BIOINFORMATICS APPLICATIONS NOTE
Vol. 27 no. 7 2011, pages 1026–1027
doi:10.1093/bioinformatics/btr057

Gene expression
Advance Access publication February 7, 2011

tigre: Transcription factor inference through gaussian process reconstruction of expression for bioconductor

Antti Honkela[1,2,*], Pei Gao[3], Jonatan Ropponen[2], Magnus Rattray[4,*] and Neil D. Lawrence[4,*]

[1]Helsinki Institute for Information Technology HIIT, University of Helsinki, Helsinki, [2]Department of Information and Computer Science, Aalto University, Helsinki, Finland, [3]Department of Public Health and Primary Care, University of Cambridge, Cambridge and [4]Sheffield Institute for Translational Neuroscience and Department of Computer Science, University of Sheffield, Sheffield, UK

Associate Editor: Trey Ideker

ABSTRACT

Summary: *tigre* is an R/Bioconductor package for inference of transcription factor activity and ranking candidate target genes from gene expression time series. The underlying methodology is based on Gaussian process inference on a differential equation model that allows the use of short, unevenly sampled, time series. The method has been designed with efficient parallel implementation in mind, and the package supports parallel operation even without additional software.

Availability: The *tigre* package is included in Bioconductor since release 2.6 for R 2.11. The package and a user's guide are available at http://www.bioconductor.org.

Contact: antti.honkela@hiit.fi; m.rattray@sheffield.ac.uk; n.lawrence@dcs.shef.ac.uk

Received on November 9, 2010; revised on January 24, 2011; accepted on January 28, 2011

1 INTRODUCTION

Understanding genome function through reverse engineering of gene regulatory relationships from experimental data is one of the key challenges in current biology (Bansal *et al.*, 2007; Bickel *et al.*, 2009). One popular technique is to use gene expression time series to infer these relationships. Unfortunately, most real-world expression time series are short (Ernst *et al.*, 2005) and contain insufficient information for any realistic reconstruction of the gene regulatory network (Smet and Marchal, 2010).

Recognizing this, the *tigre* package aims to answer a much simpler question: given time series expression data where a transcription factor (TF) is changing its activity, which genes are plausibly regulated by that TF? As a result, it provides a ranking of tested target genes according to their likelihood of being targets of the TF.

The underlying methodology was presented by Honkela *et al.* (2010a), who showed that it can yield remarkably accurate predictions from very limited data, often attaining more accurate results based on the simple wild-type time series expression data than what could be obtained using TF knockout data. The method is based on a linear ordinary differential equation model of TF protein translation and transcriptional regulation. Placing a non-parametric Gaussian process prior on unobserved mRNA concentration or protein activity time courses leads to a joint Gaussian process model over the protein activity level and all gene expression levels (Gao *et al.*, 2008). The Gaussian processes can be marginalized out analytically while the model parameters are optimized using maximum likelihood. Candidate targets are then ranked by model likelihood, which effectively measures how well the target fits a model of regulation by the inferred TF.

The method was originally implemented in MATLAB, making handling of genomic datasets much more cumbersome than in the new Bioconductor implementation, which benefits from all the pre-processing, annotation and other tools in Bioconductor (Gentleman *et al.*, 2004).

*To whom correspondence should be addressed.

2 IMPLEMENTATION

The *tigre* package is tightly integrated into the Bioconductor microarray data analysis framework, especially with the *puma* package (Pearson *et al.*, 2009), which associates expression levels derived from high-density microarray data with Bayesian confidence regions analogous to standard errors. Expression data from other platforms and analysis methods that do not provide standard error estimates can also be easily used. In this case, the observation noise level is estimated as part of fitting the model.

Functions are provided for processing data to a format suitable for the method, including estimation of standard errors of unlogged expression values for *puma* processed data (currently *puma* standard errors are for logged expression); fitting the models individually or in a batch; and plotting the models to assess the fit. The models are fitted using scaled conjugate gradient optimization (Møller, 1993). Depending on the number of genes in the model, the model is defined using from seven up to a few dozen parameters, allowing very compact storage of fitted models.

