Integrative analyses reveal novel strategies in HPV11,-16 and -45 early infection

Bogumil Kaczkowski[a*], Maria Rossing[b], Ditte K. Andersen[c], Anita Dreher[c], Marya Morevati[c], Melissa A. Visser[c], Ole Winther[a,d], Finn Cilius Nielsen[b], Bodil Norrild[c*]

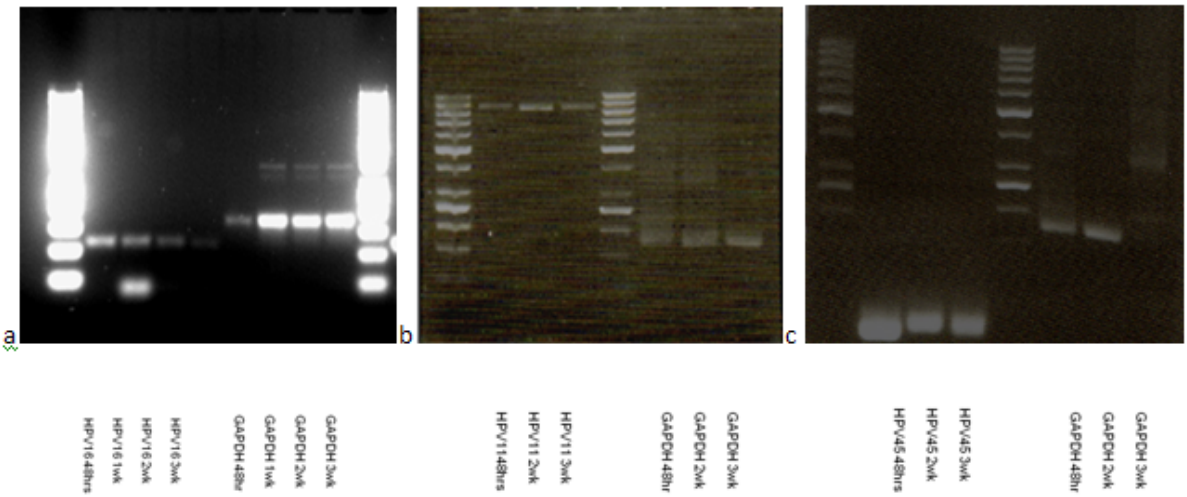
a) The Bioinformatics Centre, Department of Biology and Biomedical Research and Innovation Centre, Copenhagen University, Ole Maaloes Vej 5, 2200 Copenhagen, Denmark

b) Department of Clinical Biochemistry, Copenhagen University Hospital, Blegdamsvej 5, 2100 Copenhagen, Denmark

c) Institute of Cellular and Molecular Medicine, DNA Tumor Virus Laboratory, University of Copenhagen, Panum Institute, Blegdamsvej 3, 2200 Copenhagen, Denmark

d) DTU Informatics, Technical University of Denmark, 2800 Lyngby, Denmark

Supplement 1



Supplementary Fig.1 Analysis of HPV DNA in cells transfected with circularized genomes together with the selection marker pSV2Neo.

A. **DNA HPV16 extracted after 48hrs, 1wk, 2wk and 3wk selection with GAPDH as a control.**
B. **HPV11 extracted after 48hrs, 2wk and 3wk selection with GAPDH as a control.**
C. **HPV45 extracted after 48hrs, 2wk and 3wk selection with GAPDH as a control**

Methods

DNA extraction: After selection with 400µg/ml of G418 sulphate solution (50mg/ml) (Fisher Scientific-P02-012) with the trizol DNA purification protocol(Invitrogen)
PCR was performed with 250ng of DNA, mastermix and hot star taq polymerase(Hot star taq DNA polymerase kit, Qiagen-203205)
Primers: used (100pmol/µl):

A. HPV16 sense: 5'-nt.421 tcaaaagccactgtgtcctg nt.440-3', HPV16 antisense: 5'-nt.540 cgtgttcttgatgatctgca nt.521-3' (primers from Young et al., 1989)
B. HPV11 NCR sense: 5'- nt.7277tatatgtgtgtcagtgtg nt.7294-3', HPV11 antisense: 5'-nt.107actttccataatgcctcg nt.90-3', (HPV11 genome sequence from Dartmann, Schwarz, Gissmann, & Hausen, 1986)
C. HPV45 NCR sense: 5'-nt.7123 cgtgtacgtatacgtag nt.7139-3', HPV45 antisense:5'-nt.110 gcgcgccatcctg nt.98-3', GAPDH fwd: 5'-tgcaccaccaactgcttag-3',GAPDH rev: 5'-gatgcagggatgatgttc-3'.

PCR program: step 1, 15min/95ºC, step 2: 1min 95ºC, step 3: 1min/60ºC (HPV16 and HPV11) or 50ºC (GAPDH), step 4: 2min/72ºC, step 4: 35 cycles(step2-4), step 5: 10min/72ºC, step 6: 4ºC

The samples were analyzed on a 2,5% agarose gel.

References:

Dartmann, K., Schwarz, E., Gissmann, L., & Hausen, zur, H. (1986). The nucleotide sequence and genome organization of human papilloma virus type 11. *Virology*, *151*(1), 124–130.

