Network Mining Practical
US28-mediated signaling

M. El-Kebir G. W. Klau

September 4, 2015

Contents

1 Introduction

The *Human Cytomegalovirus (HCMV)* is a highly-contagious herpes virus [3]. Infection with HCMV in healthy humans usually does not result in symptoms. However, in humans with a compromised immune system the virus is correlated with diseases such as hepatitis and retinitis [7]. The virus may also have an oncogenic potential [1, 2, 4].

HCMV is responsible for the production of four viral G protein-coupled receptors (vGPCRs). Of these vGPCRs, US28 is the most studied and found to be hijacking the host cells

signaling pathways and stimulating proliferative signaling pathways [5, 6]. In this practical we re-analyze the same dataset using Heinz and eXamine.

Instructions are typeset in a different font: shell commands are prefixed by '$' whereas R commands do not have a prefix. There are also questions in the text, they are boxed. To set up the environment, please download and extract the following file:

```
$ wget http://homepages.cwi.nl/~elkebir/NBIC/practical.tar.bz2

$ tar xvjf practical.tar.bz2
```

2 Prerequisites

The following R packages are required.

- affy
- limma
- BioNet
- topGO

These packages can be installed as follows.

```
source("http://bioconductor.org/biocLite.R")
biocLite("affy")
biocLite("limma")
biocLite("BioNet")
biocLite("topGO")
```

- Cytoscape: download from `http://www.cytoscape.org/download.php`
- Heinz: download from `https://software.cwi.nl/software/heinz` and extract the tarball into `bin/`
- eXamine: install from `http://apps.cytoscape.org/apps/examine` by clicking on install after having started Cytoscape.

3 Processing the microarrays

The dataset consists of four Affymetrix Mouse Genome 430 2.0 arrays, of which two correspond to mock transfection and the other two correspond to US28 transfection of NIH-3T3 mouse cell lines:

- mock #1: `060804MJA_moe4302.0_RD135.CEL`,

- mock #2: `060804MJA_moe4302.0_RD136.CEL`,
- US28 #1: `060804MJA_moe4302.0_RD137.CEL`,
- US28 #2: `060804MJA_moe4302.0_RD138.CEL`.

We process these microarrays using `affy`, which uses RMA to perform background correction and normalization and also computes the expression values.

```
$ cd heinz-practical

$ R

library(affy)

# 1. Read Affymetrix data (CEL files)
myAffy = ReadAffy()

# 2. Preprocessing using RMA (Robust Multi-array Average) method:
#      - background correction
#      - data normalization
#      - expression calculation
# returns an instance of the ExpressionSet class to use in further analysis
# expression values calculated in RMA are in log2 scale
myAffyRMA = rma(myAffy)
```

Q1. How many probes are there in each microarray?

The next step is compute p-values that assess for each probe its differential expression. We do this using `limma`.

```
library(limma)

# 4. LIMMA - Linear Models for Microarray Data
# 4.a. create a design matrix to extract from the fit the contrasts of interest
samples <- c("mock", "mock", "wt", "wt")
design <- model.matrix(~factor(samples))
colnames(design) <- c("mock", "wt")

# 4.b. fit the design to a model and apply empirical Bayes method
fit <- eBayes(lmFit(myAffyRMA, design))

# 4.c. extract the results and save them to limma_results.txt
results <- topTable(fit, coef = 2, adjust = "fdr", p.value = 1, sort.by = "logFC",
n = dim(myAffyRMA)[1])
write.table(results[,c("P.Value","logFC")], file = "limma_results.txt", sep= "
", quote = FALSE, row.names = TRUE, col.names = FALSE)
```

