Bioinformatics Advance Access published May 29, 2015

Applications Note

lpNet: a linear programming approach to reconstruct signal transduction networks

Marta R. A. Matos[1,†, ††], Bettina Knapp[1,2, ††,*] and Lars Kaderali[1]

[1]Institute for Medical Informatics and Biometry, Medical Faculty Carl Gustav Carus, Technische Universität Dresden, Fetscherstr. 74, 01307 Dresden, Germany

[2]Institute of Computational Biology, Helmholtz Zentrum München, Ingolstädter Landstr. 1, 85764 Neuherberg, Germany

†Current affiliation: The Novo Nordisk Foundation Center for Biosustainability, Technical University of Denmark.

†† These authors contributed equally

Associate Editor: Dr. Igor Jurisica

ABSTRACT

Summary: With the widespread availability of high-throughput experimental technologies it has become possible to study hundreds to thousands of cellular factors simultaneously, such as coding- or non-coding mRNA or protein concentrations. Still, extracting information about the underlying regulatory or signaling interactions from these data remains a difficult challenge. We present a flexible approach towards network inference based on linear programming. Our method reconstructs the interactions of factors from a combination of perturbation/non-perturbation and steady-state/time-series data. We show both on simulated and real data that our methods are able to reconstruct the underlying networks fast and efficiently, thus shedding new light on biological processes and, in particular, into disease's mechanisms of action. We have implemented the approach as an R package available through bioconductor.

Availability and Implementation: This R package is freely available under the Gnu Public License (GPL-3) from bioconductor.org (http://bioconductor.org/packages/release/bioc/html/lpNet.html) and is compatible with most operating systems (Windows, Linux, Mac OS) and hardware architectures.

Contact: bettina.knapp@helmholtz-muenchen.de

INTRODUCTION

Using network inference approaches it is possible to understand how different cellular components (e.g. genes, proteins, metabolites) interact with each other. Several methods for network inference exist, such as Boolean networks (Bock, et al., 2012; Haider and Pal, 2012), (Dynamic) Bayesian networks (BN) (Friedman, et al., 2000; Sachs, et al., 2005), and methods based on differential equations (Gardner, et al., 2003; Kimura, et al., 2012). Boolean networks scale up well for larger networks and are easy to interpret. Yet, the biological signal is binarized which leads to a substantial loss of information. BN and differential equations allow for a more detailed modeling of the underlying processes, but they are often applicable only to small-scale problems as they usually scale-up poorly.

One of the major recent approaches to model signaling networks based on perturbation data are the Nested Effects Models (NEMs) (Markowetz, et al., 2007) which have been widely applied and extended in different ways. This approach assumes a small number of candidate pathway genes which are silenced (S-genes), and the effects of this silencing are measured on a large set of "effects" genes (E-genes) – thus using indirect observations of effects for network inference. Approaches that use direct observations from perturbed networks obtained either at a single or several time-points include (Dynamic) Deterministic Effects Propagation Networks ((D)DEPNs) (Bender, et al., 2010; Frohlich, et al., 2009), Dynamic Probabilistic Boolean Threshold Networks (D-PTBNs) (Kiani and Kaderali, 2014), and Sorad (Äijö et al, 2014).

However, many of the published network inference approaches are computationally expensive and thus not suitable for large networks. Also, although many methods are publicly available, a substantial fraction of them are dependent on 3rd-party proprietary software (e.g. Matlab) and are not properly documented, thus making them very hard to use.

Here, we present a flexible approach implemented as a bioconductor package, lpNet, freely available for any of the major operating systems. The method implemented in lpNet extends a previously developed method in (Knapp and Kaderali, 2013) by adding support for time-series data. As a result, the method is now suited for any combination of perturbation / non-perturbation and steady-state / time-series data, and not only for perturbation steady-state data, as in the previous version of 2013. Furthermore, the calculation of a node's activity, which directly influences the resulting network, was improved, leading to more accurate results. Finally, since the method is formulated as a linear programming approach, inferring networks becomes fast and efficient even for large-scale problems.

*To whom correspondence should be addressed.

METHODS

The signaling network to be inferred is modeled by a graph $G = (V, W)$, were $v_i \in V, i \in \{1, n\}$ are nodes that can represent, e.g. genes or proteins, and $w_{ij} \in W$, $w_{ij} \in \mathbb{R}$ are directed edges from node v_i to node v_j. If $w_{ij} > 0$ node v_i activates node v_j, if $w_{ij} < 0$ node v_i inhibits node v_j, and if $w_{ij} = 0$ nodes v_i and v_j are not connected. For the inference, we define an observation matrix $X \in \mathbb{R}_0^+$ which can be 2- or 3-dimensional, with dimensions representing nodes v_i, the perturbation experiments $k \in \{1, K\}$, and optionally the time points $t \in \{1, T\}$ in the 3rd dimension. We have previously described the use of linear programming for network inference from steady state data (Knapp and Kaderali, 2013). lpNet now implements an extension that enables the use of time-series data and thus uses a 3D observation matrix.

Perturbation experiments are encoded in an activation matrix $B^{n \times K} \in \{0,1\}$, where $b_{ik} = 0$ means node v_i in perturbation experiment k is silenced, otherwise it is not silenced. This matrix can also be used to encode experiments with different stimuli, by considering each stimulus to be a different perturbation experiment.

