

During sampling, three types of noise are introduced:

1. *Biological noise*: biological noise is defined as the noise that propagates through the gene network, corresponding to stochastic variations in gene expression that are unrelated to the applied experimental procedures.

2. *Experimental noise*: micro-arrays are also subject to experimental noise. Contrary to biological noise, this experimental noise does not propagate through the gene network but represents errors in measuring the correct mRNA concentration.

3. *Input noise*: input noise is only relevant when measuring multiple micro-arrays under the same external conditions. This type of noise represents the variation in gene expression level of the input genes under the same external conditions.

4 Inference algorithms

Three different network inference algorithms were used. A short description of each is given below. For further details of each tool we refer to the relevant publications.

Genomica [13] uses expression data to construct a network of interacting modules, each consisting of co-regulated genes, their regulators, and the conditions under which regulation takes place (i.e. the regulatory program). The method is based on probabilistic relational models [14], a relational extension to Bayesian networks.

SAMBA [17] is a bi-clustering algorithm that groups genes by means of a clustering of similar expression patterns in the input data over a subset of input conditions. The algorithm is based on a graph theoretic approach and statistical modeling of the data. The subsets of genes that jointly respond to specific conditions can be interpreted to form a module network.

Aracne [3, 10] explicitly infers a gene interaction network from micro-array expression profiles on the basis of *mutual information* [12] between the genes. Two parameters control the pruning of candidate interactions. The *mutual information (MI) threshold* eliminates edges that have low mutual information and thus removes edges between genes that have a nonzero MI solely due to random sampling of the data samples. The *data processing inequality (DPI)* is used to eliminate the least strong interaction of a triplet of interactions and thus for example eliminates transitive interactions between two genes if the interaction is indirectly through a third gene.

All inference algorithms were run at their default parameter settings. The only exception to this rule was Aracne, which requires the specification of the DPI threshold parameter. The authors suggest a DPI threshold level between 0 and 0.15. In our experimental setup, a DPI level of 0.10 was used as a typical threshold and a DPI level of 0 was used for comparison (indirect interactions are always pruned).

5 Evaluating performance of inference

In this section we discuss the score metrics that were used to indicate the quality of the inferred gene interaction networks. The optimal solution to the inference problem is a reconstruction of the complete interaction network with its topology and causal relations between genes. In reality exact

reconstruction of the causal relations in the network is impossible, since generators like SynTReN produce steady state data. To infer causality additional information is required such as time-course gene expression data, perturbation experiments with gene knockouts or other types of data sources such as location data and motif data. As a consequence, the input network topology has to be transformed to an undirected network to allow comparison with the reconstructed networks. Additionally, both Genomica and SAMBA generate module networks, which are a partitioning of the genes in subsets that belong together. This modular output has to be translated into a corresponding gene regulatory network to allow comparison with the original interaction network, which does not explicitly contain the concept of modules. To this end both the known network topology and all of the reconstructed networks are converted into a gene-by-gene binary adjacency matrix for further analysis.

For the original SynTReN network, this matrix is constructed by calculating the shortest undirected path length between every pair of genes in the original network, and comparing this distance to a threshold. If the distance is less than or equal to the threshold the corresponding two entries (due to symmetry) in the adjacency matrix are set to 1. If a threshold of 1 is used this procedure results in a matrix representing an undirected version of the original network. However, because of the way the adjacency matrix is constructed for the module networks, it is necessary to use a higher threshold to allow meaningful comparison. By default a threshold of 2 was used, in effect grouping genes that are linked by at most one intermediary gene into a "module".

For the module networks, the adjacency matrix is constructed as follows: starting from the gene-by-gene identity matrix M, the entries M_{ij} and M_{ji} are assigned a value of 1 if at least one module is present that contains both genes i and j. A regulatory module therefore corresponds to a clique in the graph, with a connection between every pair of genes in the module. In this way, the modules will probably contain many indirect interactions. If genes a and b share a common regulator c and all three genes are members of the module X, the adjacency matrix will contain non-zero entries for the indirect interaction between a and b.

Aracne's output can be transformed into an adjacency matrix in a straightforward way because it already contains a single p-value for each selected pair of genes that share a regulatory interaction in the inferred network. Genes that are not connected according to Aracne do not appear in its output. Therefore, the selected gene pairs correspond exactly to the 1's in the adjacency matrix,

while all other entries are 0, except for those on the diagonal (following [15]). Aracne directly infers an interaction network, rather than a module network, and is capable of efficiently pruning indirect interactions based on the MI between each pair of genes. However, the procedure to construct the adjacency matrix from the original network actually rewards the algorithm for finding indirect interactions when using a path length threshold of 2. By lowering it to a value of 1 it is possible to effectively require Aracne to infer the exact – but undirected – original network topology and to evaluate its ability to effectively prune indirect interactions.

Comparison between the resulting adjacency matrices was performed by counting the corresponding and conflicting entries in both matrices and calculating *sensitivity* (also known as *recall*), *specificity*, and *precision* (also known as *positive predictive value*). As a summary metric, the *F-measure* was used, which is the harmonic mean of precision and recall. Additional details about the scoring procedure and characteristics of the performance metrics can be found in the web supplement to this paper.

6 Results

6.1 The effect of network size

To assess the influence of network size on inference performance, subnetworks of different sizes were selected from the *E. coli* source network [16], according to the *cluster selection* method [18]. The size of the selected subnetworks varied from 50 to 300 edges in intervals of size 10 and linear interaction functions were assigned to all edges in the network. All topnodes acted as external input genes. A large expression dataset with a low noise level (1000 experimental conditions, 1 sample each, 0.05 experimental noise) was generated for each of the networks.

