netresponse
probabilistic tools for functional network analysis

Leo Lahti[1,2]*, Olli-Pekka Huovilainen[1], António Gusmão[1] and Juuso Parkkinen[1]

(1) Dpt. Information and Computer Science, Aalto University, Finland
(2) Dpt. Veterinary Bioscience, University of Helsinki, Finland

April 13, 2011

1 Introduction

Condition-specific network activation is characteristic for cellular systems and other real-world interaction networks. If measurements of network states are available across a versatile set of conditions or time points, it becomes possible to construct a global view of network activation patterns. Different parts of the network respond to different conditions, and in different ways. Systematic, data-driven identification of these responses will help to obtain a holistic view of network activity [2, 1]. This package provides robust probabilistic algorithms for functional network analysis [1, 5].

The methods are based on nonparametric probabilistic modeling and variational learning, and provide general exploratory tools to investigate the structure (ICMg; [5]) and context-specific behavior (NetResponse; [1]) of interaction networks. ICMg is used to identify community structure in interaction networks; NetResponse detects and characterizes subnetworks that exhibit context-specific activation patterns across versatile collections of functional measurements, such as gene expression data. The implementations are partially based on the agglomerative independent variable group analysis [3] and variational Dirichlet process Gaussian mixture models [4]. The tools are particularly useful for global exploratory analysis of genome-wide interaction networks and versatile collections of gene expression data.

2 Loading the package and example data

Load the package and toy data set. The *toydata* object contains the variables D (gene expression matrix) and *netw* (network matrix). The data matrix D describes measurements of the network activation over multiple conditions. This simple toy data will be analyzed in the subsequent examples. Note that the method is potentially applicable to networks with thousands of nodes and conditions; the scalability depends on network connectivity.

*leo.lahti@iki.fi

```
> library(netresponse)
> data(toydata)
> D <- as.matrix(toydata$emat)
> netw <- as.matrix(toydata$netw)
```

3 Detecting network responses

Detect network responses across the different measurement conditions in the data matrix D:

```
> model <- detect.responses(D, netw, verbose = FALSE)
```

Various network formats are supported, see help(detect.responses) for details. With large data sets, consider using the 'speedup' option.

4 Investigating the results

Subnetwork statistics: size and number of distinct responses for each subnet

```
> stat <- model.stats(model)
> stat
```

```
         subnet.size subnet.responses
Subnet-1           7                1
Subnet-2           3                3
```

List the detected subnetworks (each is a list of nodes). By default, singleton subnetworks (with only one gene) and subnetworks with only a single response (no differences between conditions) are excluded. To change the defaults, see help(get.subnets). Subnetworks can be filtered by size and number of responses. Subnetworks that have only one response are not informative of the differences between conditions, and typically ignored in subsequent analysis.

```
> get.subnets(model, min.size = 2, min.responses = 2)
```

```
$`Subnet-2`
[1] "feat4" "feat5" "feat6"
```

Each subnetwork response has a probabilistic association to each condition. Get the list of samples corresponding to each response (each sample is assigned to the response of the highest probability) with response2sample function.

```
> subnet.id <- "Subnet-2"
> response2sample(model, subnet.id)
```

Retrieve model parameters of a given subnetwork (Gaussian mixture means, covariance diagonal, and component weights):

```
> pars <- get.model.parameters(model, subnet.id)
> pars
```

```
$mu
                 feat4      feat5       feat6
Response-1  0.08115179  2.9798781 -0.02811302
Response-2 -4.96940573  0.2077142  1.92109335
Response-3  4.78052024 -3.0683949 -3.17463435

$sd
               feat4     feat5     feat6
Response-1 1.0718400 1.0609357 1.0743768
Response-2 0.9218103 0.8725388 0.9572093
Response-3 1.0882094 0.9627731 0.8818674

$w
Response-1 Response-2 Response-3
 0.3480114  0.3115483  0.3404403

$free.energy
         [,1]
[1,] 1156.278

$Nparams
[1] 21

$nodes
[1] "feat4" "feat5" "feat6"
```

Probabilistic sample-response assignments for a given subnet is retrieved with:

```
> response.probabilities <- sample2response(model, subnet.id)
```

5 Extending the subnetworks

After identifying the locally connected subnetworks, it is possible to search for features (genes) that are similar to a given subnetwork but not directly interacting with it. To order the remaining features in the input data based on similarity with the subnetwork, type

```
> g <- find.similar.features(model, subnet.id = "Subnet-1")
> subset(g, delta < 0)

      feature.name     delta
feat5        feat5 -79.48474
feat6        feat6 -52.05103
feat4        feat4 -20.43436
```

This gives a data frame which indicates similarity level with the subnetwork for each feature. The smaller, the more similar. Negative values of delta indicate the presence of coordinated responses, positive values of delta indicate independent responses. The data frame is ordered such that the features are listed by decreasing similarity.

6 Nonparametric Gaussian mixture models

The package provides additional tools for nonparametric Gaussian mixture modeling based on variational Dirichlet process mixture models and implementations by [4, 3]. See the example in help(vdp.mixt).

7 Interaction Component Model for Gene Modules

Interaction Component Model (ICMg) can be used to find functional gene modules [5] from either protein interaction data or from combinations of protein interaction and gene expression data.

A short example of how to run ICMg and obtain clustering for the nodes:

```
> library(netresponse)
> data(osmo)
> res <- ICMg.combined.sampler(osmo$ppi, osmo$exp, C = 10)
> res$comp.memb <- ICMg.get.comp.memberships(osmo$ppi, res)
> res$clustering <- apply(res$comp.memb, 2, which.max)
```

8 Citing NetResponse

Please cite [1] when using the package. When using the ICMg algorithms, additionally cite [5].

9 Version information

This document was written using:

```
> sessionInfo()

R version 2.13.0 alpha (2011-03-17 r54849)
Platform: x86_64-unknown-linux-gnu (64-bit)

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
 [5] LC_MONETARY=C              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] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] netresponse_1.3.44 minet_3.4.0        infotheo_1.1.0     graph_1.28.0
[5] igraph_0.5.5-2
```

```
loaded via a namespace (and not attached):
[1] tools_2.13.0
```

References

[1] Leo Lahti *et al.* (2010). Global modeling of transcriptional responses in interaction networks. *Bioinformatics*. Preprint: http://www.cis.hut.fi/lmlahti/publications/Lahti10bioinf-preprint.pdf

[2] Leo Lahti (2010). Probabilistic analysis of the human transcriptome with side information. PhD thesis. Aalto University School of Science and Technology, Department of information and Computer Science, Espoo, Finland, 2010. http://lib.tkk.fi/Diss/2010/isbn9789526033686/

[3] Antti Honkela *et al.* (2008). Agglomerative independent variable group analysis. *Neurocomputing* 71, 1311–1320.

[4] Kenichi Kurihara *et al.* (2007). Accelerated variational Dirichlet process mixtures. In B. Schölkopf, J. Platt, and T. Hoffman, eds., *Advances in Neural Information Processing Systems 19*, 761–768. MIT Press, Cambridge, MA.

[5] Parkkinen, J., and Kaski, S. Searching for functional gene modules with interaction component models. *BMC Systems Biology 4* (2010), 4.