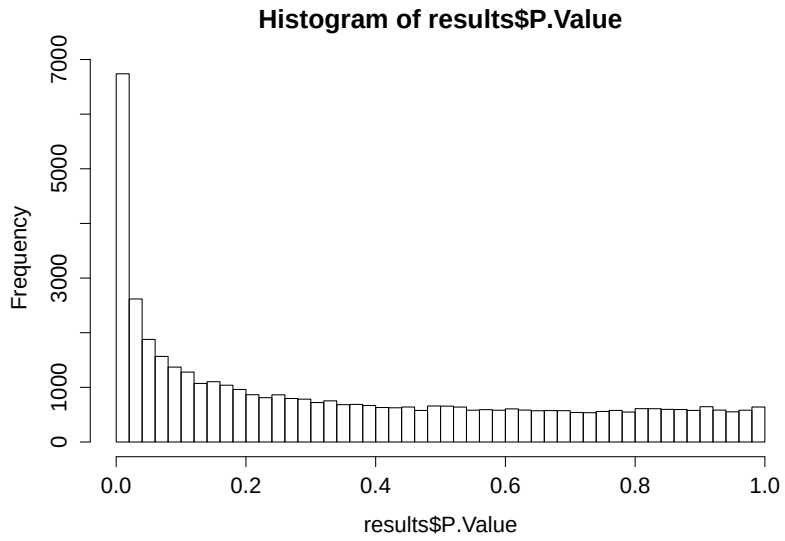

Figure 1: Probes p-value distribution

Q2. Which probe has the smallest p-value?

Let's take a look at the p-value distribution:

```
hist(results$P.Value, n=50)
```

The distribution looks good (see Figure 1): it resembles a beta-uniform mixture.

3.1 Fitting the BUM model

We are now all set to fit a BUM model. We do this in R using the `BioNet` package:

```
library(BioNet)
```

```
> fitBumModel(results$P.Value)
Beta-Uniform-Mixture (BUM) model

45101 pvalues fitted

Mixture parameter (lambda): 0.167
shape parameter (a):        0.463
log-likelihood:             13084.1
```

So we have $\hat{\lambda} = 0.167$ and $\hat{a} = 0.463$.

4 Generating Heinz input

The network that we use is given in the file `KEGG_mouse_noDupl.txt`. Let's look at its contents.

```
$ head KEGG_mouse_noDupl.txt
ENSMUSG00000057003 ENSMUSG00000062825
ENSMUSG00000057003 ENSMUSG00000068876
ENSMUSG00000057000 ENSMUSG00000094215
ENSMUSG00000057000 ENSMUSG00000060244
ENSMUSG00000002010 ENSMUSG00000022477
ENSMUSG00000002010 ENSMUSG00000030541
ENSMUSG00000002010 ENSMUSG00000025950
ENSMUSG00000002010 ENSMUSG00000028405
ENSMUSG00000002010 ENSMUSG00000021913
ENSMUSG00000002010 ENSMUSG00000020456
```

So every line defines an edge between two nodes.

> Q3. Which type of identifier is used for the nodes?

What's the size of the network? We can find this out using some shell magic:

```
$ wc -l KEGG_mouse_noDupl.txt
   36932 KEGG_mouse_noDupl.txt

$ cat KEGG_mouse_noDupl.txt | tr " " "\n" | sort -u | wc -l
    5082
```

> Q4. Can you explain what the last command exactly does?

So the network has 5082 nodes and 36932 edges.

4.1 Mapping the probes

Let's look at the results of the microarray analysis step:

```
$ head limma_results.txt
1423606_at 1.67199412482554e-06 -6.24104549326084
1439665_at 2.3696421341743e-05 -6.15530782855658
1435605_at 2.01964195887092e-05 -6.15315234258701
1429992_at 0.00312482344206735 -5.54932640637315
1418937_at 0.00040688700766659 5.37895354750547
1434186_at 0.000145803029605795 -4.98060824577276
```

```
1429993_s_at 0.00163952563986902 -4.96280675741293
1416666_at 0.0276548573948993 -4.95151701575721
1436736_x_at 0.000307018018399313 -4.91005798055806
1437990_x_at 1.67295164302197e-07 4.89936970869032
```

