—Dynamic Nested Effects Models — An example in *murine embryonic stem cells* differentiation

Benedict Anchang*

November 13, 2009

Abstract

Cellular decision making in differentiation, proliferation or cell death is mediated by molecular signalling processes, which control the regulation and expression of genes. Vice versa, the expression of genes can trigger the activity of signalling pathways. Anchang et al (2009) introduce and describe a statistical method called Dynamical Nested Effects Model (D-NEM) for analyzing the temporal interplay of cell signalling and gene expression. D-NEMs are Bayesian models of signal propagation in a network. They decompose observed time delays of multiple step signalling processes into single steps. Time delays are assumed to be exponentially distributed. Rate constants of signal propagation are model parameters, whose joint posterior distribution is assessed via Gibbs sampling. They hold information on the interplay of different forms of biological signal propagation:

The method is implemented in the R package `dnem`. Here we demonstrate its practical application to embryonic stem cell development in mice. We show in detail how the data is pre-processed and discretized, how the initial pathway is reconstructed from nem (Markowetz *et al.* (2007)), and how we apply dnem to generate the set of posterior distribution for all model parameters in the network which could then be processed to generate a refined qualitative network.

1 murine embryonic stem cell RNAi data

We apply the dnem approach to a data set on the molecular mechanisms of self-renewal in murine embryonic stem cells. Ivanova et al. used RNA

*Institute of Functional Genomics, University of Regensburg, Josef-Engert-Str 9 , 93053 Regensburg, Germany. eMail: benedict.anchang@klinik.uni-regensburg.de ; URL: http://www-cgi.uni-regensburg.de/Klinik/FunktionelleGenomik

interference techniques to downregulate six gene products associated with self-renewal regulatory function, namely *Nanog*, *Oct4*, *Sox2*, *Esrrb*, *Tbx3* and *Tcl1*. They combined perturbation of these gene products with time series of microarray gene expression measurements. Mouse embryonic stem cells (ESC) were grown in the presence of the leukemia inhibitory factor LIF thus retaining their undifferentiated self-renewing state (positive controls). Cell differentiation associated changes in gene expression were detected by inducing differentiation of stem cells through removing LIF and adding retinoic acid (RA) (negative controls). Finally, RNAi based silencing of the six regulatory genes was used in (LIF+, RA-) cell cultures to investigate, whether silencing of these genes partially activates cell differentiation mechanisms. Time series at 6-7 time points in one-day intervals were taken for the positive control culture (LIF+, RA-), the negative control culture (LIF-, RA+), and the six RNAi assays. In the context of the dnem framework the six regulatory gene products *Nanog*, *Oct4*, *Sox2*, *Esrrb*, *Tbx3* and *Tcl1* are S-genes, while all genes showing significant expression changes in response to LIF depletion are used as E-genes. Downstream effects of interest are those, where the expression of an E-gene is pushed from its level in self-renewing cells to its level in differentiated cells. Our goal is to model the temporal occurrence of these effects across all time series simultaneously.

Dataset summary, Preprocessing and E-gene selection We used log2 transformed values of MAS5.0 normalized data obtained from http://www.nature.com/nature/ journal/v442/n7102/suppinfo/nature04915.html. The dataset consists of 8 time series with 6-8 time points at one-day intervals.One time series for self-renewing stem cells (LIF+, RA-), one time series for cells passing through early differentiation (LIF-, RA+), and 6 time series for LIF stimulated ESCs with one of the six regulators *Nanog*, *Oct4*, *Sox2*, *Esrrb*, *Tbx3* and *Tcl1* silenced by RNAi. In a comparison of the (LIF+, RA-) to the (LIF-, RA+) cell cultures 135 genes showed a greater than 2 fold up or down regulation across all time points. These were used as E-genes in our analysis.

```
> library(dnem)
> data("stemcellexpr")
```