Young, L. S., Bevan, I. S., Johnson, M. A., Blomfield, P. I., Bromidge, T., Maitland, N. J., & Woodman, C. B. (1989). The polymerase chain reaction: a new epidemiological tool for investigating cervical human papillomavirus infection. *BMJ (Clinical research ed.)*, *298*(6665), 14–18.

Supplement 4

Sweave report to reproduce the results of:

Integrative analyses reveal novel strategies in HPV11,-16 and -45 early infection

Supplement 4 for: Integrative analyses reveal novel strategies in HPV11,-16 and -45 early infection
Sweave report to reproduce the results

Bogumil Kaczkowski, Maria Rossing, Ditte Andersen, Anita Dreher, Melissa A. Visser, Ole Winther, Finn Cilius Nielsen, Bodil Norrild

April 10, 2012

Contents

1 Pre-processing of Affy mRNA data.

The following document contains the R code which enable to reproduce all the results and pictures that were generated in R enviroment. I used Sweave to incorporate R code in the LaTeX document. Any questions and comments as well as requests for Sweave source code (.Rnw file) and R should be directed to b.kaczkowski@gmail.com.

1.1 Reading in the CEL files and normalization

The affy mRNA expression data are loaded and normalized using OLIGO package. The names of samples are extracted from file names.

```
> library(oligo)
> geneCELs <- list.celfiles("cel_files", full.names = TRUE)
> cel_names = sub("-HuGene-1_0-st-v1-a1.CEL", "", geneCELs)
> cel_names = sub("cel_files/", "", cel_names)
```

[⊠] The results that were generated in R enviroment.

```
> cel_names = sub("-MRos", "", cel_names)
> cel_names = substr(cel_names, 13, 20)
> o = order(cel_names)
> cel_names = cel_names[o]
> geneCELs = geneCELs[o]
> affyGeneFS <- read.celfiles(geneCELs, sampleNames = cel_names)
```

```
Reading in : cel_files/FCNie-RH-01-HPV11-1-HuGene-1_0-st-v1-a1.CEL
Reading in : cel_files/FCNie-RH-02-HPV11-2-HuGene-1_0-st-v1-a1.CEL
Reading in : cel_files/FCNie-RH-03-HPV11-3-HuGene-1_0-st-v1-a1.CEL
Reading in : cel_files/FCNie-RH-04-HPV16-1-HuGene-1_0-st-v1-a1.CEL
Reading in : cel_files/FCNie-RH-05-HPV16-2-HuGene-1_0-st-v1-a1.CEL
Reading in : cel_files/FCNie-RH-06-HPV16-3-HuGene-1_0-st-v1-a1.CEL
Reading in : cel_files/FCNie-RH-07-HPV45-1-HuGene-1_0-st-v1-a1.CEL
Reading in : cel_files/FCNie-RH-09-HPV45-3-HuGene-1_0-st-v1-a1.CEL
Reading in : cel_files/FCNie-RH-10-pSVneo-1-HuGene-1_0-st-v1-a1.CEL
Reading in : cel_files/FCNie-RH-11-pSVneo-2-HuGene-1_0-st-v1-a1.CEL
Reading in : cel_files/FCNie-RH-12-pSVneo-3-HuGene-1_0-st-v1-a1.CEL
```

```
> geneCore <- rma(affyGeneFS, target = "core")
```

```
Background correcting
Normalizing
Calculating Expression
```

1.2 Annotating the probes

Now the cryptic affy probe names are substituted with universal PPI IDs (the IDs recognized by BioNet, which consist the HUGO gene names and ENTREZIDs) First we get the Hugo symbols and ENTREZIDs and create PPI IDs from it.

```
> library(hugene10sttranscriptcluster.db)
> affy_ID_to_gene = cbind(toTable(hugene10sttranscriptclusterSYMBOL),
+     toTable(hugene10sttranscriptclusterENTREZID))
> summary(affy_ID_to_gene[, 1] == affy_ID_to_gene[, 3])
```

```
   Mode    TRUE    NA's
logical   21995       0
```

```
> affy_ID_to_gene[, 5] = paste(affy_ID_to_gene$symbol, "(", affy_ID_to_gene$gene_id,
+     ")", sep = "")
> colnames(affy_ID_to_gene)[5] = "PPI_ID"
> affy_ID_to_gene = affy_ID_to_gene[order(affy_ID_to_gene$probe_id),
+     ]
```

Let's see how the affy ID to gene annotation looks:

```
> library(xtable)
> xtable(affy_ID_to_gene[1:5, ])
```

	probe_id	symbol	probe_id.1	gene_id	PPI_ID
1	7896740	OR4F17	7896740	81099	OR4F17(81099)
2	7896742	SEPT14	7896742	346288	SEPT14(346288)
3	7896744	OR4F16	7896744	81399	OR4F16(81399)
4	7896754	LOC100287934	7896754	100287934	LOC100287934(100287934)
5	7896756	FAM87A	7896756	157693	FAM87A(157693)