For each algorithm the F-measure was plotted in function of increasing network size. All three algorithms show clearly different quantitative behavior when faced with increasingly larger networks.

As illustrated in Figure 3(a), SAMBA's overall performance as indicated by the F-measure, stays the same regardless of size, although there is clearly a larger variation for smaller networks. Its precision, however, decreases with increasing network size (see Figure 3(b)).

As illustrated in Figure 3(c), Genomica produces lower F-measure scores as the networks to infer grow, but levels off eventually. Genomica did not output results for many of the larger networks at the settings that were used, even after relatively long running times. In cases where the running time exceeded 12 hours, the experiments were stopped and these data points are missing from the analysis. Aracne's performance also decreases for larger networks, but does not seem to level off within the range that was tested (Figure 3(d)).

In summary, inference performance drops as network size increases, even if enormous amounts of data are available. This decay however is not as drastic as what might be expected, especially for Aracne which shows no sign of substantially reaching the end of its application scope for larger networks. In order to infer larger networks with high confidence, it seems that micro-array data alone is insufficient and that complementary approaches such as adding additional data sources or incorporating domain knowledge are needed.

6.2 The effect of network topology

In the following series of experiments smaller networks of approximately 50 genes were used. The effect of graph topology on the performance of the inference algorithms was studied by creating gene networks with a topology derived from different random graph models that are known to approximate biological networks. The models that were studied are the Erdös-Rényi [6] (ER) random graph model, the Albert-Barabási [1] (AB) scale-free network model, the Watts-Strogatz [19] (WS) small-world model and the directed scale free (DSF) model as described by Bollobás [5]. Apart from these random models, a set of topologies was also generated by selecting subnetworks from the previously described *E. coli* [16] and *S. cerevisiae* [8] transcriptional network, as described in [18]. These latter graphs more closely approximate the topological properties of known transcriptional networks [18].

For each random graph model a small number (approximately 10) of representative graphs topologies was created by sweeping the model parameters across a range of values. The parameter values were varied around a default set of values that was chosen to produce graphs whose properties are close to the *E. coli* network (this was investigated in a previous study [18]). For every generated graph topology, a series of 10 experimental runs was performed. In each run several

distinct synthetic gene networks with the same network topology were created by assigning linear interaction functions to the edges in the graph (see also Section 3) and synthetic gene expression data was generated for each of the resulting gene networks. Each dataset consisted of 100 simulated experimental conditions, with biological noise set at 0.05, and no experimental or input noise. For every inference algorithm about 5600 expression datasets were supplied for inference, covering 92 different network topologies.

Figure 4(a) indicates that SAMBA's performance does not differ markedly between any of the graph classes with regard to sensitivity or specificity. Sensitivity is low across graph classes, because few true positive interactions are found.

Both Genomica (Figure 4(b)) and Aracne (Figure 4(c)) achieve very similar sensitivity and specificity values for both AB and WS graphs, causing these graph models to be largely co-located on both the plots (upper left). Other random graphs, like the ER and DSF graphs, cover a much wider range of sensitivity/specificity combinations. The *S. cerevisiae* and *E. coli* subnetworks also show significant overlap with each other but clearly cover a different sensitivity/specificity than the AB and WS graphs.

For both Genomica and Aracne it is interesting to note that some of the *E. coli* and *S. cerevisiae* subnetworks show a specificity of zero at high sensitivity. This is due to the fact that the particular network topologies involved – which were in all cases subnetworks selected with the cluster addition method – are in fact one big star-like structure, with a single regulator regulating a large number of genes. Since all gene-pairs are separated by at most two edges, the adjacency matrix for a path length threshold of two is an all-one matrix. As a consequence, the number of true negative interactions will always be zero and thus result in a specificity value of zero.

Summarized, the topology of the network can have a strong impact on the performance of an inference algorithm. This should be taken into account when evaluating inference algorithms using synthetic datasets. It is encouraging to note that for two of the algorithms tested here, namely Genomica and Aracne, the inference results on topologies that are known to be biologically more plausible, are better.

6.3 The effect of various noise types

The impact of noise was investigated by generating expression data from the same network under a variety of noise conditions. The network topology that was used to generate data, was a subnetwork selected from *E. coli* [16] with 50 genes and 76 edges in which 5 genes acted as external inputs. In each of the 20 experimental runs, linear interaction functions were assigned to the edges in the network and expression data was sampled for a range of noise levels. The tested range varied from 0.0 to 1.0 in intervals of 0.05 for each of the noise types (biological noise, experimental noise and input noise). The nominal dataset consisted of 100 experimental conditions, represented by 200 array datasets (2 samples per condition).

In the following paragraphs, because of the large number of tested parameter combinations, not all of the results are accompanied by figures. Additional figures are available in the web supplement. Also note that for a linear interaction sampled at a 1.0 noise level, the correlation between the regulator and the regulated gene is still quite high (approximately 0.7) if the expression data sufficiently spans the range of possible regulator values. This means that at a noise level of 1.0, the data still contains much information compared to genuinely random data.

Since the effect of input noise only manifests itself as a difference between multiple samples measured under the same external conditions, its effect is very small under the given experimental conditions for each of the three algorithms. Creating datasets with a smaller number of experimental conditions and a larger number of samples could possibly provide more insight, but this analysis was not performed to limit the computational costs.

