Mining regulatory network connections by ranking transcription factor target genes using time series expression data

Antti Honkela, Magnus Rattray, and Neil D. Lawrence

August 4, 2011

Abstract

Reverse engineering the gene regulatory network is challenging because the amount of available data is very limited compared to the complexity of the underlying network. We present a technique addressing this problem through focussing on a more limited problem: inferring direct targets of a transcription factor from short expression time series. The method is based on combining Gaussian process priors and ordinary differential equation models allowing inference on limited potentially unevenly sampled data. The method is implemented as an R/Bioconductor package and it is demonstrated by ranking candidate targets of the p53 tumour suppressor.

1 Introduction

Understanding the function and regulation of all human genes is one of the most important challenges that needs to be solved to fully reap the benefits of the Human Genome Project and the genomic era.

There are several regulatory mechanisms affecting protein expression. We will focus on transcriptional regulation, which is mediated by transcription factors (TFs). They are proteins that bind the DNA to activate or repress the transcription of their target genes. The relationships between TFs and their target genes can be represented as a graph which is called the gene regulatory network. In reality, this network is context-sensitive with many connections being, for instance, tissue specific.

Inferring the gene regulatory network from data is a very challenging problem. Even with high-throughput measurement techniques providing data on a genome-wide scale, the amount of data is tiny compared to the potential complexity of the regulatory network. Taking a conservative estimate of 1500 human TFs [1] regulating some 22500 genes [2] yields an astronomical number of more than 10^{450} potential networks. Even with a more realistic assumption of at most 5 regulating TFs for each gene there are more than 10^{18} potential networks, or from another perspective more than $6.3 \cdot 10^{13}$ potential sets of regulators for each gene. Assuming the regulatory network can be captured by a differential equation model with as many parameters, twice as many experiments would be needed to identify all the parameters [3]. If a simplified model of regulation with fewer parameters (such as a linear differential equation) can be assumed, the number of experiments will drop accordingly [4], but naturally such a model cannot capture all possible modes of combinatorial regulation.

The difficulty of general network inference even for simpler organisms was recently demonstrated by the DREAM5 Network Inference challenge[1], where one of the tasks was to infer the regulatory network for yeast using practically all available expression data. As a result, the best-performing team in this subtask achieved an area under the ROC curve of 0.539, which is only marginally better than the result 0.5 corresponding to random guessing.

1.1 Probabilistic dynamical models of gene regulation

To avoid these difficulties, we will focus on the specific task of identifying the targets of a TF in a time series experiment where the TF activity is changing [5,6]. The method works based on expression data alone. The TF activity can be estimated using information from known target genes or using the TF mRNA, if the TF is assumed to be primarily under transcriptional control. Given an estimate of TF activity and a model of transcription based on this activity, we can rank candidate targets based on how well they fit the model of regulation by this TF.

[1] See `http://wiki.c2b2.columbia.edu/dream/results/DREAM5/?c=4_1`

As our model of gene transcription regulated by a TF and optionally TF protein translation we use the following linear ordinary differential equations (ODEs):

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

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

where $p(t)$ is the TF protein at time t, $m_j(t)$ is the jth target mRNA concentration and $f(t)$ is the TF mRNA. 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 in [7]. The parameter δ is the decay rate of the TF protein [6].

In order to infer the protein activity $p(t)$, we need a prior for it. As the functions $f(t), p(t)$ and $m_j(t)$ are deterministically linked by the ODEs, this can be accomplished by placing a Gaussian process prior on $f(t)$ [2]. This leads to a joint Gaussian process over the three continuous-time functions $f(t), p(t)$ and $m_j(t)$. Data are observations of the expression levels at arbitrary specific times (not necessarily evenly spaced) and we assume a Gaussian noise model: $\hat{m}_j(t_i) \sim N(m_j(t_i), \sigma^2_{i,m_j})$ and $\hat{f}(t_i) \sim N(f(t_i), \sigma^2_{i,f})$ with known (derived from *puma*, see below) or estimated gene-specific noise variance parameters. The parameters of the model as well as other parameters of the Gaussian process covariance are optimised by maximising the marginal likelihood.

