

netresponse

Leo Lahti, Olli-Pekka Huovilainen and António Gusmão

January 17, 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. The NetResponse algorithm, implemented in this R package, provides efficient probabilistic tools for such analysis [1].

In summary, the algorithm detects local subnetworks that show particular activation patterns in a subset of conditions. The algorithm characterizes these responses, and identifies their activating conditions. The algorithm can be viewed as a type of subspace clustering where the feature space is constrained by the network. The method is based on nonparametric probabilistic modeling and variational learning, and it provides general exploratory tools for functional network analysis.

The current version provides the algorithmic implementation. The implementations are partially based on the agglomerative independent variable group analysis [2] and variational Dirichlet process mixture model algorithms [3]. Further tools for visualization and analysis will be provided in the later versions.

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.

```
> require(netresponse)
> data(toydata)
> D <- toydata$emat
> netw <- toydata$netw
```

3 Detecting network responses

Detect network responses across the conditions:

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

4 Analyzing the results

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

```
> stat <- result.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)

```
> get.subnets(model)

$`Subnet-1`
[1] "feat1" "feat2" "feat3" "feat7" "feat8" "feat9" "feat10"

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

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, stat = stat, min.size = 2, min.responses = 2)

$`Subnet-1`
[1] "feat1" "feat2" "feat3" "feat7" "feat8" "feat9" "feat10"

$`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)

```
> 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 0.08115196 2.9798782 -0.02811313
Response-2 -4.96940560 0.2077142  1.92109334
Response-3  4.78052024 -3.0683949 -3.17463435

```

```

$sd
      feat4      feat5      feat6
Response-1 1.0352970 1.0300173 1.0365215
Response-2 0.9601096 0.9340979 0.9783707
Response-3 1.0431728 0.9812100 0.9390779

```

```

$w
Response-1 Response-2 Response-3
0.1858341  0.3765066  0.4376592

```

```

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

```

```

$K
NULL

```

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

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

5 Variational Dirichlet process Gaussian mixture model

The package provides additional tools for nonparametric Gaussian mixture modeling based on previous work and implementations by [3, 2]. See the example in `help(vdp.mixt)`.

6 Citing NetResponse

Please cite [1] when using the package.

7 Version information

This document was written using:

```
> sessionInfo()
```

```
R version 2.12.1 (2010-12-16)
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=C
[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_1.0.2
```

loaded via a namespace (and not attached):

```
[1] tools_2.12.1
```

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