Now we substitute the affy IDs to PPI IDs

```
> library(affy)
> messeger_exp = exprs(geneCore)
> dim(messeger_exp)

[1] 33297    11

> messeger_exp = messeger_exp[rownames(messeger_exp) %in% as.character(affy_ID_to_gene$probe_id),
+     ]
> messeger_exp = messeger_exp[order(rownames(messeger_exp)), ]
> summary(rownames(messeger_exp) == as.character(affy_ID_to_gene$probe_id))

   Mode    TRUE    NA's
logical   21995       0

> rownames(messeger_exp) = affy_ID_to_gene$PPI_ID
> summary(duplicated(rownames(messeger_exp)))

   Mode   FALSE    TRUE    NA's
logical   19983    2012       0
```

Let's see how our expression data matrix looks like

```
> library(xtable)
> xtable(messeger_exp[1:5, 1:6])
```

	HPV11-1	HPV11-2	HPV11-3	HPV16-1	HPV16-2	HPV16-3
OR4F17(81099)	4.09	4.04	4.18	4.18	4.23	4.16
SEPT14(346288)	9.11	8.86	9.28	8.23	8.41	8.82
OR4F16(81399)	6.41	6.39	6.79	6.15	6.54	6.16
LOC100287934(100287934)	9.60	9.80	10.07	8.49	8.92	8.70
FAM87A(157693)	6.40	6.30	6.19	6.14	6.28	6.08

We see that some of the genes are represented my multiple probes.. we choose the one with highest variance and remove the rest. [1]

```
> o = order(apply(messeger_exp, 1, var), decreasing = TRUE)
> messeger_exp = messeger_exp[o, ]
> messeger_exp = messeger_exp[!duplicated(rownames(messeger_exp)),
+     ]
> summary(duplicated(rownames(messeger_exp)))

   Mode   FALSE    NA's
logical   19983       0
```

In the end we get 19983 genes , each represented by one unique probe.

1.3 Exporting the normalized expression data

Let's export the data for further analysis

```
> dim(messeger_exp)

[1] 19983    11

> write.csv(messeger_exp, file = "HPV_messeger_exp.csv")
```

[1] This is indeed what BioNet does with its *mapByVar* function , however the *mapByVar* did not work for our array platform : HuGene-1-0-st-v1 , so I did it myself

1.4 Session Info

The information about version of R and packages that were used in the analysis, which may be useful to reproduce the exact results.

```
> print(sessionInfo(), locale = FALSE)

R version 2.13.0 (2011-04-13)
Platform: x86_64-apple-darwin9.8.0/x86_64 (64-bit)

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
 [1] affy_1.30.0                          xtable_1.5-6
 [3] hugene10sttranscriptcluster.db_7.0.1 org.Hs.eg.db_2.5.0
 [5] AnnotationDbi_1.14.1                 pd.hugene.1.0.st.v1_3.2.0
 [7] RSQLite_0.9-4                        DBI_0.2-5
 [9] oligo_1.16.2                         preprocessCore_1.14.0
[11] oligoClasses_1.14.0                  Biobase_2.12.1

loaded via a namespace (and not attached):
[1] Biostrings_2.20.1 IRanges_1.10.4    affxparser_1.24.0 affyio_1.20.0
[5] bit_1.1-7         ff_2.2-3          splines_2.13.0
```

2 Differential expression analysis.

2.1 Preparing the design matrix

We start by loading the expression data, as generated beforehand.
We extract the factor/type from the column names. 'pSVneo' type is renamed do 'control'.

```
> messeger_exp = data.matrix(read.csv("HPV_messeger_exp.csv", row.names = 1))
> HPV_group = substr(colnames(messeger_exp), 1, 5)
> HPV_group[HPV_group == "pSVne"] = "control"
> HPV_group = as.factor(HPV_group)
> HPV_group

 [1] HPV11   HPV11   HPV11   HPV16   HPV16   HPV16   HPV45   HPV45   control
[10] control control
Levels: HPV11 HPV16 HPV45 control

> HPV_color = c("green", "blue", "red", "grey")[HPV_group]
```

We use the grouping of our samples to produce design matrix.

```
> HPV.design = model.matrix(~0 + HPV_group)
> colnames(HPV.design) = sub("HPV_group", "", colnames(HPV.design))
> HPV.design

   HPV11 HPV16 HPV45 control
1      1     0     0       0
2      1     0     0       0
3      1     0     0       0
4      0     1     0       0
5      0     1     0       0
6      0     1     0       0
7      0     0     1       0
8      0     0     1       0
9      0     0     0       1
10     0     0     0       1
11     0     0     0       1
attr(,"assign")
[1] 1 1 1 1
attr(,"contrasts")
attr(,"contrasts")$HPV_group
[1] "contr.treatment"
```

2.2 Differential expression analysis with *limma*

We use *limma* to fit linear model to each gene and calculate differential expression statistics.

