# tigre Quick Guide

Antti Honkela, Pei Gao, Jonatan Ropponen,
Magnus Rattray, and Neil D. Lawrence

March 16, 2016

## 1 Abstract

The *tigre* package implements our methodology of Gaussian process differential equation models for analysis of gene expression time series from single input motif networks. The package can be used for inferring unobserved transcription factor (TF) protein concentrations from expression measurements of known target genes, or for ranking candidate targets of a TF.

The purpose of this quick guide is to present a small subset of examples from the User Guide that can be run very quickly. For a more comprehensive (although slower-running) presentation, please refer to the *tigre* User Guide.

## 2 Citing tigre

The *tigre* package is based on a body of methodological research. Citing *tigre* in publications will usually involve citing one or more of the methodology papers [1, 2, 3] that the software is based on as well as citing the software package itself [4].

## 3 Introductory example analysis - Drosophila development

In this section we introduce the main functions of the *puma* package by repeating some of the analysis from the PNAS paper [1][1].

### 3.1 Installing the tigre package

The recommended way to install *tigre* is to use the biocLite function available from the bioconductor website. Installing in this way should ensure that all appropriate dependencies are met.

```
> source("http://www.bioconductor.org/biocLite.R")
> biocLite("tigre")
```

To load the package start R and run

```
> library(tigre)
```

### 3.2 Loading the data

To get started, you need some preprocessed time series expression data. If the data originates from Affymetrix arrays, we highly recommend processing it with mmgmos from the *puma* package. This processing extracts error bars on the expression measurements directly from the array data to allow judging the reliability of individual measurements. This information is directly utilised by all the models in this package.

[1] Note that the results reported in the paper were run using an earlier version of this package for MATLAB, so there can be minor differences.

To start from scratch on Affymetrix data, the .CEL files from ftp://ftp.fruitfly.org/pub/embryo_tc_array_data/ may be processed using:

```
> # Names of CEL files
> expfiles <- c(paste("embryo_tc_4_", 1:12, ".CEL", sep=""),
+               paste("embryo_tc_6_", 1:12, ".CEL", sep=""),
+               paste("embryo_tc_8_", 1:12, ".CEL", sep=""))
> # Load the CEL files
> expdata <- ReadAffy(filenames=expfiles,
+                     celfile.path="embryo_tc_array_data")
> # Setup experimental data (observation times)
> pData(expdata) <- data.frame("time.h" = rep(1:12, 3),
+                              row.names=rownames(pData(expdata)))
> # Run mmgMOS processing (requires several minutes to complete)
> drosophila_mmgmos_exprs <- mmgmos(expdata)
> drosophila_mmgmos_fragment <- drosophila_mmgmos_exprs
```

This data needs to be further processed to make it suitable for our models. This can be done using

```
> drosophila_gpsim_fragment <-
+   processData(drosophila_mmgmos_fragment,
+               experiments=rep(1:3, each=12))
```

Here the last argument specifies that we have three independent time series of measurements.

In order to save time with the demos, a part of the result of this is included in this package and can be loaded using

```
> data(drosophila_gpsim_fragment)
```

## 3.3 Learning an individual model

Let us now learn a model for the TF twist and one of its potential targets

```
> # The probe identifier for TF 'twi'
> twi <- "143396_at"
> # The probe identifier for the target gene
> target <- "152715_at"
> # Learn the model using only one of the 3 repeats in the data
> model <- GPLearn(drosophila_gpsim_fragment[,1:12],
+                  TF=twi, targets=target,
+                  useGpdisim=TRUE, quiet=TRUE)
> # Display the model parameters
> show(model)
```

```
Gaussian process driving input single input motif model:
  Number of time points:
  Driving TF: 143396_at
  Target genes (1):
    152715_at
  Basal transcription rate:
    Gene 1: 0.000261535840329817
  Kernel:
    Multiple output block kernel:
    Block 1
    Normalised version of the kernel.
    RBF inverse width: 0.7638576 (length scale 1.144179)
    RBF variance: 1.755031
    Block 2
    Normalised version of the kernel
    DISIM decay: 0.2426514
    DISIM inverse width: 0.7638576 (length scale 1.144179)
```