The internal function *nem.discretize* implements the following two preprocessing steps: First, we select the genes as effect reporters (E-genes), which showed a greater than 2 fold up or down regulation across all time points in comparison to (LIF+, RA-) and (LIF-, RA+).Next we transform the continuous expression data to binary values. We set an *E-gene* in a certain silencing experiment and time point to 1, if its expression value is sufficiently close to the negative controls, i.e. the intervention interrupted the

information flow, otherwise we set it to 0. Let $C(i,k,s)$ denote the continuous expression measurement of E_k at time point t_s of a time series recorded after perturbation of S_i. Moreover, let $C^+(k,s)$ and $C^-(k,s)$ denote the corresponding measurements in positive and negative controls respectively. We set

$$D(i,k,s) = \begin{cases} 1 & \text{if} \quad C(i,k,s) < \kappa \cdot C^+(k,s) + (1-\kappa) \cdot C^-(k,s) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Discretization and smoothing We need binary data for each gene at each time point for each condition. Note that we have only three measurements per constellation:1 negative control, 1 positive control and the measurement from the RNAi assay. In order to obtain robust estimates, data needs to be aggregated across time points. D-NEMs assume that once a perturbation effect has reached an *E-gene*, it persists until the end of the time series.In other words, a one at time point t indicates that a downstream effect has reached the *E-gene* prior to t and not that it is still observable at this time. Hence, a typical discretized time series starts with zeros, eventually switches to ones and then stays one until the end of the series. We refer to these patterns as admissible patterns. For the vast majority of *E-genes*, the discretized data roughly follows admissible patterns. Nevertheless, exceptions are observed most likely due to measurement noise. We replace the time series for each gene by the closest admissible pattern, based on edit distances. In the case where several admissible patterns had the same edit distance to the time series, we chose the pattern holding the most ones.All the above is done by one single function `nem.discretize.smooth`

```
data(stemcellexpr)
D <- nem.discretize.smooth(Data2, neg.control = 1:3,
     pos.control = 2:4,cutoff = 0.7)
```

2 Applying Dynamic Nested Effects Models

Since long computation times for Gibbs sampling prohibit the reconstruction of the nework's topology from scratch using D-NEMs, we used the triplet search approach for the standard nested effect approach Markowetz applied to the final time point to determine a topology for the network. Note, that the final time point of an admissible pattern accumulates information along the time series, because it reports a one whenever a downstream signal has reached the E-gene at any time. The first network in Figure 1 shows the reconstructed network from the last time point. A nested structure is visible with the following backbone linear structure apparent: *Nanog* $\longrightarrow$ *Sox2* $\longrightarrow$ *Oct4* $\longrightarrow$ *Tcl1*. The topology is based exclusively on the nesting of downstream effects. Time delays of signal propagation can now be used for fine tuning the topology with the help of Gibbs sampling.

```
Data=D[[1]] # binary time series data
dim(Data)
Data=Data[-which(rowSums(Data[,,8])==0),,]
# Remove all E-genes with no effect across all S-genes
control = set.default.parameters(unique(colnames(Data)),
para=c(0.1, 0.1))
G<- nem(Data[,,8],inference="triples", control=control)$graph
# double edge between Oct4-Tcl
m<-graph2adj(G)
m[6,2]=0 # Remove edge Tcl->Oct4 to make graph DAG
G1=adj2graph(m)
D0<-array(0,dim=c(122,6,8))
colnames(D0)<-colnames(Data)
rownames(D0)<-rownames(Data)
Data1<-abind(D0[,,1],Data,along=3)
# set initial time t0 effects to 0 for all E-genes
```

For running the Gibbs sampler we initialize all model parameters as follows: The output is stored in a different directory for covenience and post processing of results.

```
theta.init=sample(c(1,2,3,4,5,6),122,replace=TRUE)
 # initial E-S gene positions
lags.init=sample(c(0,1,2,3,4,5,6,7,8),20,replace=TRUE)
# initial time lags
B=2  # number of iterations
a=0.2  # model type1 error
b=0.1  # model type2 error
```

```
n.chains=1 # markov chains
## Running the Gibbs sampler #####
Output2<-dnem(n.chains,Data1,B,G1,theta.init,lags.init,
a,b,delta=0.2,file="output path name")
```

2.1 Output of Dynamic Nested Effects Models

After running the Gibbs sampler for 10000 iterations with 3 chains we discard a burn in of 3000 and summarize the remaining posterior values into (1.) A heatmap of posterior distribution of average time delays with rows corresponding to edges of the input network including those between S-genes and E-genes and columns refering to average time delays. Marginal posterior probabilities are gray-scale coded (2.) Prior transively closed graph and a posterior graph containing signaling and nonsignaling edges. For convergence diagnostics one can use the mcmc package. This is not implemented in this package. We assume here that sampled chains have reached convergence. See supplement material from Anchang et.al (2009) for convergence diagnostics.

```
fig=dnemoutput(G,file1="input path name",
file2="output path name",burnin=1000,np=20,T=8,cutoff=0.4)
```