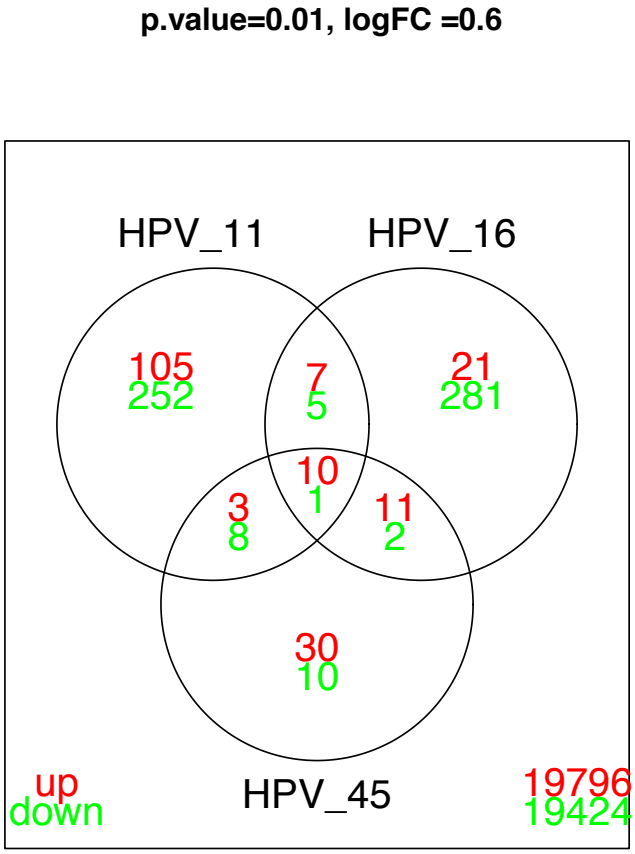
```
> library(limma)
> mRNA.lm <- lmFit(messeger_exp, HPV.design)
> mRNA.lm = eBayes(mRNA.lm)
```

To compare each virus type versus control we fit contrasts.

```
> hpv_contrasts = makeContrasts(HPV_11 = HPV11 - control, HPV_16 = HPV16 -
+     control, HPV_45 = HPV45 - control, levels = HPV.design)
> mRNA.lm.contrasts = contrasts.fit(mRNA.lm, hpv_contrasts)
> mRNA.lm.contrasts = eBayes(mRNA.lm.contrasts)
```

Let's look at the Venn diagram of differentially expressed genes. We use p-value threshold of 0.01 (adjusted with Benjamini-Hochberg method) and fold change of absolute value above 1.5 (aka log FC = 0.6)

```
> mRNA.decide = decideTests(mRNA.lm.contrasts, adjust.method = "BH",
+     p.value = 0.01, lfc = 0.6)
> vennDiagram(mRNA.decide[, 1:3], main = " p.value=0.01, logFC =0.6",
+     include = c("up", "down"), counts.col = c("red", "green"))
```



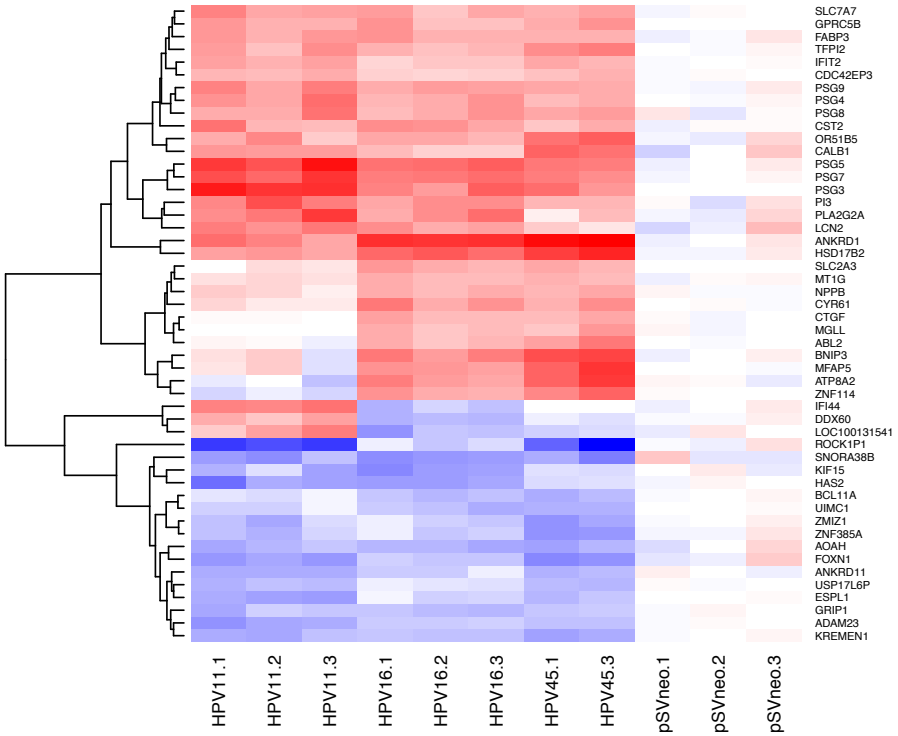
2.3 Heatmap of gene differentially expressed in at leat 2 HPV types

Now let's look at the genes that are differentially expressed in more that 1 type (in the intersections of the Venn diagram), using heatmap. Note that now, the colors in the heatmap are centered (white) on control and red indicates upregulation and blue indicates dowregulation in relation to pSVneo/control.

```
> control.mean = apply(messeger_exp[, 9:11], 1, mean)
> messeger_exp_loc_control = messeger_exp - control.mean

> library(gplots)
> x = messeger_exp_loc_control[apply(mRNA.decide[, 1:3], 1, function(x) {
+     sum(abs(x))
+ }) > 1, ]
> adjust_matrix = function(x) {
+     y = x[x > 0]
+     y = y/max(y)
```

```
+     x[x > 0] = y
+     y = x[x < 0]
+     y = -y/min(y)
+     x[x < 0] = y
+     rownames(x) = sub(pattern = "\\(.*\\)", "", rownames(x))
+     x
+ }
> heatmap(adjust_matrix(x), col = colorpanel(100, low = "blue",
+     "white", "red"), margins = c(5, 6), cexRow = 0.5, cexCol = 0.8,
+     Colv = NA, scale = "none")
```