1.2 Related approaches

Other approaches for ranking TF targets using similar data include the rHVDM package [8] which implements a non-probabilistic variant of the same transcription ODE model [7]. rHVDM does not support using the TF mRNA observations to infer TF activity. Another alternative that defines

[2] 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 $p(t)$.

the model based only on TF expression data without a possibility of using information from the targets to infer TF activity is TSNI [9].

In general, the main difficulty in linking putative target genes with their regulators using expression data are the different degradation rates of mRNA molecules of different genes. If the degradation rates are available and can be compensated for, estimation of the regulator activation profiles is greatly simplified [10]. Our method can easily use such information as well.

2 Materials

The presented target ranking approach is implemented in the *tigre* package which is available in Bioconductor [11] for R (since Bioconductor 2.6 for R-2.11). *tigre* can make effective use of error estimates for expression levels provided by preprocessing methods. For Affymetrix arrays these are most easily available from the Bioconductor package *puma* [12], for Illumina arrays the corresponding information is available from the Bioconductor package *lumi* [13].

3 Methods

3.1 Data collection and experimental design

The primary data used by the model are expression time series. They can be measured using any quantitative technique such as mRNA sequencing (RNA-seq) or microarrays.

The application of our ranking method requires time series data. Due to financial constraints, most biological time series are very short, which poses problems for data modelling. We have applied the ranking to data sets with as few as 6–7 time points [6, 14], although longer time series ($>$ 10 time points) are significantly more informative.

Given a fixed experimental budget, it seems in general preferable to have more experimental perturbations and more finely sampled time series rather than more replicates of the same measurements, except possibly for some

single key time points. The models can use continuity in the time series to compensate for the noise that is usually detected using the replicates.

3.2 Preprocessing

The expression data should be preprocessed using the best tools for that particular platform. The *tigre* ranking method can make use of error or variance estimates from preprocessing. At the time of writing, the first such tools for RNA-seq data are only beginning to emerge. For microarrays, for example the *puma* [12] Bioconductor package provides such estimates for Affymetrix GeneChips, while the *lumi* [13] package provides this for Illumina Bead Arrays.

In order to use the data with *tigre*, it must be imported using the function `processData` that normalises the output from *puma* and *lumi*, or `processRawData` that handles plain expression data matrices. These functions require information about the experimental setup of the data, including observation times of different samples and which replicate time series they belong to. An example of the preprocessing using the p53 activation data from [7] is presented below. More details are available in the vignettes and other documentation of the respective packages.

```
library(ArrayExpress)

## Get data from ArrayExpress
p53.affybatch <- ArrayExpress('E-MEXP-549')
## Sort the arrays in a sensible order
mynames <- rownames(pData(p53.affybatch))
I <- order(strtrim(mynames, 5),
           pData(p53.affybatch)['Factor.Value..time.'])
p53.affybatch <- p53.affybatch[,I]

## Run mmgMOS preprocessing
library(puma)
p53.exprReslt <- mmgmos(p53.affybatch)

## Run tigre data normalisation
exps <- rep(1:3, each=7)      # replicates: 1...12...23...3
```

```
p53.timeseries <- processData(p53.exprReslt, experiments=exps)
```

3.3 Ranking

Ranking candidate targets of a TF requires inferring the TF activity profile. This can be done using TF mRNA expression levels if the TF can be assumed to be under transcriptional control, or using expression data of known targets, or combining both. Both of these cases are handled by the *tigre* function `GPRankTargets`.