So in order to overlay the probes on the KEGG network, we need to map the probe identifiers to Ensembl gene identifiers. This is done in the script `scripts/mapProbes.py`.

```
$ scripts/mapProbes.py affy2ensg.txt limma_results.txt > mapping_results.txt

$ wc -l mapping_results.txt
   16747 results/mapping_results.txt

$ head results/mapping_results.txt
ENSMUSG00000025968 1425143_a_at 0.968757650956378 -0.00690465431073051
ENSMUSG00000028180 1454652_at 0.429097209632422 0.154985162174474
ENSMUSG00000028182 1431957_at 0.850211750759503 0.0274331285509192
ENSMUSG00000002017 1416445_at 0.81469589363243 0.0280679795983616
ENSMUSG00000028184 1444906_at 0.381184775020791 0.123345739270905
ENSMUSG00000002015 1456279_a_at 0.531692890368797 0.0637517921743083
ENSMUSG00000002014 1435709_at 0.783911900640168 0.0362625065957071
ENSMUSG00000028189 1429943_at 0.643985859404648 0.237878765678175
ENSMUSG00000028188 1429662_at 0.0192131600849529 -0.573163121708832
ENSMUSG00000053218 1441135_at 0.172333562454501 0.201016590393955
```

Q5. How does the script deal with probes that map to different Ensembl gene IDs?
Q6. How many such probes are there?
Q7. How many probes are there that do not map to a gene present in the network?

4.2 Taking the intersection

The next step is to intersect `mapping_results.txt` with `KEGG_mouse_noDupl.txt`. We do this using `scripts/induce.py`.

```
$ scripts/induce.py mapping_results.txt KEGG_mouse_noDupl.txt > nodes.txt 2> edges.txt
```

Q8. What do '>' and '2>' in the command above mean?
Q9. How many nodes and edges does the resulting network have?

5 Running Heinz

Heinz can be run as follows, which takes a few minutes:

```
$ bin/heinz -p -n nodes.txt -e edges.txt -a 0.463 -lambda 0.167 -FDR 4e-3 > module.dot
```

The parameters mean the following:

- `-p`: enable preprocessing,
- `-n nodes.txt`: nodes input file,
- `-e edges.txt`: edges input file,
- `-a 0.463`: sets $\hat{a}$ to 0.463,
- `-lambda 0.167`: sets $\hat{\lambda}$ to 0.167,
- `-FDR 4e-3`: sets FDR to 0.004.

The resulting module can be visualized using `graphviz` in the following way.

```
$ neato -Tpdf module.dot -o module.pdf
```

> Q10. What happens when the FDR is increased?
> Q11. And what happens when it is decreased?

In `input/Heinz_Nodes_FDR_7e-04.txt` the nodes score that were used to generate the module referred to in the presentation are given. Observe that this time no p-values but actual scores are used.

```
$ head Heinz_Nodes_FDR_7e-04.txt
#node score1
ENSMUSG00000057003 -5.07238096949341
ENSMUSG00000057000 -5.48958095921254
ENSMUSG00000002010 -4.16253938119165
ENSMUSG00000002015 -4.14682702039506
ENSMUSG00000002014 -3.50225399424938
ENSMUSG00000021135 -5.23401263118499
ENSMUSG00000040147 -4.5724581899087
ENSMUSG00000046598 -5.996673488742
ENSMUSG00000074886 -1.30085060276539
```

To run heinz on `input/nodes_7e-4.txt`, we omit the parameters `-a`, `-lambda` and `-FDR`. This will take a few minutes.

```
$ bin/heinz -p -n Heinz_Nodes_FDR_7e-04.txt -e KEGG_mouse_noDupl.txt -o US28_module.hnz
> US28_module.dot

$ neato -Tpdf US28_module.dot -o US28_module.pdf
```

```
$ head -n3 US28_module.hnz
#label score
ENSMUSG00000044485 NaN
ENSMUSG00000063713 NaN
```

The output file `US28_module.hnz` contains all the nodes in the network appended with either the node score in case the node is part of the solution otherwise NaN is printed. Let's extract only the nodes that are in the solution:

```
$ grep -v NaN US28_module.hnz > US28_module.txt
```

Q12. How many nodes are there in US28_module?

