Fast Parallel Bayesian Networks Reconstruction with BNFinder

Alina Frolova[1] and Bartek Wilczynski[2]

[1] Institute of Molecular Biology and Genetics,
Zabolotnogo 150, 03680 Kyiv, Ukraine
a.o.frolova@imbg.org.ua

[2] Institute of Informatics, University of Warsaw,
Banacha 2, 02-089 Warsaw, Poland
bartek@mimuw.edu.pl

Abstract. Bayesian networks are probabilistic graphical models widely used to infer interactions between biological entities such as genes or proteins. In general, learning Bayesian networks from experimental data is NP-hard, leading to widespread use of heuristic search methods giving suboptimal results. However, in a number of important special cases, it is possible to find the optimal network in polynomial time. While our method makes it possible to reconstruct optimal networks in polynomial time, in cases where there is large amount of experimental data the running times can rise up to days of computations on a single CPU. In this work we present a new and improved version of BNFinder - our tool for learning optimal Bayesian networks. The improvement consist of parallelized inference algorithm providing significant speedup with good efficiency. In this work we outline the parallel algorithm and show its performance measured on simulated datasets as well as real biological data regarding phosphorylation network inference.

Keywords: Bayesian networks reconstruction, parallel computing, python multiprocessing.

1 Introduction

Bayesian networks (BNs) are a graphical representation of a multivariate joint probability distribution that exploits the dependency structure of distributions to provide a compact and natural repressentation of them. A BN is a directed acyclic graph, in which the nodes correspond to the variables and the edges correspond to direct probabilistic dependencies between them [1]. In general, inferring BN is NP-hard [2], however it was showed by Dojer [3] that it is possible to find optimal network in polynomial time when datasets are finite and there are external constraints ensuring network acyclicity. This algorithm was implemented in BNFinder - a tool for BNs reconstruction from experimental data [4].

One of the common use of BNs in bioinformatics is inference of interactions between genes [5] and proteins [6]. However, flexibility of BNFinder allowed us to move further from original concept of inferring regulatory networks from expression data. BNFinder was successfully applied to linking expression data with sequence motif information [7], identifying histone modifications connected to enhancer activity [8] and to predicting gene expression profiles of tissue-specific genes [9]. However, the widespread adoption of the algorithm is limited by its long running times bound by the time it takes to find the optimal set of parents for the most complex variable. Since the algorithm, as published by Dojer [3], was amenable to parallelization , we have developed a new version that is able to take advantage of multiple cores via the multiprocessing python module.

2 Implementation

The BNFinder algorithm is based on the following scheme: for each of the random variables find the best possible set of parent variables by considering them in a carefully chosen order of increasing cost function. Current version of BNFinder [10] includes a simple parallelization based on distributing the work done on each variable to a different process. However, this approach has natural limitations. Firstly, the number of parallelized tasks is bound by the number of random variables in the problem, meaning that in cases where only a few variables are considered (e.g. in classification by BNs) we get a very limited performance boost. Secondly, this kind of parallelization is sensitive (in terms of performance) to highly heterogeneous variables in the input data. If we consider an example where the true optimal network has a few nodes with multiple parents and majority of nodes with few parents, the potential gain in algorithm performance is not greater than in the case where all the nodes are of the most difficult category. This is often the case in biological networks given their scale-free network topology consisting of a few hub nodes with many parents and a large number of nodes that have one or small number of connections [11].

The alternative approach to parallelization of the BNfinder algorithm is to process variables sequentially, but consider different possible parent sets in parallel taking advantage of all available cores at the same time. This approach leads to a slightly more complex algorithm, however it yields superior results in terms of speedup and efficiency in virtually all realistic scenarios.

As the first approach to parallelization (each variable on a separate core) can theoretically outperform the second approach (different parent sets on different cores) due to slightly lower synchronization overhead, we have also implemented a hybrid approach which first parallelizes the variables into different cores and then subsequently the parent sets for each variable.

3 Results

In this work we compared two different implementations: hybrid algorithm with 2 layers of parallelization (between random variables and parents sets) and the second approach (simple algorithm) distributing only the parents set scoring and considering the variables sequentially. The original implementation serves as a baseline for computing the speedup and efficiency of the parallelization. Note, that hybrid algorithm behaves exactly the same as original BNFinder when the number of cores is less or equals the number of random variables. To compare their performance we used synthetic benchmark data as well as real datasets concerning protein phosphorylation network published by Sachs et al [12]. The algorithms performance on synthetic data (20 genes network) were almost identical, whether speedup of hybrid algorithm was better - 34x versus 29x (Fig. 1). Efficiency comparison showed that hybrid algorithm has unstable behaviour, performing better when number of cores correlates with number of genes (Fig. 1).

