Do we need to claim original algorithms are in MATLAB and this is a big step forward to integrate with Bioconductor?

BIOINFORMATICS

Vol. 00 no. 00 2010
Pages 1–2

tigre: Transcription factor Inference through Gaussian process Reconstruction of Expression for Bioconductor

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

[1] Department of Information and Computer Science, Aalto University School of Science and Technology, Helsinki, Finland
[2] Department of of Public Health and Primary Care, University of Cambridge, Cambridge, UK
[3] Department of Computer Science, University of Sheffield, Sheffield, UK

Sheffield Institute for Translational Neuroscience and

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

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 using short unevenly sampled time series. The method has been designed with efficient parallel implementation in mind, and the package supports several alternative approaches for parallel operation on commodity hardware.

Availability: The package is included in Bioconductor release 2.6 for R 2.11.

Contact: antti.honkela@tkk.fi

1 INTRODUCTION

Understanding genome function through reverse engineering of gene regulatory relationships from experimental data is one of the key challenges in current biology (The ENCODE Project Consortium, 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).

Recognising this, the *tigre* package aims at answering a much simpler question: given time series expression data where a transcription factor (TF) is changing its activity, is a gene 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. (2010b), who showed that it can yield remarkably accurate predictions from very limited data, often attaining more accurate results based on the simple wild time series expression data than could be obtained using TF knock-out data.

2 METHODS

The *tigre* package is an implementation of the Gaussian process single input motif framework of Gao et al. (2008) and the related TF target ranking method of Honkela et al. (2010b). This framework is based on a linear ordinary differential equation model of TF protein translation and transcription regulation described by the equations

$$\frac{\mathrm{d}p(t)}{\mathrm{d}t} = f(t) - \delta p(t) \, , \tag{1}$$

$$\frac{\mathrm{d}m_j(t)}{\mathrm{d}t} = B_j + S_j p(t) - D_j m_j(t) \, , \tag{2}$$

where $p(t)$ is the TF protein and $m_j(t)$ is the jth target mRNA concentration at time t. The parameters B_j, S_j and D_j are the baseline transcription rate, sensitivity and decay rate respectively for the mRNA of the jth target (as described by Barenco et al. (2006)). The parameter δ is the decay rate of the TF protein (Honkela et al., 2010b).

Placing a Gaussian process prior on $f(t)$[1] leads to a joint Gaussian process over all continuous-time activity functions. The parameters of the model as well as other parameters of the Gaussian process covariance are optimised by maximising the marginal likelihood.

3 IMPLEMENTATION

The *tigre* package is tightly integrated into Bioconductor microarray data analysis framework, especially with the *puma* package (Pearson et al., 2009).

Functions are provided for processing data to a format suitable for the method, including estimation of variances of absolute expression values for *puma* processed data; fitting the models individually or in a batch; and plotting the models to assess the fit. The models are fitted using scaled conjugate gradient optimisation (Møller, 1993). Fitted models can be stored very compactly by just storing their parameters. They can also be easily recreated afterwards without rerunning the optimiser used in the fitting.

3.1 Parallelisation

The method implemented by *tigre* includes no Monte Carlo simulations and is thus relatively fast, but still takes some time,

do we only have this implemented? Would be cool to be more general.

*to whom correspondence should be addressed

[1] If the TF protein is under significant post-translational regulation, Eq. (1) may be omitted and the prior placed directly on $p(t)$. In this case multiple known targets are needed to reliably infer $p(t)$.

ranging from seconds to up to a few minutes per gene depending on the data and the number of targets in the models.

tigre has been designed for efficient parallelisation. In the ranking, each gene is handled completely independently. This makes the code trivially parallelisable up to the level of running each gene in a separate machine. This linear parallelisation to potentially several thousands of processes is impossible in more tightly coupled modelling methods.

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 an 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 also allows relatively easy running of the ranking in a highly parallelised fashion in a cloud computing setting.

4 DISCUSSION

tigre can provide useful results on very short time series. We have successfully applied it to data sets with as few as 6 and 7 time points (Honkela et al., 2010b,a).