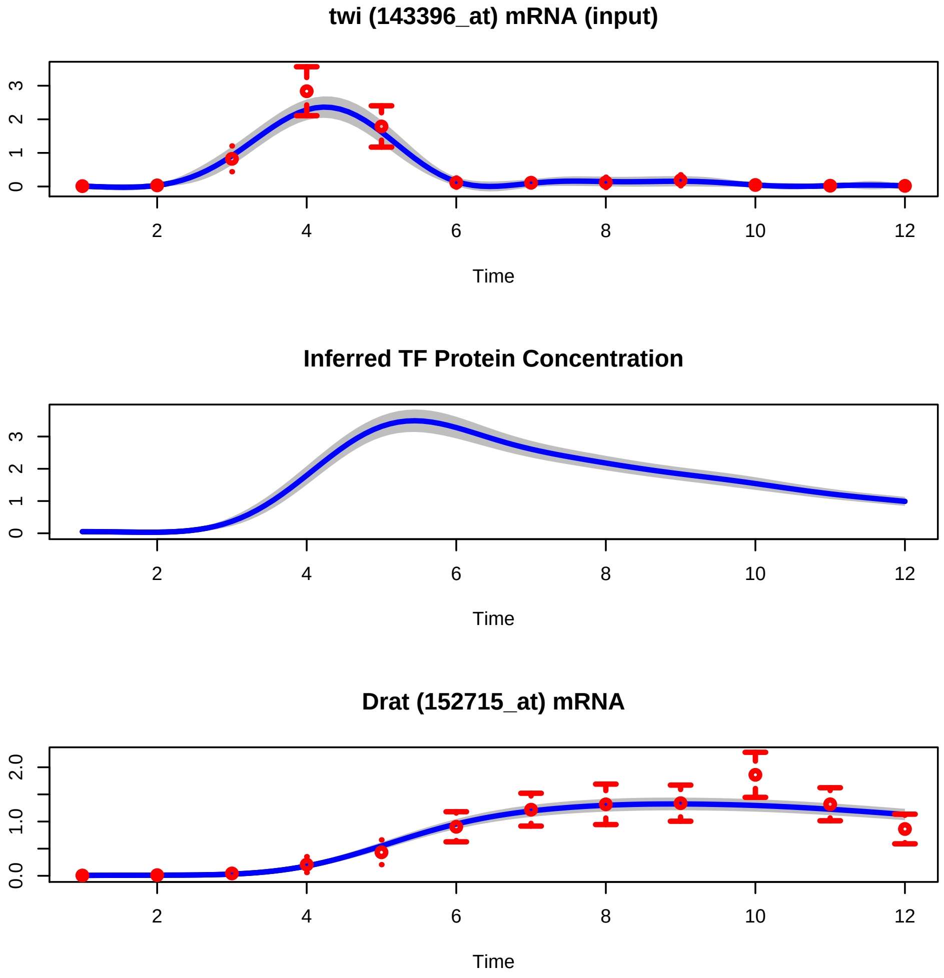

Figure 1: Single target models for the gene FBgn0003486. The models for each repeated time series are shown in different columns.

```
    DISIM Variance: 1
    SIM decay: 0.2427822
    SIM Variance: 0.02955712
    RBF Variance: 1.755031
  Log-likelihood: -1.005906
```

## 3.4 Visualising the model

The model can be plotted using the command

```
> GPPlot(model)
```

the results of which can be seen in Fig. 1.

## 3.5 Ranking the targets

Please refer to the User Guide for details on bulk ranking of candidate targets.

## 4 Session Info

```
> sessionInfo()
```

```
R version 3.2.4 Revised (2016-03-16 r70336)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 14.04.4 LTS

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8        LC_COLLATE=C
 [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] stats4    parallel  stats     graphics  grDevices
[6] utils     datasets  methods   base

other attached packages:
 [1] drosgenome1.db_3.2.2 org.Dm.eg.db_3.2.3
 [3] RSQLite_1.0.0        DBI_0.3.1
 [5] AnnotationDbi_1.32.3 IRanges_2.4.8
 [7] S4Vectors_0.8.11     tigre_1.24.2
 [9] Biobase_2.30.0       BiocGenerics_0.16.1

loaded via a namespace (and not attached):
 [1] XML_3.98-1.4       gtools_3.5.0
 [3] bitops_1.0-6       xtable_1.8-2
 [5] KernSmooth_2.23-15 gplots_2.17.0
 [7] annotate_1.48.0    gdata_2.17.0
 [9] BiocStyle_1.8.0    tools_3.2.4
[11] caTools_1.17.1
```

## References

[1] 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. URL: http://dx.doi.org/10.1073/pnas.0914285107, doi:10.1073/pnas.0914285107.

[2] 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. URL: http://dx.doi.org/10.1093/bioinformatics/btn278, doi:10.1093/bioinformatics/btn278.

[3] Neil D. Lawrence, Guido Sanguinetti, and Magnus Rattray. Modelling transcriptional regulation using Gaussian processes. In B. Schölkopf, J. C. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems*, volume 19, pages 785–792. MIT Press, Cambridge, MA, 2007.

[4] Antti Honkela, Pei Gao, Jonatan Ropponen, Magnus Rattray, and Neil D Lawrence. tigre: Transcription factor inference through gaussian process reconstruction of expression for bioconductor. *Bioinformatics*, 27(7):1026–1027, Apr 2011. URL: http://dx.doi.org/10.1093/bioinformatics/btr057, doi:10.1093/bioinformatics/btr057.