Dynamic Deterministic Effects Propagation Networks (DDEPN) - exemplary workflow

Christian Bender *

June 21, 2010

Abstract

Network modelling in systems biology has become an important tool to study molecular interactions, especially in the medical field like cancer research. The understanding of the interplay of proteins in cellular signalling is the basis for the development of novel drugs and therapies. Here, we set up a new method for the reconstruction of signalling networks from time course protein data after external perturbation. We show how to use protein expression and phosphorylation data measured on Reverse Phase Protein Arrays to infer a signalling network among proteins of the ERBB signalling cascade in a human breast cancer cell line.

Our method models the signalling dynamics by a boolean signal propagation mechanism that defines a sequence of state transitions for a given network structure. A likelihood score is proposed that describes the probability of our measurements given a particular state transition matrix. We identify the optimal sequence of state transitions via a Hidden Markov Model. Network structure search is performed by a genetic algorithm that optimises the overall likelihood of a population of candidate networks. We test our method on simulated networks and data and show its increased performance in comparison to another Dynamical Bayesian Network approach. The reconstruction of a network in our real data results in several known signalling chains from the ERBB network, showing the validity and usefulness of our approach.

*German Cancer Research Center, Im Neuenheimer Feld 580, 69120 Heidelberg, Germany. eMail: c.bender@dkfz-heidelberg.de

1 Using DDEPN for network inference on simulated data sets

This document shows an exemplary workflow to reconstruct a signalling network from simulated data. An analysis on real data can be performed analogously.

2 Simulating data

In this section we show how to generate artificial networks and data. A reference signalling network is simulated and used to sample measurements that incorporate the network structure.

First, simulate a network with 6 nodes and 2 distinct input stimuli.

```
> set.seed(12345)
> n <- 6
> signet <- signalnetwork(n = n, nstim = 2, cstim = 0, prop.inh = 0.2)
> net <- signet$phi
> stimuli <- signet$stimuli
> weights <- signet$weights
```

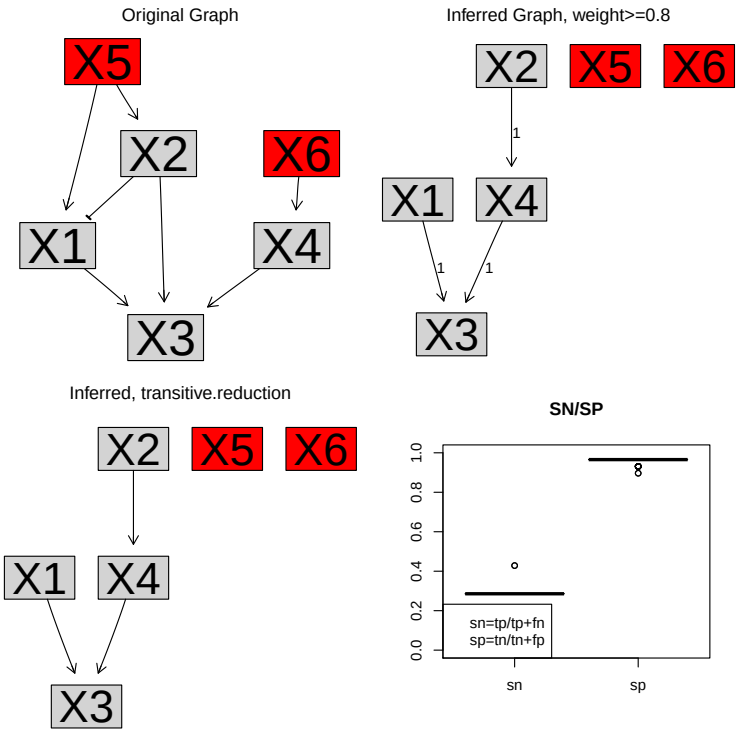
Second get intensities for each protein that are based on the network structure generated above.

```
> dataset <- makedata(net, stimuli, mu.bg = 1200, sd.bg = 400,
+     mu.signal.a = 2000, sd.signal.a = 1000)
```

3 Running the genetic algorithm

Now run the genetic algorithm to reconstruct the network from above. Alternatively, if real are used, reconstruct the unknown network structure, based on the measurements alone.

```
> ret <- ddepn(dataset$datx, phiorig = net, phi = NULL, stimuli = NULL,
+     th = 0.5, inference = "netga", pdf = NULL, multicores = TRUE,
+     maxiterations = 50, p = 100, q = 0.3, m = 0.8, P = NULL,
+     usebics = TRUE, cores = 2, lambda = NULL, B = NULL, maxiter = 100)
```



4 Examining the results

After the reconstruction, the generated network can be viewed as follows:

```
> plotrepresult(ret)
```

Session Information

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

- R version 2.10.1 (2009-12-14), `x86_64-pc-linux-gnu`
- Locale: `LC_CTYPE=en_US.utf8`, `LC_NUMERIC=C`, `LC_TIME=en_US.utf8`, `LC_COLLATE=en_US.utf8`, `LC_MONETARY=C`, `LC_MESSAGES=en_US.utf8`, `LC_PAPER=en_US.utf8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.utf8`, `LC_IDENTIFICATION=C`

- Base packages: base, datasets, graphics, grDevices, methods, stats, tools, utils