In the results for SAMBA, the F-measure linearly decreases with increasing levels of biological noise. A similar decrease in F-measure is observed with increasing experimental noise. Detailed analysis shows a gradual increase in precision to a level of 1 – the level at which all inferred interactions are correct – reflecting the fact that SAMBA outputs module networks with very small or even empty modules at high noise levels. In that case the only non-zero entries left in the adjacency matrix are those on the diagonal which results in a high precision. The sensitivity however drops to zero at high noise, globally resulting in decreasing F-measure. The effect of input noise on the performance is much less pronounced.

In Genomica's case, the F-measure shows a very small decrease across the tested range of biolog-

ical noise and experimental noise levels. Precision (Figure 5(a)) decreases linearly with increasing bio- or experimental noise, while sensitivity remains at an almost constant level (Figure 5(b)). The highest F-measure values are not achieved with zero noise, but with a small amount of noise. A possible explanation for this observation is the fact that Genomica uses a correlation based clustering as an initial step to infer the module network. In zero noise conditions such a correlation based approach has the tendency to overconnect the network, because indirectly interacting genes will show as strong a correlation as those that are directly interacting. Similar behavior was observed when testing other correlation based methods, the data of which is not included in this study.

For Aracne, the effect of noise on the quality of the inferred output network depends strongly on the scoring procedure. When using the default path length threshold of 2 (see Section 5), the F-measure shows a reverse-S-shaped decrease in F-measure with increasing levels of biological noise (Figure 6(a)). This decrease is much more linear for experimental noise (Figure 6(c)). Again, input noise has a less pronounced effect (value around 0.7) although the F-measure scores show a very small decrease at high noise levels.

On the other hand, as explained in Section 5, it is possible to require Aracne to precisely infer the original network and evaluate its ability to remove indirect interactions. In this set-up, the algorithm's DPI-threshold was set to 0.00 for most strict pruning of indirect interactions. In this case the F-measure (Figure 6(b)) shows a very different behavior. The performance metrics, which are at a low level when little or no biological noise is present, show a steep *increase* with rising levels of noise to a plateau from which they then decline very slowly. A possible explanation for this behavior is that when almost no noise is present the mutual information (MI) between indirectly connected genes is almost equal to that of the direct interactions, while the propagating biological noise will tend to decrease the MI between indirectly connected genes relative to those that are directly connected, resulting in more efficient pruning. In this setting the effect of both experimental and input noise is also quite different: the F-measure remains unchanged and at a low level (see Figure 6(d)), with an increase in precision offset by a decrease in sensitivity. Possibly, Aracne can not prune the indirect interactions efficiently in this case, because these types of noise do not propagate through the network and tend to decrease both the MI of the direct and that of the indirect interactions in equal amounts.

In conclusion, Aracne is quite robust in the face of increasing biological noise. When no

biological noise is present, the performance is high if the algorithm is not penalized for indirect interactions (Figure 6(a)). With much biological noise, the performance remains high –and in a way becoming even better– because indirect interactions are effectively removed. For experimental noise, on the other hand, performance degrades gradually under the first scoring regime (Figure 6(c)). However, this is not due to the removal of indirect interactions as it is for biological noise, since performance is low across the entire noise range for the second scoring regime (Figure 6(d)).

In summary, a clear effect of biological noise and experimental noise on the inference performance of all algorithms is observed, with input noise having the smallest influence. This illustrates that noise is an important, and sometimes required factor for network inference. All three noise types available in the SynTReN generator provoked a qualitatively different inference behavior, which supports their adoption in generators for synthetic data in general.

6.4 The effect of available expression data

This series of experiments aims at showing the impact of the amount of available expression data on inference results. The *E. coli* subnetwork with cluster addition of size 50 genes was chosen as the source network topology and 20 experimental runs were done by assigning linear interaction functions to the edges and subsequently sampling data from the network. Datasets varied in size form 5 simulated experimental conditions to 200 with a step size of 5. To evaluate the dependency between the number of arrays that are needed to infer the network, and the amount of noise in the dataset, the simulated expression data contains experimental noise levels ranging from 0 to 1 with step size 0.2. All other parameters were kept constant during the experiments.

Figure 7 shows the F-measure for SAMBA, F-measure and precision for Genomica and F-measure for Aracne. Error bars were calculated but are not shown on the plots to improve readability. Their relative size was comparable to those of the preceding figures such as Figure 6. The behavior of the tested algorithms when presented with increasing data and noise differs substantially, so we start by splitting up the discussion per algorithm.

SAMBA's F-measure scores increase gradually when the algorithm is provided with increasing amounts of expression data (Figure 7(a)). No performance plateau was reached within 200 experiments, and this behavior is consistent across noise levels. More detailed analysis (plots not shown)

shows that precision gets lower with an increasing amount of data, but is offset by an increase in sensitivity, leading to the increasing F-measure on Figure 7(a). This implies that more interactions are recovered, of which an ever increasing ratio are false (precision drops) but this is compensated by an increasing ratio of correctly inferred interactions (sensitivity increases) leading to the overall increasing F-measure. The higher the noise level the consistently lower the inference quality.

For Genomica (Figure 7(b)) a similar increase in F-measure is observed for increasing input data, but a plateau a reached after which further addition of expression data does not result in significant changes in performance. With increases in experimental noise this behavior is generally similar although the maximum score is slightly lower. An interesting effect occurs when examining the score for zero noise: as shown in Figure 7(b) the highest F-measure values are reached for a noise level of 0.2, not in absence of noise as might be expected. This is due to a significantly lower sensitivity at a noise level of 0 than at small positive noise (plot not shown). Precision values on the other hand are ranked according to noise level as expected, with the highest values for 0 noise, as shown in Figure 7(d).