We assume the signal to propagate through the network as an information flow, starting in the source nodes and ending in the sink nodes. The signal propagation is interrupted when either a silenced node or a node inhibited by its parent nodes is reached. Furthermore, we assume the signal to propagate from parent nodes to child nodes one step per time point. The *activity* of a node i is then given by:

$$w_i^0 + \sum_{j \neq i} w_{ji} x_{jkt-1} \qquad (1)$$

where $w_i^0 \in \mathbb{R}$ is the baseline activity of node v_i and $x_{jkt-1} \in \mathbb{R}_0^+$ is the observation value for node v_j, in perturbation experiment k, at time point $t - 1$. A node is said to be active if $x_{ikt} \geq \delta_i$, otherwise it is inactive. $\delta_i \in \mathbb{R}^+$ is a user-defined threshold, and is key for the method's performance. Only active nodes at time point $t - 1$ can influence other nodes at t. If a given node v_j is considered as inactive at $t - 1$ with $x_{jkt-1} < \delta_i$, or if it was silenced, $b_{ik} = 0$, then x_{jkt-1} is set to 0 in eq. (1), i.e. node v_j does not contribute to node v_j's activity.

Given all the stated assumptions, the network $G = (V, W)$ is inferred by solving a linear programming problem as described in the supplementary information, section 1, using the simplex method, whose complexity has been shown to be polynomial in practice.

RESULTS

The performance of the first version of lpNet, suitable only for perturbation steady-state data, has been assessed (Knapp and Kaderali, 2013). As for the latest version, with support for a combination of perturbation / non-perturbation and steady-state / time-series data, it has been used in the DREAM 8 competition, in the HPN-DREAM breast cancer network inference challenge, where it scored 3rd for the *in silico* challenge (AUROC: 0.68) and 29th for the experimental challenge (AUROC: 0.57) without using prior information, among more than 60 competing groups. In the *in silico* challenge the goal was to infer the causal edges in a 20 node network given a dataset containing the 20 nodes observations across 10 time points and 4 perturbation experiments (one of these being the control). In the experimental challenge the goal was to infer 32 causal networks, one for each combination of cell line + stimulus – there were 4 cell lines and 8 different stimuli. Each of the 32 datasets contained ~45 nodes observations across 7 time points and 4 inhibition experiments (one of these being the control). In the supplementary information, section 3, we briefly compare lpNet with DDEPN (Bender, 2013), a method also implemented as a well documented R package and suitable for the same type of data. Two conclusions from this comparison are worth emphasizing: i) lpNet is very robust against noise; ii) on the same platform, lpNet takes on average 15min to infer a network with 10 nodes, 10 time points, and 2 perturbations, while DDEPN takes, on average, 101 min (computations done on an Intel Xeon X5460 @ 3GHz, 2×6MB L2 cache, 32GB RAM, 64bit Linux OS). More detailed results are presented in the supplementary materials and in (Matos, 2013).

DISCUSSION AND CONCLUSION

The lpNet package supports now any combination of perturbation / non-perturbation and steady-state / time-series data, and not only perturbation steady-state data as in the 2013 version. Moreover, by formulating network inference as a linear programming problem and using the simplex method to solve it, lpNet is computationally very efficient and runs, on average, 6 to 7 times faster than the DDEPN approach. Results are robust against noise, and due to the fast running time, cross-validation can be used to fit model parameters such as δ.

Funding: This work was supported by the German Ministry of Education and Research (BMBF) via GerontoSys/AgeNet (0315898), the European Union seventh framework program via SysPatho (grant number 260429), and the German Research Foundation (SPP1395/InKoMBio Busch 900/6-1). MM also acknowledges the Erasmus scholarship 2012-1-PT1-ERA02-12586 (11.3) 217/201.

Conflict of Interest: none declared.

REFERENCES

Äijö, T. *et al.*, Sorad: a systems biology approach to predict and modulate dynamic signaling pathway response from phosphoproteome time-course measurements. *Bioinformatics*, 2013.

Bender, C. ddepn: Dynamic Deterministic Effects Propagation Networks: Infer signalling networks for timecourse RPPA data. R package version 2.2, 2013.

Bender, C., *et al.* Dynamic deterministic effects propagation networks: learning signalling pathways from longitudinal protein array data. *Bioinformatics* 2010;26(18):i596-602.

Bock, M., *et al.* Hub-centered gene network reconstruction using automatic relevance determination. *PloS one* 2012;7(5):e35077.

Friedman, N., *et al.* Using Bayesian networks to analyze expression data. *Journal of computational biology : a journal of computational molecular cell biology* 2000;7(3-4):601-620.

Frohlich, H., *et al.* Deterministic Effects Propagation Networks for reconstructing protein signaling networks from multiple interventions. *BMC bioinformatics* 2009;10:322.

Gardner, T.S., *et al.* Inferring genetic networks and identifying compound mode of action via expression profiling. *Science* 2003;301(5629):102-105.

Haider, S. and Pal, R. Boolean network inference from time series data incorporating prior biological knowledge. *BMC genomics* 2012;13 Suppl 6:S9.

Kiani N. and Kaderali L., Dynamic probabilistic threshold networks to infer signaling pathways from time-course perturbation data. *BMC bioinformatics*,. 2014.

Kimura, S., *et al.* Inference of S-system models of genetic networks by solving one-dimensional function optimization problems. *Mathematical biosciences* 2012;235(2):161-170.

Knapp, B. and Kaderali, L. Reconstruction of cellular signal transduction networks using perturbation assays and linear programming. *PloS one* 2013;8(7):e69220.

Markowetz, F., *et al.* Nested effects models for high-dimensional phenotyping screens. *Bioinformatics* 2007;23(13):i305-312.

Sachs, K., *et al.* Causal protein-signaling networks derived from multiparameter single-cell data. *Science* 2005;308(5721):523-529.