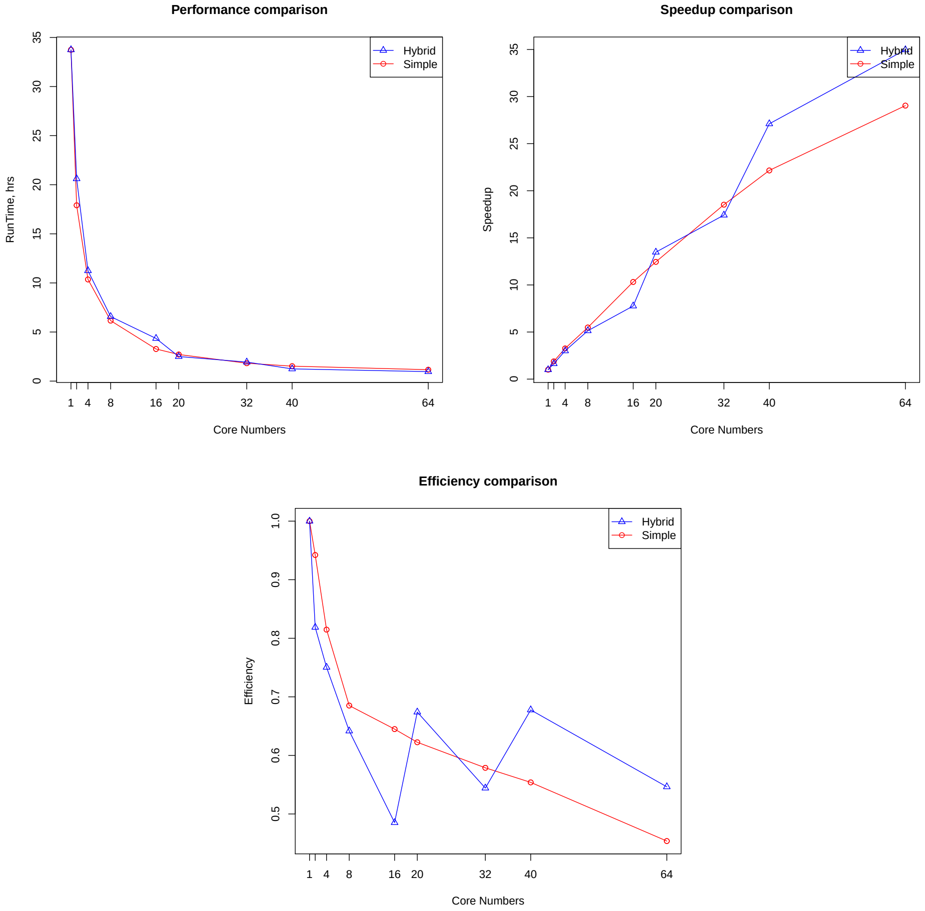
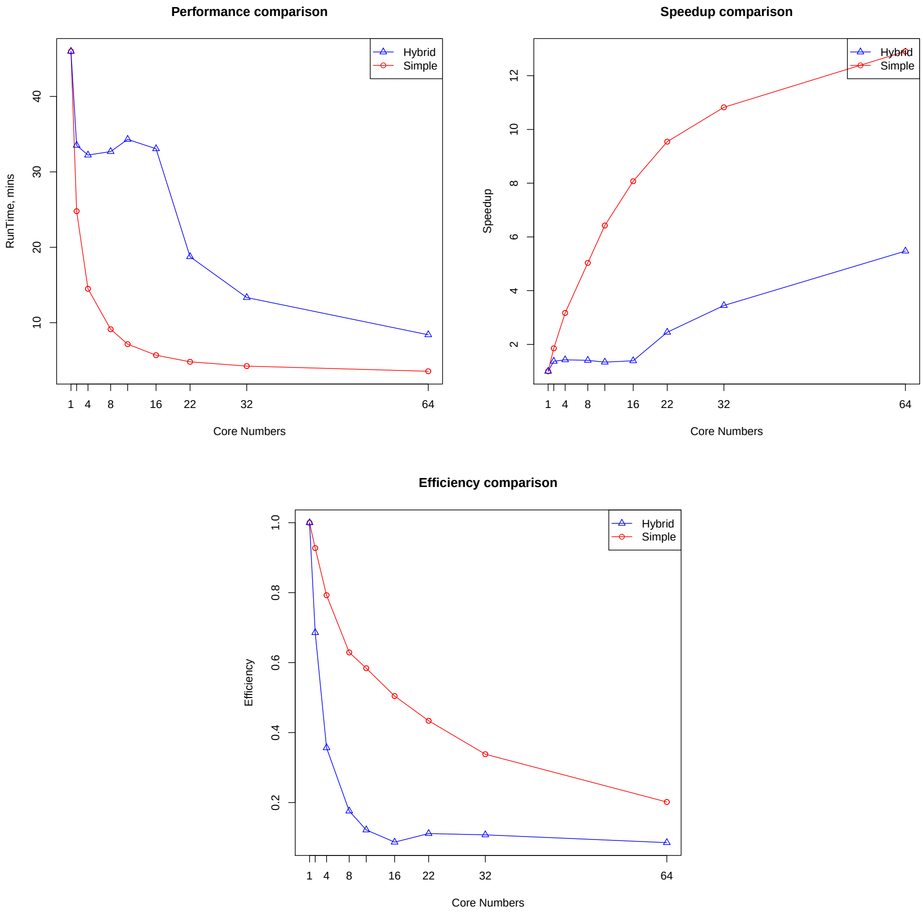


Fig. 1. Hybrid and simple algorithms comparison on synthetic data

When we took a real experimental dataset, it revealed more significant difference between two algorithms. Clearly, the simpler algorithm outperformes the hybrid one: with the efficiency of 0.5 it showed 8x speedup on Phosphorylation network data (11 genes network) while the alternative algorithm showed only 1.5x speedup (Fig. 2). Importantly, the sets of parents in this dataset are quite diverse with respect to the number of parents needed for accurate representation of its conditional probability distribution.