The results for Aracne in Figure 7(c) show that it performs quite well even with relatively small datasets, reaching maximum plateau performance quickly. When noise is introduced the behavior changes qualitatively, with larger amounts of data required to achieve good results. No plateau is reached in this case.

In conclusion, there is a substantial but decreasing benefit of supplying more expression datasets when trying to infer interaction networks. The algorithms tested behave differently, sometimes reaching a maximum performance in noiseless datasets (Aracne), sometimes benefitting from noise (Genomica). The absolute scores achieved also differ markedly. Except for the special case where Aracne infers a network from noiseless data, reaching the inference plateau takes enormous amounts of data, given the fact that the target network aimed for consists of only 50 genes.

6.5 The effect of different interaction types

An additional experiment was performed to assess the impact of different interaction types between genes. More specifically, we were interested in examining to what extent a highly non-linear interaction can act as a buffer to mask the activity of downstream genes in an interaction cascade.

Should such a buffering effect occur, we can expect inference results downstream of such genes to be inferior, which could be taken into account when validating results on real-world data. A detailed description of the experimental setup and results, along with figures, can be found in the web supplement.

The experiment shows that highly non-linear interactions do act as a buffer, masking the activity of downstream genes. In the case of a single non-linear interaction this effect is dependent on the type of algorithm and the output that it generates. When multiple non-linear interactions are present it becomes increasingly difficult to detect the relationship between all the genes in an interaction cascade, irrespective of the inference algorithm.

7 Conclusion

In this paper we have used synthetic data to gain insight into the different aspects of the expression data on the quality of the inferred network. Three different types of inference algorithms were tested, each of which exhibited different behavior to varying parameters of the synthetic data. The parameters tested were network size, network topology, type and degree of noise, availability of expression data and interaction types between genes.

No attempt was made to explicitly fine-tune the parameters of the inference algorithms, since the main goal of this work was to study the effect of different properties of the data on the inference procedure in a qualitative sense, and not a quantitative performance comparison of the algorithms themselves.

Experiments show that inference performance drops as network size increases, even if enormous amounts of data are available. This decay however is not as drastic as what might be expected. In order to infer larger networks with high confidence it seems that complementary approaches such as adding additional data sources or incorporating domain knowledge are needed.

The topology of the network can have a strong impact on the performance of an inference algorithm. This should be taken into account when evaluating inference algorithms using synthetic datasets. It is encouraging that for two of the algorithms tested in this study, Genomica and Aracne, the inference results on topologies that are known to be biologically more plausible, are better.

Different types of noise clearly have distinct effects on the inference performance of each al-

gorithm. Experiments show that noise is an important, and sometimes required factor during inference. All three noise types available in the SynTReN generator provoked a qualitatively different inference behavior, which supports their adoption in generators for synthetic data in general.

We observed a substantial but decreasing benefit of providing more expression data when trying to infer interaction networks. The tested algorithms behave differently, sometimes reaching a maximum performance where adding more arrays becomes pointless. Reaching the inference quality plateau requires enormous amounts of data relative to the size of the inferred network.

The results show the added value of synthetic data in revealing operational characteristics of inference algorithms unlikely to be discovered by means of biological micro-array data, and make a strong case for computer models of biological systems in leveraging Systems Biology research.

8 Acknowledgments

This work is partially supported by: 1. IWT projects: GBOU-SQUAD-20160; GBOU-020176; 2. Research Council KULeuven: GOA Mefisto-666, GOA-Ambiorics, IDO genetic networks, Center of Excellence EF/05.007 SymBioSys; 3. FWO projects: G.0115.01 and G.0413.03; 4. IUAP V-22 (2002-2006). Thomas Dhollander is research assistant of the Fund for Scientific Research Flanders (FWO Vlaanderen).

References

[1] Albert, R., & Barabási, A. (2000). Topology of evolving networks: local events and universality. *Physical Review Letters*, 85, 5234–5237.

[2] Albert, R., & Barabási, A. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74, 47–97.

[3] Basso, K., Margolin, A. A., Stolovitzky, G., Klein, U., Dalla-Favera, R., & Califano, A. (2005). Reverse engineering of regulatory networks in human B cells. *Nature Genetics*, 37, 382–390.

[4] Bollobás, B. (1985). *Random graphs*. New York: Academic Press.

[5] Bollobás, B., Borgs, C., Chayes, C., & Riordan, O. (2003). Directed scale-free graphs. *Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms*, 132–139.

[6] Erdös, P., & Rényi, A. (1959). On random graphs. *Publicationes Mathematicae Debrecen*, 6, 290–297.

[7] Fersht, A. (1985). *Enzyme structure and mechanism*, vol. 2. New York: W.H. Freeman and Company.

[8] Guelzim, N., Bottani, S., Bourgine, P., & Kepes, F. (2002). Topological and causal structure of the yeast transcriptional regulatory network. *Nature Genetics*, 31, 60–63.

[9] Hofmeyr, J. H., & Cornish-Bowden, A. (1997). The reversible Hill equation: how to incorporate cooperative enzymes into metabolic models. *Computer Applications in the Biosciences*, 13, 377–385.

[10] Margolin, A. A., Nemenman, I., Basso, K., Klein, U., Wiggins, C., Stolovitzky, G., Dalla Favera, R., & Califano, A. (2006). Aracne: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, 7 (Suppl 1), S7.

[11] Mendes, P., Sha, W., & Ye, K. (2003). Artificial gene networks for objective comparison of analysis algorithms. *Bioinformatics*, 19, II122–II129.