The resulting file `US28_module.txt` will serve as input for the enrichment analysis.

6 Performing enrichment analysis

6.1 Gene Ontology enrichment

In the file `ensgid2go.map` the correspondence between Ensembl gene IDs and their GO terms is given. This is the file that we use as input to `topGO`, which is wrapped in `scripts/enrichment.R`.

```
$ scripts/enrichment.R ensgid2go.map US28_module.txt MF
```

The third argument of the script is the Gene Ontology (GO) category:

- `MF`: molecular function
- `BP`: biological process
- `CC`: cellular component

Q13. What is the 50th most enriched term in category `MF`?

To perform GO enrichment on the other categories do the following.

```
$ scripts/enrichment.R ensgid2go.map US28_module.txt BP

$ scripts/enrichment.R ensgid2go.map US28_module.txt CC
```

6.2 KEGG pathway enrichment

In `pathways.txt` a list of KEGG mouse pathways is given. The mapping between Ensemble IDs and its containing pathways is given in `mapping-ensg-pathways.txt`. Pathway enrichment can be performed as follows.

```
$ cd scripts

$ ./enrichmentPathway.sh ../pathways.txt ../US28_module.txt ../mapping-ensg-pathways.txt
```

Q14. What is the most enriched pathway?

7 Inspecting the module using eXamine

You can setup the Cytoscape networks yourself by following the tutorial on `http://homepages.cwi.nl/~elkebir/eXamine/tutorial.pdf`. To save time, you could also open up the session file `examine/US28.cys`.

Follow the steps listed in pages 8–14 of the paper. Additional questions:

Q15. What happens when you hover over an edge?
Q16. What happens when you scroll on a category label?

References

[1] Jr. Cinatl J, Vogel JU, Kotchetkov R, and Wilhelm Doerr H. Oncomodulatory signals by regulatory proteins encoded by human cytomegalovirus: a novel role for viral infection in tumor progression. *FEMS Microbiol Rev*, 28:59–77, 2004.

[2] Charles S. Cobbs, Lualhati Harkins, Minu Samanta, G. Yancey Gillespie, Suman Bharara, Peter H. King, L. Burt Nabors, C. Glenn Cobbs, and William J. Britt. Human cytomegalovirus infection and expression in human malignant glioma. *Cancer Research*, 62(12):3347–3350, 2002.

[3] Maher K Gandhi and Rajiv Khanna. Human cytomegalovirus: clinical aspects, immune regulation, and emerging treatments. *The Lancet Infectious Diseases*, 4(12):725–738, 2004.

[4] Lualhati Harkins, Andrea L Volk, Minu Samanta, Ivan Mikolaenko, William J Britt, Kirby I Bland, and Charles S Cobbs. Specific localisation of human cytomegalovirus nucleic acids and proteins in human colorectal cancer. *The Lancet*, 360(9345):1557–1563, 2002.

[5] Ellen V. Langemeijer, Erik Slinger, Sabrina de Munnik, Andreas Schreiber, David Maussang, Henry Vischer, Folkert Verkaar, Rob Leurs, Marco Siderius, and Martine J. Smit. Constitutive β-catenin signaling by the viral chemokine receptor US28. *PLoS ONE*, 7(11):e48935, 11 2012.

[6] David Maussang, Ellen Langemeijer, Carlos P. Fitzsimons, Marijke Stigter-van Walsum, Remco Dijkman, Martin K. Borg, Erik Slinger, Andreas Schreiber, Detlef Michel, Cornelis P. Tensen, Guus A.M.S. van Dongen, Rob Leurs, and Martine J. Smit. The human cytomegalovirusencoded chemokine receptor US28 promotes angiogenesis and tumor formation via cyclooxygenase-2. *Cancer Research*, 69(7):2861–2869, 2009.

[7] C Söderberg-Nauclér. Does cytomegalovirus play a causative role in the development of various inflammatory diseases and cancer? *Journal of Internal Medicine*, 259(3):219–246, 2006.