Performance comparison

Speedup comparison

Efficiency comparison

Fig. 2. Hybrid and simple algorithms comparison on biological data

The reason for such drastic difference was stated above - sensitivity of the hybrid algorithm to highly heterogeneous variables in the input data, which in case of Sachs data - one gene with 6 parents and other genes with 1-2 parents. Importantly, the better performing algorithm is also the one showing more consistent behaviour. As we can see in Fig 2, the simple algorithm's results correlate well with Amdahl's Law in terms of performance and efficiency with approximately 10% of the algorithm being strictly serial [13].

All tests were performed on the same server with AMD Opteron(TM) Processor 6272 (4 CPUs with total of 64 cores) and 512GB RAM. During the tests server was loaded only by regular system processes, but to ensure statistical significance we performed each test several times, so Fig. 1 and Fig. 2 represent average results.

The latest source code is available from the following repository - `https://code.launchpad.net/~fshodan/bnfinder/trunk`.

4 Conlusions

In summary, the new version of BNFinder constitutes a major improvement over original implementation for users who want to use the power of multiprocessor setups. As the new version of BNFinder is highly parallelized, it can reach more than 30x faster running times on large datasets provided a sufficiently large computer. Given the growing popularity of multi-core personal computers, we think that it will be useful to majority of BNfinder users. Importantly, the improved implementation is quite insensitive to heterogenous datasets, i.e. situations where complex variables with multiple parents are mixed with simple variables. The results in our computational experiments show high correlation of running times with those predicted by Amdahl's Law and indicating the fraction of non-parallelizable code to be on the order of 10 per cent of the total computational time. These features taken together make the new implementation much more suitable for infering network topologies from biological data, as they tend to contain many variables with heterogenous parent set sizes.

Acknowledgments. This work was partially supported by the National Center for Science grant decision numberDEC-2012/05/B/N22/0567 and Foundation for Polish Science within the SKILLS programme.

References

1. Friedman, N., Koller, D.: Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks. Mach. Learn. 50, 95–125 (2003)
2. Chickering, D., Heckerman, D., Meek, C.: Large-sample learning of Bayesian networks is NP-hard. J. Mach. Learn. Res. 5, 1287–1330 (2004)
3. Dojer, N.: Learning Bayesian networks does not have to be NP-hard. Math. Found. Comput. Sci. 305–314 (2006)
4. Wilczynski, B., Dojer, N.: BNFinder: exact and efficient method for learning Bayesian networks. Bioinformatics. 25 (2), 286–287 (2009)
5. Zou, M., Conzen, S.D.: A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. Bioinformatics. 21, 71–9 (2005).

6. Jansen, R., Yu, H., Greenbaum, D., Kluger, Y., Krogan, N.J., Chung, S., Emili, A., Snyder, M., Greenblatt, J.F., Gerstein, M.: A Bayesian networks approach for predicting protein-protein interactions from genomic data. Science. 302, 449–453 (2003).
7. Dabrowski, M., Dojer, N., Zawadzka, M., Mieczkowski, J., Kaminska, B.: Comparative analysis of cis-regulation following stroke and seizures in subspaces of conserved eigensystems. BMC Syst. Biol. 4 (1), 86 (2010).
8. Bonn, S., Zinzen, R.P., Girardot, C., Gustafson, E.H., Perez-Gonzalez, A., Delhomme, N., Ghavi-Helm, Y., Wilczyski, B., Riddell, A., Furlong, E.E.M.: Tissue-specific analysis of chromatin state identifies temporal signatures of enhancer activity during embryonic development. Nat. Genet. 44, 148–156 (2012).
9. Wilczynski, B., Liu, Y.-H., Yeo, Z.X., Furlong, E.E.M.: Predicting spatial and temporal gene expression using an integrative model of transcription factor occupancy and chromatin state. PLoS Comput. Biol. 8, e1002798 (2012).
10. Dojer, N., Bednarz, P., Podsiadlo, A., Wilczynski, B.: BNFinder2: Faster Bayesian network learning and Bayesian classification. Bioinformatics. 29 (16), 2068–2070 (2013)
11. Barabsi, A.-L., Oltvai, Z.N.: Network biology: understanding the cells functional organization. Nat. Rev. Genet. 5, 101–13 (2004).
12. Sachs, K., Perez, O., Peer, D., Lauffenburger, D.A., Nolan, G.P.: Causal protein-signaling networks derived from multiparameter single-cell data. Science. 308, 523–529 (2005).
13. McCool, M., Reinders, J., Robison, A.: Structured Parallel Programming: Patterns for Efficient Computation. Morgan Kaufmann, Waltham (2012).