[12] Papoulis, A. (1984). *Probability, Random Variables, and Stochastic Processes*, chap. 15. second ed., Columbus: McGraw-Hill.

[13] Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., Koller, D., & Friedman, N. (2003). Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, 34, 166–176.

[14] Segal, E., Taskar, B., Gasch, A., Friedman, N., & Koller, D. (2001). Rich probabilistic models for gene expression. *Bioinformatics*, 17, S243–S252.

[15] Shakhnovich, B. E., Reddy, T. E., Galinsky, K., Mellor, J., & Delisi, C. (2004). Comparisons of predicted genetic modules: identification of co-expressed genes through module gene flow. *Genome Informatics Ser Workshop Genome Informatics*, 15 (1), 221–8.

[16] Shen-Orr, S. S., Milo, R., Mangan, S., & Alon, U. (2002). Network motifs in the transcriptional regulation network of Escherichia coli. *Nature Genetics*, 31, 64–68.

[17] Tanay, A., Sharan, R., Kupiec, M., & Shamir, R. (2004). Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. *Proceedings of the National Academy of Sciences U.S.A*, 101 (9), 2981–2986.

[18] Van den Bulcke, T., Van Leemput, K., Naudts, B., van Remortel, P., Ma, H., Verschoren, A., De Moor, B., & Marchal, K. (2006). SynTReN: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC Bioinformatics*, 7, 43.

[19] Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393, 440–442.

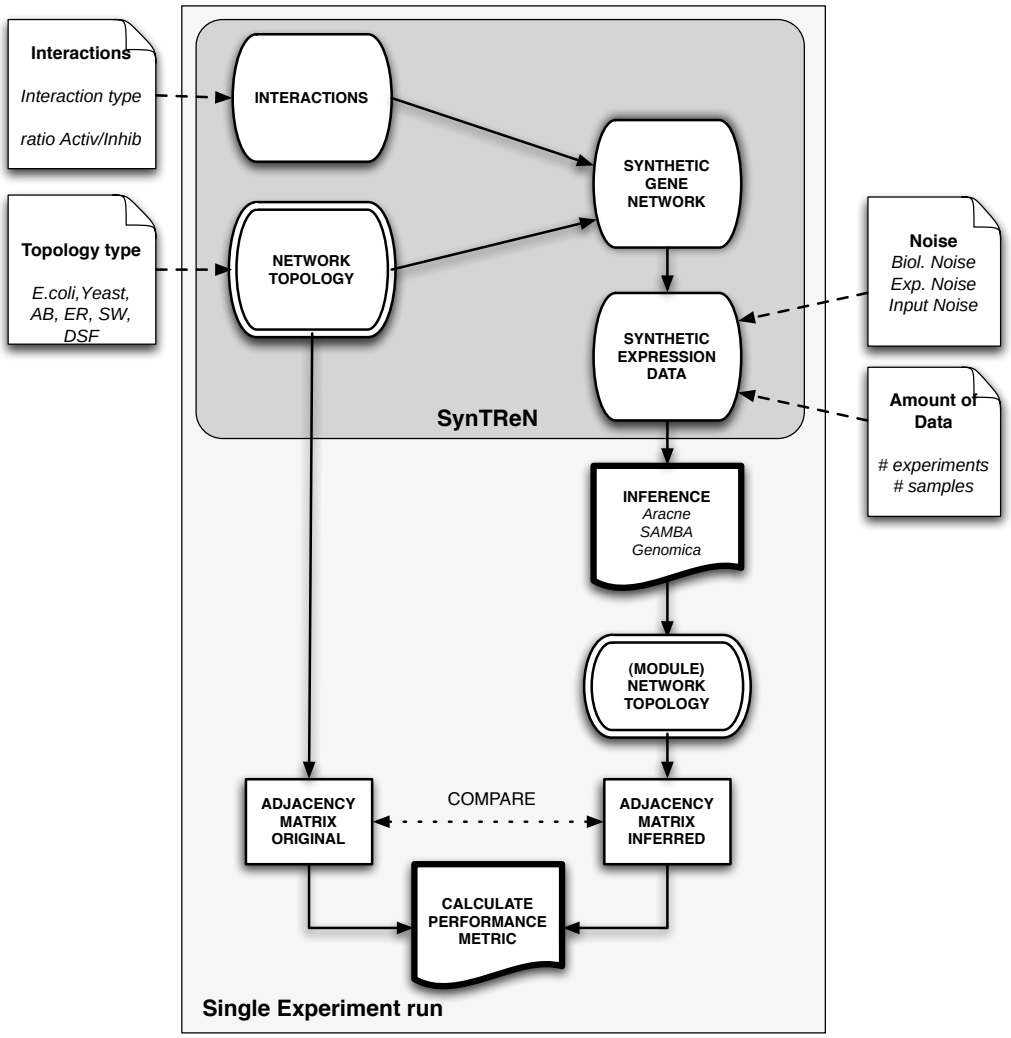


Figure 1: Overview of the experimental setup used: a synthetic interaction network is generated and expression data is derived, which is used as input to several inference algorithms. The inferred networks are then compared to the original by means of calculations on corresponding adjacency matrices. Parameters to be defined are shown on the outer edge of the diagram.