2.1 Parallelization of the ranking

The method implemented by *tigre* includes no Monte Carlo simulations, but typical running times still range from seconds to up to a few minutes per gene depending on the data and the number of targets in the models. Therefore, *tigre* has been designed for efficient parallelization. In the ranking, each gene is handled completely independently. This makes the code trivially parallelizable up to the level of running each gene in a separate machine. This linear parallelization to potentially several thousands of processes is impossible in more tightly coupled modeling methods.

In terms of required additional software, the easiest way to run *tigre* in parallel is to simply split the task to a number of jobs that can be run independently, possibly by submitting them as independent jobs to a queuing system. The package does not include integration with a message passing interface (MPI) environment, because that matches its requirements poorly. A number of independent jobs should also be easier to schedule than a single large MPI job.

An alternative technique for running *tigre* in parallel is based on MapReduce (Dean and Ghemawat, 2008). The ranking approach fits this paradigm perfectly: the mapper fits models to each gene independently and the reducer forms the final ranking. We have implemented this approach using Hadoop and RHIPE. This approach provides an alternative method of running the ranking in a highly parallelized fashion in a cloud computing setting.

3 DISCUSSION

tigre can fit the models already on very short time series. We have successfully applied it to datasets with as few as 6 and 7 time points (Honkela *et al*., 2010a, b).

The output of *tigre* is a ranked set of models for each candidate target gene together with their associated log-likelihoods used to form the ranking. Direct interpretation of the log-likelihood scores can be difficult. We have found it very useful to use visualizations of the models to assess the fit and to select thresholds for filtering weakly expressed genes (for details, see Honkela *et al*., 2010a). We have also developed a web interface called *tigreBrowser* (http://pypi.python.org/pypi/tigreBrowser) to aid in browsing the visualizations.

The linearity of the differential equation transcription model greatly simplifies the algorithm, but it may be too crude an assumption for some situations. We are working on a method based on a more realistic non-linear model. Such models require, however, more advanced computational techniques that often tend to be more fragile. *tigre* avoids this pitfall by using a simple enough model allowing very robust inference while effectively capturing the essential degrees of freedom in the transcription regulatory process.

ACKNOWLEDGEMENT

The authors wish to thank Jennifer Withers for useful comments on the package and Miika-Petteri Matikainen for implementing *tigreBrowser*.

Funding: This work was supported by the Academy of Finland (121179 to A.H.); the Engineering and Physical Sciences Research Council (EP/F005687/1 to M.R. and N.D.L.); and the IST Programme of the European Community, under the PASCAL2 Network of Excellence (IST-2007-216886). This publication only reflects the authors' views.

Conflict of Interest: none declared.

REFERENCES

Bansal,M. *et al*. (2007) How to infer gene networks from expression profiles. *Mol. Syst. Biol.*, **3**, 78.

Bickel,P.J. *et al*. (2009) An overview of recent developments in genomics and associated statistical methods. *Philos. Transact. A Math. Phys. Eng. Sci.*, **367**, 4313–4337.

Dean,J. and Ghemawat,S. (2008) MapReduce: simplified data processing on large clusters. *Commun. ACM*, **51**, 107–113.

Ernst,J. *et al*. (2005) Clustering short time series gene expression data. *Bioinformatics*, **21**(Suppl. 1), i159–i168.

Gao,P. *et al*. (2008) Gaussian process modelling of latent chemical species: applications to inferring transcription factor activities. *Bioinformatics*, **24**, i70–i75.

Gentleman,R.C. *et al*. (2004) Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol.*, **5**, R80.

Honkela,A. *et al*. (2010a) Model-based method for transcription factor target identification with limited data. *Proc. Natl Acad. Sci. USA*, **107**, 7793–7798.

Honkela,A. *et al*. (2010b) Ranking of gene regulators through differential equations and Gaussian processes. In *Proceedings of 2010 IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2010)*, Kittilä, Finland, pp. 154–159.

Møller,M.F. (1993) A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, **6**, 525–533.

Pearson,R.D. *et al*. (2009) puma: a Bioconductor package for propagating uncertainty in microarray analysis. *BMC Bioinformatics*, **10**, 211.

Smet,R.D. and Marchal,K. (2010) Advantages and limitations of current network inference methods. *Nat. Rev. Microbiol.*, **8**, 717–729.