If known targets are available, the function first fits a model using these known targets to infer the TF activity. All other genes are then screened by fitting additional models by adding them to the target set one at a time. If there are no known targets, all genes are screened by simply fitting them one at a time. In our p53 example this is achieved using the code below.

```
## Known target probe-sets from Barenco et al. (2006)
## These are the most informative probes for genes
## 'DDB2', 'CDKN1A', 'PA26', 'BIK', 'TNFRSF10B'
knownprobes <- c('203409_at', '202284_s_at', '218346_s_at',
                 '205780_at', '209295_at')

## Run the ranking.  This may take several hours to run.
## Results are saved to file 'p53ranking.RData'.
scores <- GPRankTargets(p53.timeseries, knownTargets=knownprobes,
                        scoreSaveFile='p53ranking.RData')

## Sort the list according to likelihood
scores <- sort(scores, descending=TRUE)

## Find 10 top-ranking probes
genes(scores)[1:10]

## Find the corresponding gene symbols
library(annotate)
mget(unlist(genes(s)[1:10]),
     getAnnMap('SYMBOL', annotation(p53.timeseries)))
```

3.4 Visualisation and analysis

The most effective way to explore the ranking results is by looking at visualisations of the models. Given that more than 10000 models are produced in the above analysis, this can be a daunting task without proper tools. A very useful tool for browsing the models is the *tigreBrowser* tool[3] The scores and the corresponding model visualisations can be exported for *tigreBrowser* as below.

```
## Export the scores and model visualisations to tigreBrowser
export.scores(scores, datasetName='Barenco2006',
              experimentSet='GPSIM 5 known',
              database='p53_results.sqlite',
              preprocData=p53.timeseries)
```

The results can then be viewed most easily by running the `tigreServer.py` script and pointing it to the saved `p53_results.sqlite` file. A screen shot of the browser is shown in Figure 1.

As there is no suitable genome-wide validation data available, we used a sequence-based method of validation. To do this, we looked for strong occurrences of the known p53 binding motif in the promoters of highly ranked targets. This was done by using the `p53scan` tool of Smeenk et al. [15] on the sequences of 5000 bp upstream of annotated transcription starts of RefSeq genes with annotated 5' UTRs downloaded from hg19 assembly in the UCSC Genome Browser [16]. Figure 2 shows the enrichment of the strongest 10% of the binding motif instances identified by `p53scan` as ranked by the score on the promoters of highly ranked genes. While these high scoring motif instances are the most likely true binding sites, some of them may still be non-functional. On the other hand, it is likely that many other genes are regulated through a weaker site on the promoter or a more distant site which cannot be detected without additional information.

[3] tigreBrowser is available for download at `http://users.ics.tkk.fi/ahonkela/tigre/`.