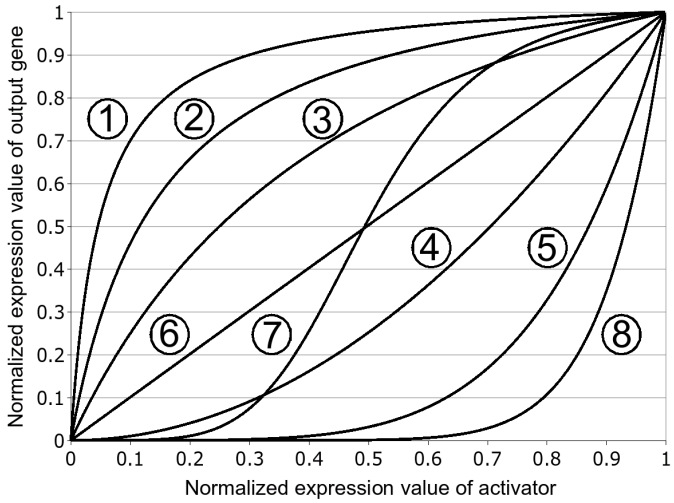
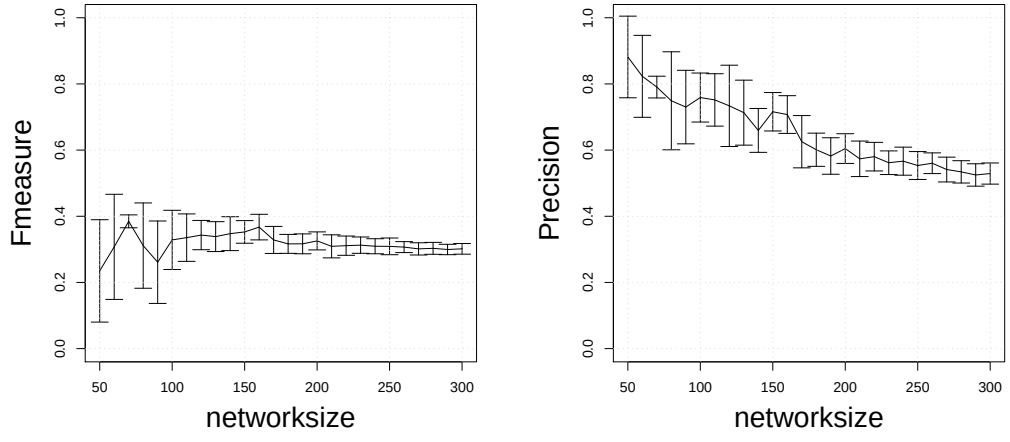
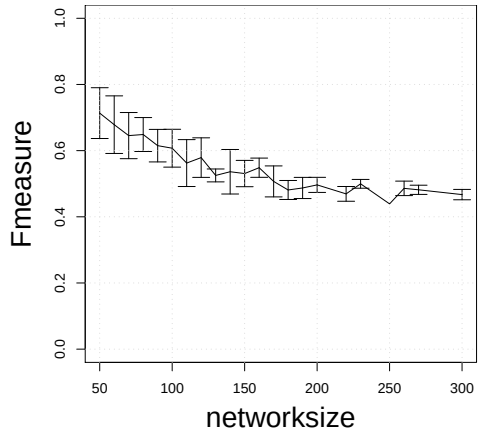


Figure 2: Examples of one-input activating transition functions for different combinations of the kinetic parameter.