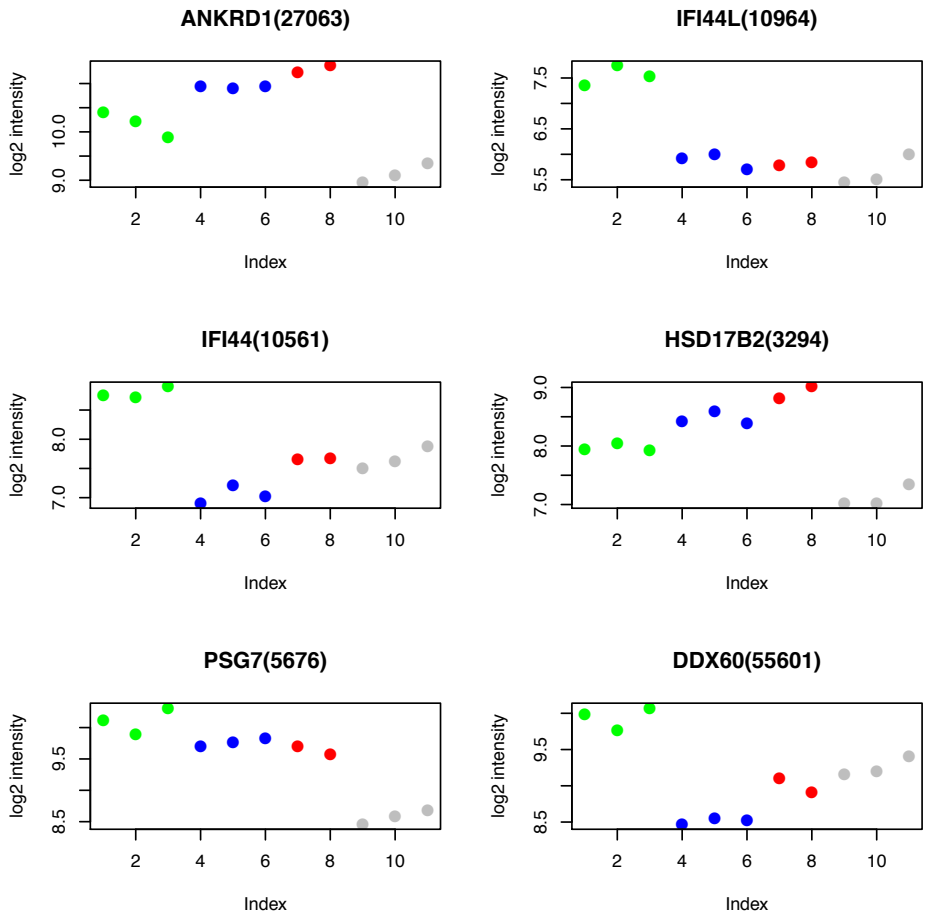
Now let's save the lists of differentially expressed genes for each virus type

```
> genes = sub(pattern = "\\(.*\\)", "", mRNA.lm.contrasts$genes$ID)
> write.csv(toptable(mRNA.lm.contrasts, coef = 1, p.value = 0.01,
+     lfc = 0.6, genelist = genes, number = 10000)[, 1:5], file = "HPV11_vs_control_pvalue.csv")
> write.csv(toptable(mRNA.lm.contrasts, coef = 2, p.value = 0.01,
+     lfc = 0.6, genelist = genes, number = 10000)[, 1:5], file = "HPV16_vs_control_pvalue.csv")
> write.csv(toptable(mRNA.lm.contrasts, coef = 3, p.value = 0.01,
+     lfc = 0.6, genelist = genes, number = 10000)[, 1:5], file = "HPV45_vs_control_pvalue.csv")
```

And let's see at the 6 interestiong genes in more details...

```
> ftest.selected = mRNA.lm.contrasts$genes$ID[rank(mRNA.lm.contrasts$F.p.value) <
+     7]
```

```
> plot_gene = function(x) {
+     plot(messeger_exp[rownames(messeger_exp) == x, ], col = HPV_color,
+         ylab = "log2 intensity", pch = 15 + batch, main = x,
+         cex = 1.5)
+ }
> par(mfrow = c(3, 2))
> for (i in 1:6) plot_gene(ftest.selected[i])
```



We need the logFC and p-value information for all genes and for all 3 viruses for protein-protein interaction analysis with BioNet. The simplest way is to export the limma generated object.

```
> save(mRNA.lm.contrasts, file = "mRNA.lm.contrasts")
```

2.4 Clustering of all differentially expressed genes

In this part, we look closer at the differentially expressed genes and look for the patterns in their expression across control and three HPV type. The gene has to be differentially expressed in at least one HPV type to be included in analysis. The differential expression thresholds are: a) absolute log2 Fold Change > 0.6 , b) adjusted p-value (FDR) < 0.01 (see the differential expression report).