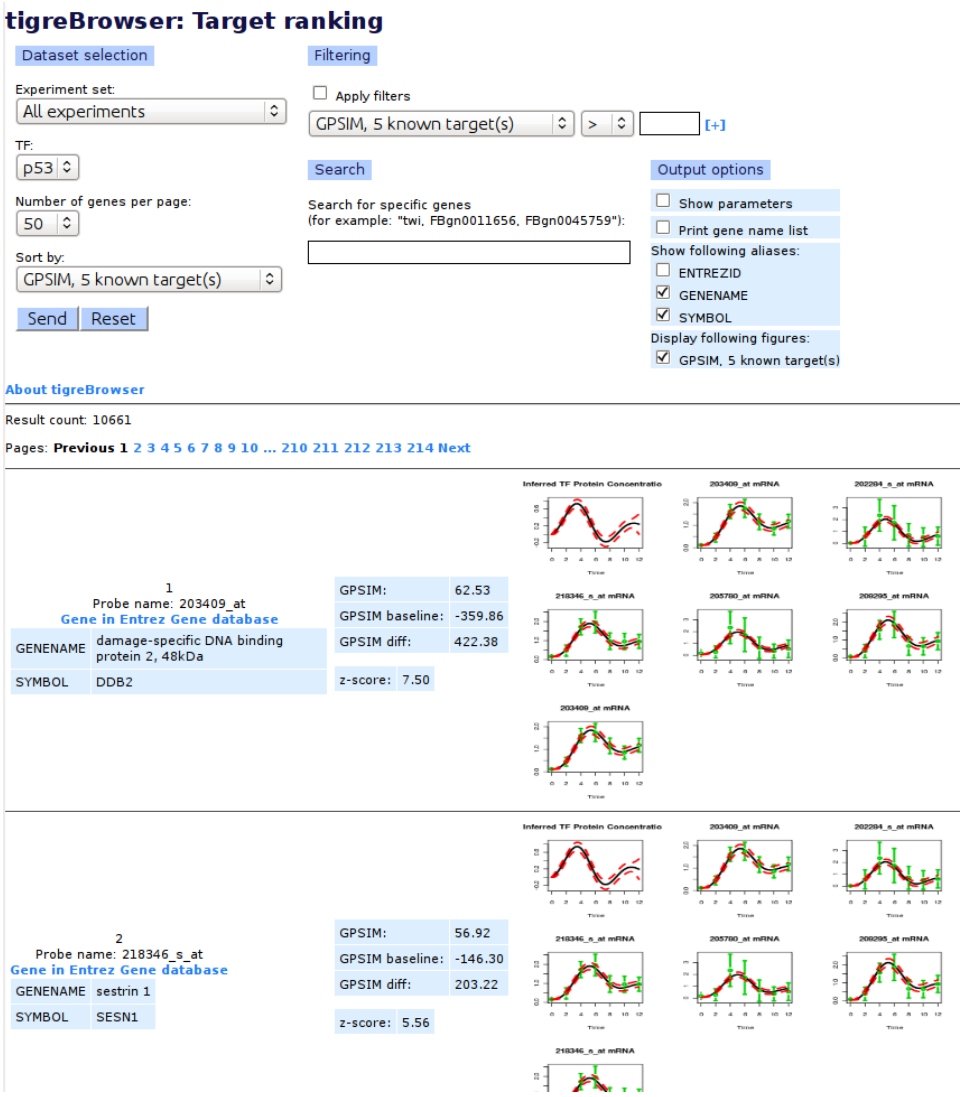

Figure 1: Screen shot of the tigreBrowser showing the top results of the p53 experiment.

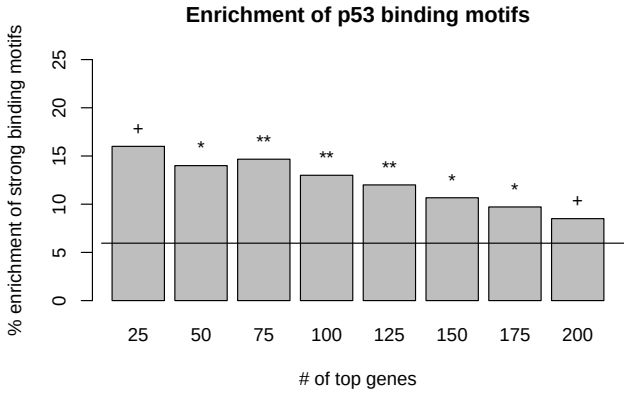

Figure 2: Enrichment of strong p53 binding motifs in the promoters of selected number of top predicted targets. p-values of the enrichments are denoted by '***': $p < 0.001$, '**': $p < 0.01$, '*': $p < 0.05$, '+': $p < 0.1$ (tail probability in hypergeometric distribution).

4 Notes

One critical question in applying *tigre* is whether to use the TF mRNA data. The extra information can potentially greatly help in identifying the TF activity profile and hence making better predictions of new targets, but it can also lead to incorrect predictions if it turns out incorrect. Based on our experience, the critical question is whether some outside influence, such as a external signal, is the rate-limiting factor in the production of the active TF. Simple post-translational modifications such as dimerisation to not appear to significantly hinder the use of TF mRNA data, as witnessed by the results in [6] for Drosophila TFs Mef2 and Twi which are known to function as dimers.

References

[1] Juan M Vaquerizas, Sarah K Kummerfeld, Sarah A Teichmann, and Nicholas M Luscombe. A census of human transcription factors: func-

tion, expression and evolution. *Nat Rev Genet*, 10(4):252–263, Apr 2009.