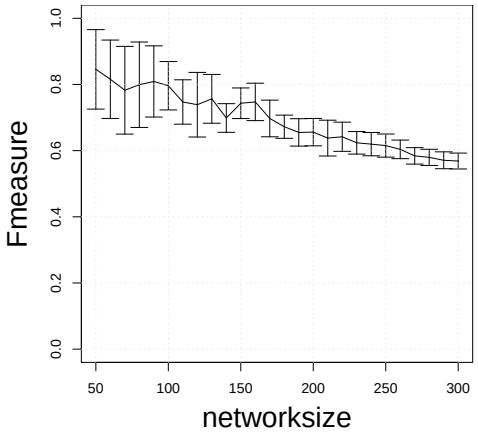


(a) F-measure - SAMBA

(b) Precision - SAMBA

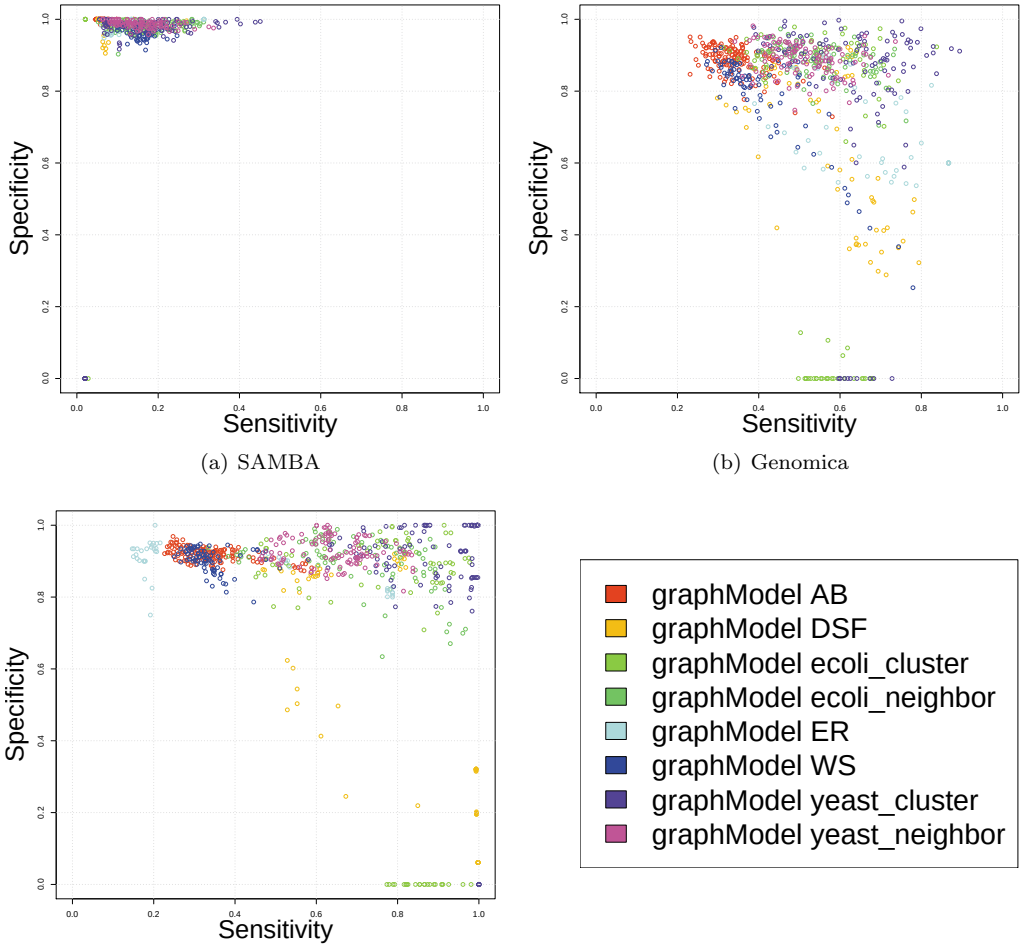


(c) F-measure - Genomica



(d) F-measure - Aracne

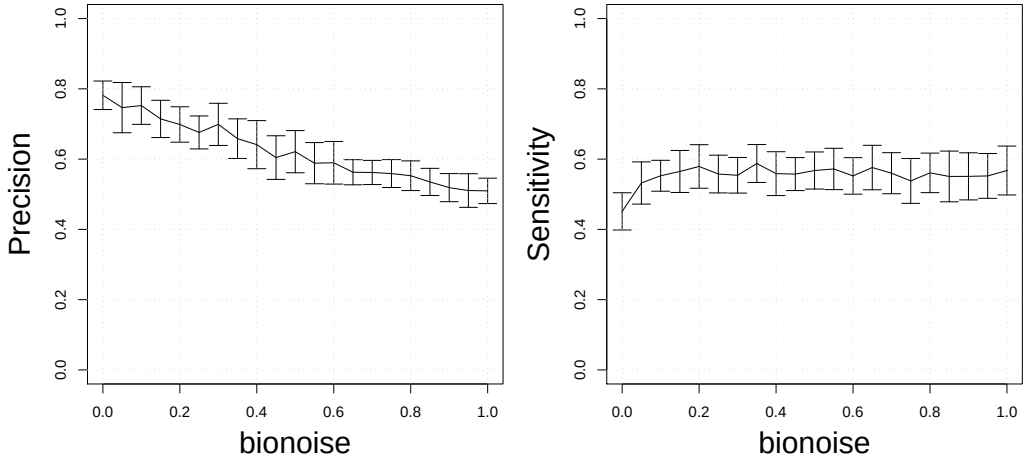
Figure 3: Impact of network size on the performance (F-measure and precision) of Aracne, Genomica and SAMBA.



(a) SAMBA (b) Genomica

(c) Aracne (d) Legend

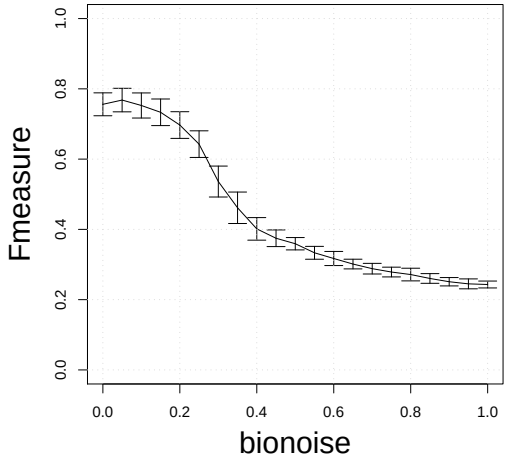
Figure 4: Impact of network topology on Sensitivity and Specificity for Aracne, Genomica and SAMBA.



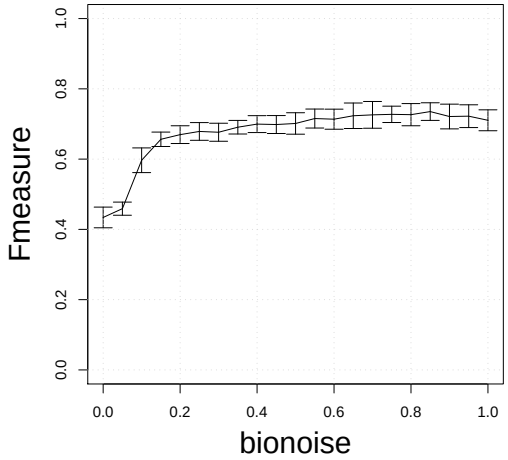
(a) Impact of biological noise on the precision for Genomica.

(b) Impact of biological noise on the sensitivity for Genomica.