Fig.3 shows the results of the stem cell data analysis. The heatmap correspond to posterior distribution of average time delays. Rows correspond t edges of the network including those between S-E genes, where as columns refer to average time delays . Marginal posterior probabilities are gray-scale coded. Next to the Heat map is the transitively closed nested effects model estimated from last time poit stem cell data using static NEM. The rightmost figure corresponds to the final network structure estimated by time delayanalysis using D-NEM. Edges in Red are non existent after D-NEM modeling. The rate of signal flow on the edges can be directly estimated from the heat map.

Conclusion From this tutorial,we have used the package 'dnem' to estimate signal propagation rates in a network from time series data of perturbation effects using gibbs sampling. The package takes as input (1.) a transitively closed directed graph, representing a hypothetical pathway , and (2.) binary time series high-dimensional phenotypic readout of perturbation effects (e.g. gene expression or morphological profiles). The output is a non necessarily transitive directed graph representing the phenotypic hierarchy with edges representing the rate of signal flow.

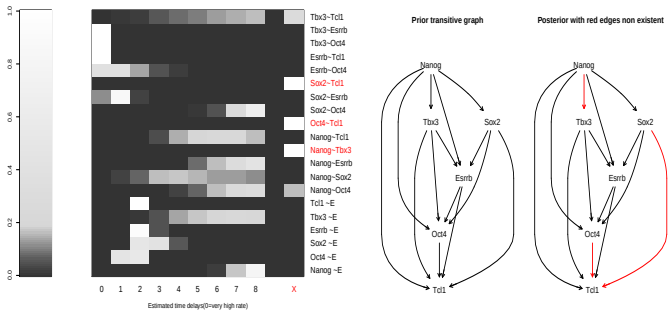

Figure 1: Heatmap of posterior distribution with input and output graph of signal flow

References

[1] Anchang, B ,Mohammad J., Juby Jacob, Marcel O. Vlad, Peter J. Oefner, Achim Tresch and Rainer Spang (2009) Modeling the temporal interplay of molecular signaling and gene expression by using dynamic nested effects models *PNAS* 106(16): 6447-6452.

[2] Ivanova, N and Dobrin, R and Lu, R and Kotenko, I and Levorse, J and Decoste, C and Schafer, X and Lun, Y and Lemischka, I (2006) Dissecting self-renewal in stem cells with RNA interference *Nature* 442:533-538.

[3] Markowetz, F and Kostka, D and Troyanskaya, O G and Spang, R (2007) Nested effects models for high-dimensional phenotyping screens *Bioinformatics* 13:i305-12.

[4] Froehlich, H and Fellmann, M and Sueltmann, H and Poustka, A and Beissbarth, T (2008) Estimating Large Scale Signaling Networks through Nested Effect Models with Intervention Effects from Microarray Data *Bioinformatics* 10.

[5] Froehlich, H and Beissbarth, T and Tresch, A and Kostka, D and Jacob, J and Spang, R and Markowetz, F (2008) Analyzing Gene Perturbation Screens With Nested Effects Models in R and Bioconductor *Bioinformatics* 2.

[6] Markowetz, F and Bloch, J and Spang, R (2005) Non-transcriptional pathway features reconstructed from secondary effects of RNA interference *Bioinformatics* 21:4026-32.

Session Information

The version number of R and packages loaded for generating the vignette were:

- R version 2.9.1 (2009-06-26), `x86_64-unknown-linux-gnu`
- Locale: `C`
- Base packages: base, datasets, grDevices utils, graphics, methods, stats, grid
- Other packages: Biobase 1.12.2, RBGL 1.18.0, Rgraphviz 1.22.0, RColorBrewer 1.0-2, annotate 1.12.0, class 7.2-46, e1071 1.5-19, geneplotter 1.12.0,bootstrap 1.0-22, graph 1.22.0, nem 2.8.0, time 1.0, plotrix 2.5-4, cluster 1.12.0