[2] Mihaela Pertea and Steven L Salzberg. Between a chicken and a grape: estimating the number of human genes. *Genome Biol*, 11(5):206, 2010.

[3] E. D. Sontag. For differential equations with r parameters, 2r+1 experiments are enough for identification. *J. Nonlinear Sci.*, 12:553–583, 2002.

[4] J. Stark, D. Brewer, M. Barenco, D. Tomescu, R. Callard, and M. Hubank. Reconstructing gene networks: what are the limits? *Biochem Soc Trans*, 31(Pt 6):1519–1525, Dec 2003.

[5] Pei Gao, Antti Honkela, Magnus Rattray, and Neil D Lawrence. Gaussian process modelling of latent chemical species: applications to inferring transcription factor activities. *Bioinformatics*, 24(16):i70–i75, Aug 2008.

[6] Antti Honkela, Charles Girardot, E. Hilary Gustafson, Ya-Hsin Liu, Eileen E M Furlong, Neil D Lawrence, and Magnus Rattray. Model-based method for transcription factor target identification with limited data. *Proc Natl Acad Sci U S A*, 107(17):7793–7798, Apr 2010.

[7] Martino Barenco, Daniela Tomescu, Daniel Brewer, Robin Callard, Jaroslav Stark, and Michael Hubank. Ranked prediction of p53 targets using hidden variable dynamic modeling. *Genome Biol*, 7(3):R25, 2006.

[8] M. Barenco, E. Papouli, S. Shah, D. Brewer, C. J. Miller, and M. Hubank. rHVDM: an R package to predict the activity and targets of a transcription factor. *Bioinformatics*, 25(3):419–420, Feb 2009.

[9] Giusy Della Gatta, Mukesh Bansal, Alberto Ambesi-Impiombato, Dario Antonini, Caterina Missero, and Diego di Bernardo. Direct targets of the TRP63 transcription factor revealed by a combination of gene

expression profiling and reverse engineering. *Genome Res*, 18(6):939–948, Jun 2008.

[10] Martino Barenco, Daniel Brewer, Efterpi Papouli, Daniela Tomescu, Robin Callard, Jaroslav Stark, and Michael Hubank. Dissection of a complex transcriptional response using genome-wide transcriptional modelling. *Mol Syst Biol*, 5:327, 2009.

[11] Robert C Gentleman et al. Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol*, 5(10):R80, 2004.

[12] Richard D Pearson, Xuejun Liu, Guido Sanguinetti, Marta Milo, Neil D Lawrence, and Magnus Rattray. puma: a Bioconductor package for propagating uncertainty in microarray analysis. *BMC Bioinformatics*, 10:211, 2009.

[13] Pan Du, Warren A Kibbe, and Simon M Lin. lumi: a pipeline for processing Illumina microarray. *Bioinformatics*, 24(13):1547–1548, Jul 2008.

[14] 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, 2010.

[15] Leonie Smeenk, Simon J van Heeringen, Max Koeppel, Marc A van Driel, Stefanie J J Bartels, Robert C Akkers, Sergei Denissov, Hendrik G Stunnenberg, and Marion Lohrum. Characterization of genome-wide p53-binding sites upon stress response. *Nucleic Acids Res*, 36(11):3639–3654, Jun 2008.

[16] Pauline A Fujita, Brooke Rhead, Ann S Zweig, Angie S Hinrichs, Donna Karolchik, Melissa S Cline, Mary Goldman, Galt P Barber, Hiram Clawson, Antonio Coelho, Mark Diekhans, Timothy R Dreszer, Belinda M Giardine, Rachel A Harte, Jennifer Hillman-Jackson, Fan

Hsu, Vanessa Kirkup, Robert M Kuhn, Katrina Learned, Chin H Li, Laurence R Meyer, Andy Pohl, Brian J Raney, Kate R Rosenbloom, Kayla E Smith, David Haussler, and W. James Kent. The UCSC Genome Browser database: update 2011. *Nucleic Acids Res*, 39(Database issue):D876–D882, Jan 2011.