The genes are clustered to find a shared pattern in gene expression. We then visualize the expression patterns by heatmap and look for biological function of each cluster using
http://www.broadinstitute.org/gsea/msigdb/annotate.jsp

```
> colors = c("black", "yellow", "darkblue", "green3", "red", "grey")
> names(colors) = 1:length(colors)
> load("mRNA.lm.contrasts")
> mRNA.lm.contrasts$adjusted.p.value = mRNA.lm.contrasts$p.value[,
+     1:3]
> mRNA.lm.contrasts$adjusted.p.value[, 1] = p.adjust(mRNA.lm.contrasts$p.value[,
+     1], method = "BH")
> mRNA.lm.contrasts$adjusted.p.value[, 2] = p.adjust(mRNA.lm.contrasts$p.value[,
+     2], method = "BH")
> mRNA.lm.contrasts$adjusted.p.value[, 3] = p.adjust(mRNA.lm.contrasts$p.value[,
+     3], method = "BH")
> mRNA.lm.contrasts$diff.pval = mRNA.lm.contrasts$adjusted.p.value <
+     0.01
> mRNA.lm.contrasts$diff.FC = abs(mRNA.lm.contrasts$coefficients[,
+     1:3]) > 0.6
> mRNA.lm.contrasts$diff.exp = mRNA.lm.contrasts$diff.pval & mRNA.lm.contrasts$diff.FC
> selected.genes <- apply(mRNA.lm.contrasts$diff.exp, 1, sum) >
+     0
> expr = data.matrix(read.csv("HPV_messeger_exp.csv", row.names = 1))
> control.mean = apply(expr[, 9:11], 1, mean)
> expr = expr - control.mean
> sel.expr = expr[selected.genes, ]
```

2.5 Visualizing the clustering of all differentially expressed genes by heatmap

Let's look at the heatmap of differentially expressed genes (see figure 1) . Pam (k-means like) clutering methods was used to divide genes into 6 groups. 1-Pearson correlation was used as distance measure.

```
> d = as.dist((1 - cor(t(sel.expr), method = "pearson"))/2)
> library(cluster)
> gene.cl = pam(d, k = 6)$clustering
> o = order(gene.cl)
> library(gplots)
> heatmap(sel.expr[o, ], Colv = NA, Rowv = NA, RowSideColors = colors[gene.cl[o]],
+     labRow = NA, col = colorpanel(100, low = "blue", mid = "white",
+         high = "red"), scale = "none")
> write.csv(gene.cl, "clustering_of_the_genes.csv")
```

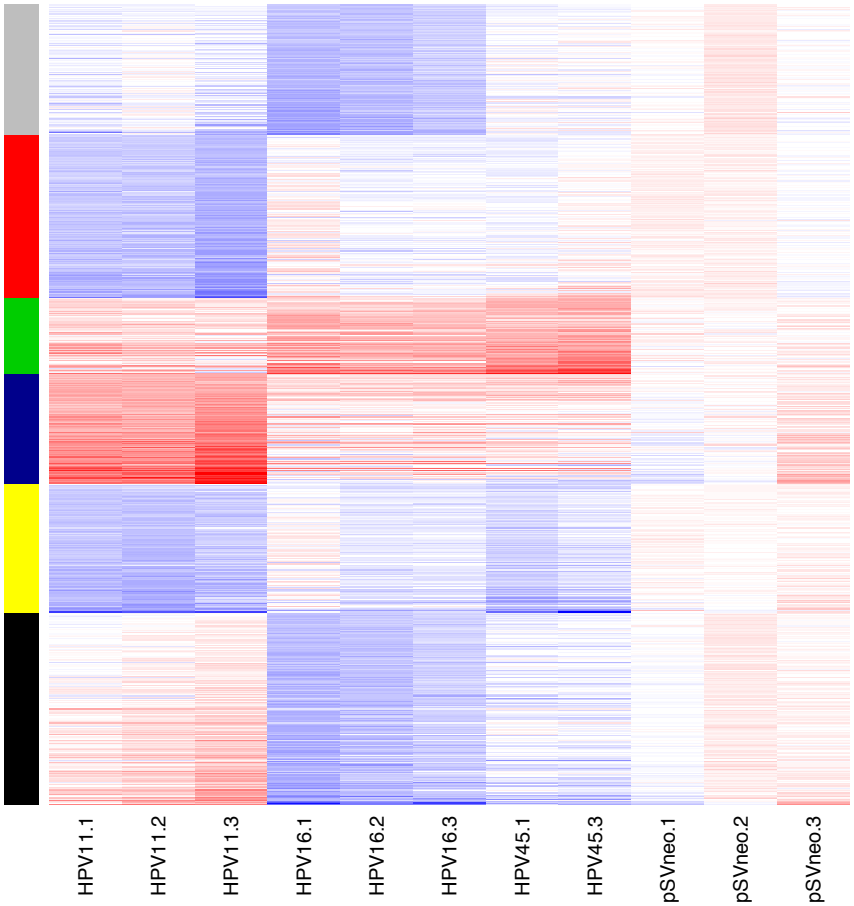


Figure 1: Heatmap of differentially expressed gene expression in HPV infected cells.