The linearity of the differential equation transcription model greatly simplifies the algorithm, but it may be too crude assumption for some situations. We are working on an method based on a more realistic model. However, it ~~seems difficult~~ it will be challenging to turn that framework into something as convenient as *tigre*, which nicely captures the essential degrees of freedom in the transcription regulatory process.

ACKNOWLEDGEMENT

The authors wish to thank Jennifer Withers for useful comments on the package.

Funding: A.H. was supported by Postdoctoral Researcher's Project No 121179 of the Academy of Finland. M.R. and N.D.L acknowledge support from EPSRC Grant No EP/F005687/1 ”Gaussian Processes for Systems Identification with Applications in Systems Biology”. This work was supported in part by the IST Programme of the European Community, under the PASCAL2 Network of Excellence, IST-2007-216886. This publication only reflects the authors' views.

REFERENCES

M. Barenco, D. Tomescu, D. Brewer, R. Callard, J. Stark, and M. Hubank. Ranked prediction of p53 targets using hidden variable dynamic modeling. *Genome Biol*, 7 (3):R25, 2006. doi: 10.1186/gb-2006-7-3-r25. URL `http://dx.doi.org/10.1186/gb-2006-7-3-r25`.

P. J. Bickel, J. B. Brown, H. Huang, and Q. Li. An overview of recent developments in genomics and associated statistical methods. *Philos Transact A Math Phys Eng Sci*, 367(1906): 4313–4337, Nov 2009. doi: 10.1098/rsta.2009.0164. URL `http://dx.doi.org/10.1098/rsta.2009.0164`.

J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, 2008. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/1327452.1327492.

J. Ernst, G. J. Nau, and Z. Bar-Joseph. Clustering short time series gene expression data. *Bioinformatics*, 21 Suppl 1: i159–i168, Jun 2005. doi: 10.1093/bioinformatics/bti1022. URL `http://dx.doi.org/10.1093/bioinformatics/bti1022`.

P. Gao, A. Honkela, M. Rattray, and N. D. Lawrence. Gaussian process modelling of latent chemical species: applications to inferring transcription factor activities. *Bioinformatics*, 24(16): i70–i75, Aug 2008. doi: 10.1093/bioinformatics/btn278. URL `http://dx.doi.org/10.1093/bioinformatics/btn278`.

Gs

A. Honkela, M. Milo, M. Holley, M. Rattray, and N. D. Lawrence. Ranking of gene regulators through differential equations and gaussian processes. In *Proc. 2010 IEEE Int. Workshop on Machine Learning for Signal Processing (MLSP 2010)*, pages 154–159, Kittilä, Finland, 2010a.

A. Honkela, C. Girardot, E. H. Gustafson, Y.-H. Liu, E. E. M. Furlong, N. D. Lawrence, and M. Rattray. Model-based method for transcription factor target identification with limited data. *Proc Natl Acad Sci U S A*, 107(17):7793–7798, Apr 2010b. doi: 10.1073/pnas.0914285107. URL `http://dx.doi.org/10.1073/pnas.0914285107`.

M. F. Møller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6(4):525–533, 1993.

R. D. Pearson, X. Liu, G. Sanguinetti, M. Milo, N. D. Lawrence, and M. Rattray. puma: a Bioconductor package for propagating uncertainty in microarray analysis. *BMC Bioinformatics*, 10:211, 2009. doi: 10.1186/1471-2105-10-211. URL `http://dx.doi.org/10.1186/1471-2105-10-211`.

R. D. Smet and K. Marchal. Advantages and limitations of current network inference methods. *Nat Rev Microbiol*, 8 (10):717–729, Oct 2010. doi: 10.1038/nrmicro2419. URL `http://dx.doi.org/10.1038/nrmicro2419`.

The ENCODE Project Consortium. Identification and analysis of functional elements in 1% of the human genome by the encode pilot project. *Nature*, 447(7146): 799–816, Jun 2007. doi: 10.1038/nature05874. URL `http://dx.doi.org/10.1038/nature05874`.