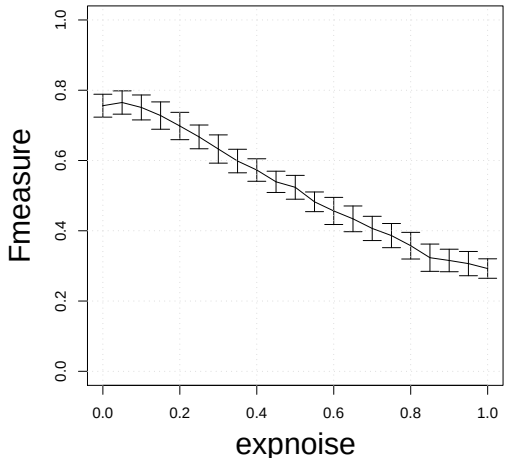
Figure 5: Impact of different noise types on performance measures for Genomica.



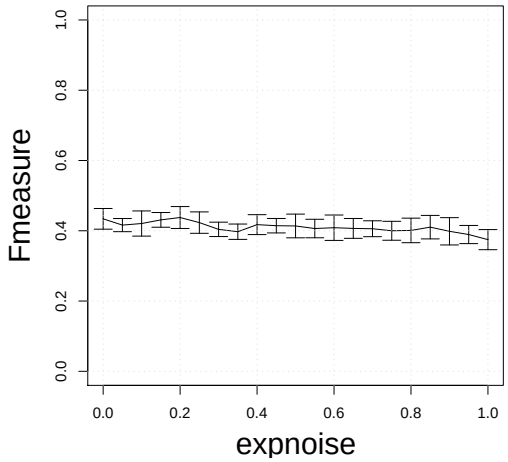
(a) Impact of biological noise on the F-measure for Aracne using default scoring regime (See Section 5).



(b) Impact of biological noise on the F-measure for Aracne using second scoring regime (See Section 5.)

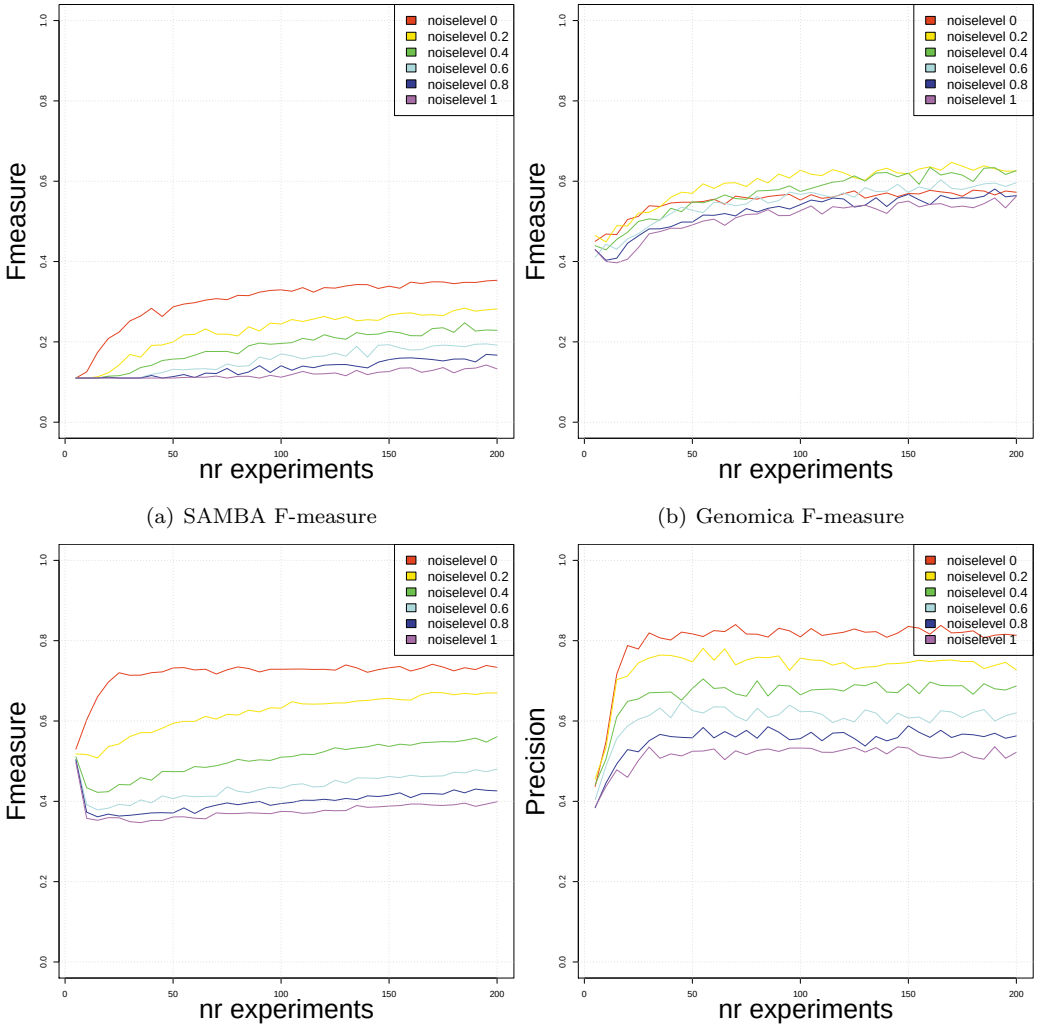


(c) Impact of experimental noise on the F-measure for Aracne using default scoring regime (See Section 5).



(d) Impact of experimental noise on the F-measure for Aracne using second scoring regime (See Section 5).

Figure 6: Impact of different noise types on performance measures for Aracne, Genomica and SAMBA.



(a) SAMBA F-measure

(b) Genomica F-measure

(c) Aracne F-measure

(d) Genomica precision

Figure 7: Impact of amount of available expression data on performance measures for Aracne, Genomica and SAMBA.