Let's check how many genes per cluster we have.

```
> table(gene.cl)

gene.cl
  1   2   3   4   5   6
179 119 102  71 151 121
```

3 Looking for differentially expressed networks of protein-protein interacting genes.

I load the the BioNet library and protein-protein interaction data.

```
> library(BioNet)
> library(DLBCL)
> data(interactome)
> load("mRNA.lm.contrasts")
> network = subNetwork(rownames(mRNA.lm.contrasts), interactome)
> network <- largestComp(network)
> network <- rmSelfLoops(network)
```

I reduce the protein-protein interaction network to the proteins present on the chip we use for mRNA profiling and clean the network a bit using *largestComp()* and *rmSelfLoops()* functions.

3.1 HPV-11

Now the p-values from differential expression analysis are used to calculate the differentially expressed modules i.e. group of genes that interact at protein levels that are differentially expressed. The networks are exproted in XGMML and visualized in Cytoscape.

```
> pval_hpv11 = mRNA.lm.contrasts$p.value[, 1]
> logFC_hpv11 <- mRNA.lm.contrasts$coefficients[, 1]
> fb_hpv11 <- fitBumModel(pval_hpv11, plot = FALSE)
> fb_hpv11

Beta-Uniform-Mixture (BUM) model

19983 pvalues fitted

Mixture parameter (lambda):        0.018
shape parameter (a):                 0.371
log-likelihood:                        13662.9

> scores_hpv11 <- scoreNodes(network = network, fb = fb_hpv11,
+     fdr = 0.001)
> module_hpv11 <- runFastHeinz(network, scores_hpv11)
> hpv11NetworkGenes = module_hpv11@nodes
> nodeDataDefaults(module_hpv11, attr = "diff") <- ""
> nodeData(module_hpv11, n = nodes(module_hpv11), attr = "diff") <- logFC_hpv11[nodes(module_hpv11)]
> saveNetwork(module_hpv11, file = "module_hpv11", type = "XGMML")

[1] "...adding nodes"
[1] "...adding edges"
[1] "...writing to file"
```

Ok lets look at how our network/module looks like ...

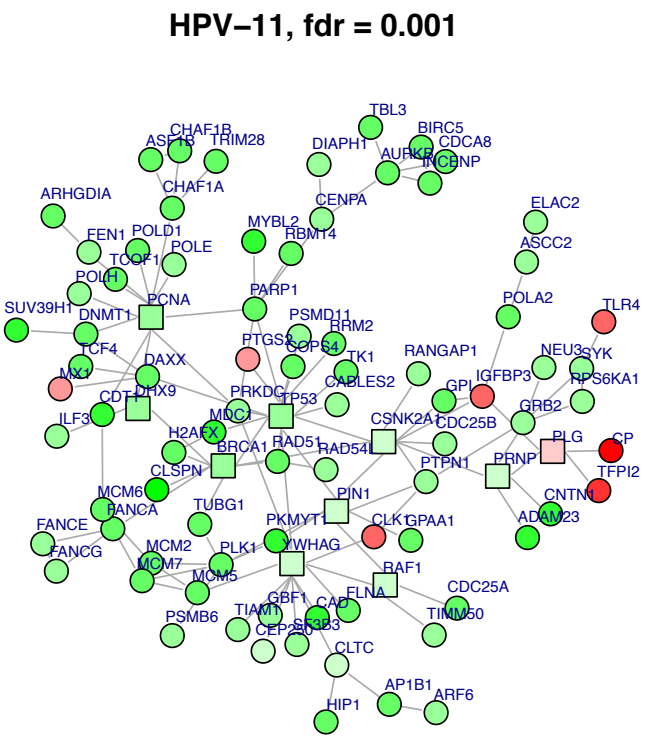


Figure 2: The differentially expressed module in HPV11, using false discovery rate = 0.001, nodes in red represent upregulated genes and nodes in green downregulated genes.

3.2 HPV-16

Now the p-values from differential expression analysis of HPV-16 are used to calculate the differentially expressed modules i.e. group of genes that interact at protein levels that are differentially expressed.

```
> pval_hpv16 = mRNA.lm.contrasts$p.value[, 2]
> logFC_hpv16 <- mRNA.lm.contrasts$coefficients[, 2]
> fb_hpv16 <- fitBumModel(pval_hpv16, plot = FALSE)
> fb_hpv16

Beta-Uniform-Mixture (BUM) model

19983 pvalues fitted

Mixture parameter (lambda):        0.000
shape parameter (a):                 0.359
log-likelihood:                        15240.3

> scores_hpv16 <- scoreNodes(network = network, fb = fb_hpv16,
+     fdr = 0.002)
> module_hpv16 <- runFastHeinz(network, scores_hpv16)
> hpv16NetworkGenes = module_hpv16@nodes
```

```
> nodeDataDefaults(module_hpv16, attr = "diff") <- ""
> nodeData(module_hpv16, n = nodes(module_hpv16), attr = "diff") <- logFC_hpv16[nodes(module_hpv16)]
> saveNetwork(module_hpv16, file = "module_hpv16", type = "XGMML")

[1] "...adding nodes"
[1] "...adding edges"
[1] "...writing to file"
```

Ok lets look at how our network/module looks like ...

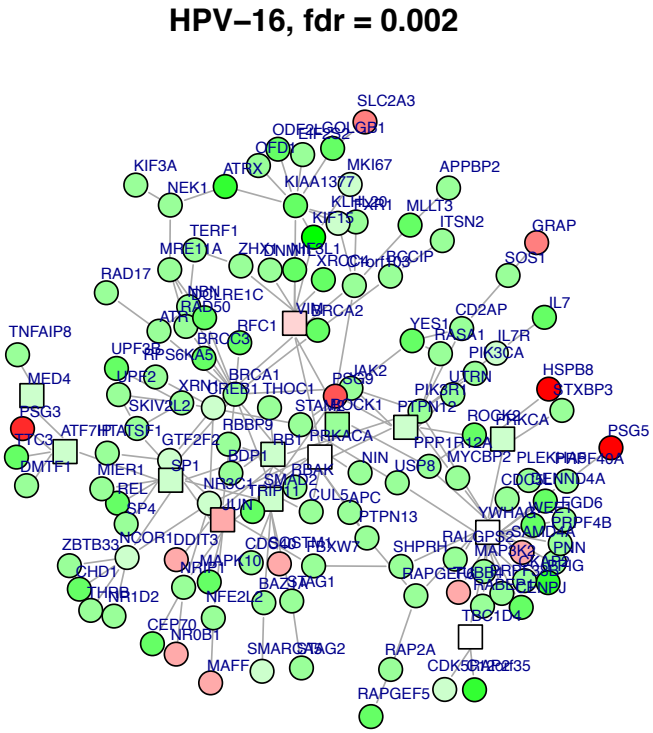


Figure 3: The differentially expressed module in HPV16, using false discovery rate = 0.002, nodes in red represent upregulated genes and nodes in green downregulated genes.

3.3 HPV-45

```
> pval_hpv45 = mRNA.lm.contrasts$p.value[, 3]
> logFC_hpv45 <- mRNA.lm.contrasts$coefficients[, 3]
> fb_hpv45 <- fitBumModel(pval_hpv45, plot = FALSE)
> fb_hpv45

Beta-Uniform-Mixture (BUM) model

19983 pvalues fitted
```

```
Mixture parameter (lambda):        0.000
shape parameter (a):                 0.528
log-likelihood:                        5108.9
```

```
> scores_hpv45 <- scoreNodes(network = network, fb = fb_hpv45,
+     fdr = 0.02)
> module_hpv45 <- runFastHeinz(network, scores_hpv45)
> hpv45NetworkGenes = module_hpv45@nodes
> nodeDataDefaults(module_hpv45, attr = "diff") <- ""
> nodeData(module_hpv45, n = nodes(module_hpv45), attr = "diff") <- logFC_hpv45[nodes(module_hpv45)]
> saveNetwork(module_hpv45, file = "module_hpv45", type = "XGMML")
```

```
[1] "...adding nodes"
[1] "...adding edges"
[1] "...writing to file"
```

Ok lets look at how our network/module looks like ...

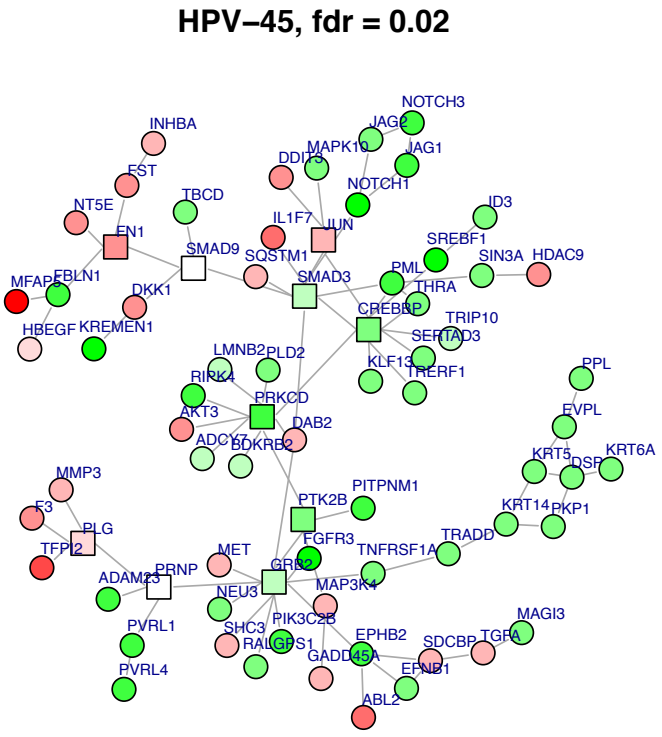


Figure 4: The differentially expressed module in HPV45, using false discovery rate = 0.02, nodes in red represent upregulated genes and nodes in green downregulated genes.

3.4 Session Info

```
> print(sessionInfo(), locale = FALSE)

R version 2.13.0 (2011-04-13)
Platform: x86_64-apple-darwin9.8.0/x86_64 (64-bit)

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] limma_3.8.2    DLBCL_1.3.1    BioNet_1.10.1  RBGL_1.28.0    graph_1.30.0
[6] Biobase_2.12.1

loaded via a namespace (and not attached):
[1] AnnotationDbi_1.14.1 DBI_0.2-5            RSQLite_0.9-4
[4] XML_3.4-0            igraph_0.5.5-2       tools_2.13.0
```