Bayesian model comparison of nested effect models for Wnt signalling analysis

Supplement to “Nested effects modelling of Wnt signalling suggests the sensitization of cancer cells to Wnt ligands in colorectal cancer”.

Giusi Moffa

```r
# global chunk options
opts_chunk$set(cache = FALSE, autodep = TRUE)
```

NEM analysis of WNT reads

In the following we provide the code for the complete analysis, which were run under R version `3.0.0` (nem version `2.36.0`, limma version `3.16.1`).

Record session information.

```r
sessionInfo()

## R version 3.0.3 (2014-03-06)
## Platform: x86_64-apple-darwin10.8.0 (64-bit)
##
## locale:
## [1] en_GB.UTF-8/en_GB.UTF-8/en_GB.UTF-8/C/en_GB.UTF-8/en_GB.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] knitr_1.6
##
## loaded via a namespace (and not attached):
## [1] evaluate_0.5.5 formatR_0.10   highr_0.3      stringr_0.6.2
## [5] tools_3.0.3
```

Install packages

```r
source("http://bioconductor.org/biocLite.R")
biocLite("Rgraphviz", type = "source")
biocLite("edgeR")
library(edgeR)
```

Data

The reads obtained by RNA-Sequencing are read into R and normalised through the `voom` function Law et al. [2014] of the `limma` package. As it is standard practice the columns of the data matrix correspond to samples and the rows to genes. The data as normalised by `voom` is then further processed for differential expression analysis with `limma`.

Loading data

From within the working folder, containing files `wntCounts.RData` and `wntExonLen.RData`.

```
load("wntCounts.RData")  ### contains a variable D.counts with the the count data
load("wntExonLen.RData")  ### contains a variable Dlength with the gene exon lengths
```

Remove short genes

Genes shorter than 150bp are removed from the analysis since these should not be theoretically present, and eventual reads probably only happen by contamination.

Select only the replicates which passed the quality control filter

```
DsubNames <- grep("_1|_2", colnames(D2analysis), value = TRUE)
DsubNames <- DsubNames[-c(14:18)]
DsubNames
DsubJ <- D2analysis[, DsubNames]
colnames(DsubJ)
dim(DsubJ)
# barplot(colSums(DsubJ)*1e-6, names=seq_along(length(DsubNames)))
```

Data transformation via voom

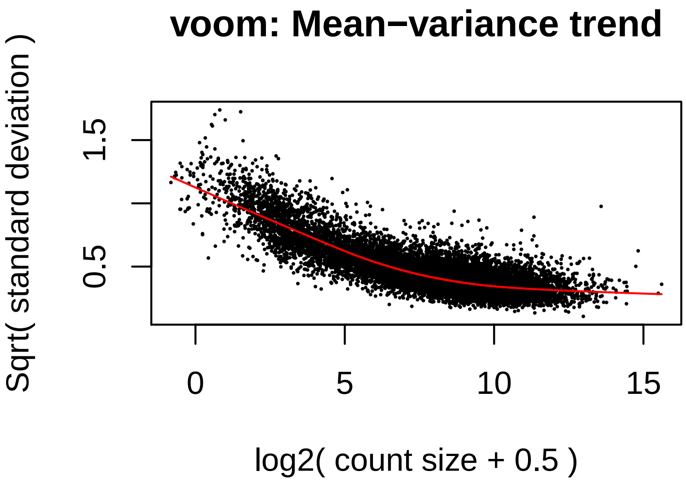
Transform the data with voom to then apply the limma pipeline for differential analysis.

```
library(limma)
library(edgeR)
isexpr <- rowSums(cpm(DsubJ) > 1) >= 1
## keep only genes with at least one count per million reads in at least one
## sample (Note that cpm are not integers)

## cpmD <- cpm(DsubJ) cpmDbool <- cpmD[which(rowSums(cpmD>1) >= 1),]
## min(rowSums(cpmDbool))

Dsel <- DsubJ[isexpr, ]
colnames(Dsel) <- colnames(DsubJ)
dim(Dsel)
pickNames <- function(x) head(unlist(strsplit(x, split = "_")), 1)
exps <- factor(unname(sapply(colnames(Dsel), pickNames)))
design <- model.matrix(~0 + exps)
colnames(design) <- make.names(gsub("exps", "", colnames(design)))
# image(t(apply(t(design), 1, rev)) ) # graphical visualization of design
# matrix
colnames(Dsel) <- unname(sapply(colnames(Dsel), pickNames))
```

```
D4nem <- voom(Dsel, design, plot = TRUE)
```



Definition of some functions which are used in the main procedure

Define a number of functions so that the same processing can be more easily applied to different selections of experiments to model.

Function to perform the differential analysis via limma

Among other statistics posterior probabilities that genes are affected by certain perturbations are provided as output. The routine comes from the package `limma` and is based on the Bayesian methods described in Smyth [2004]. The parameter `pr` gives the assumed prior probability of differential expression. In our analysis below we will set this parameter to .004, meaning that 4 genes in a thousand are a priori assumed to be differentially expressed. The prior reflects expert knowledge in that it was set in a way that the length of identified target lists roughly matches reports found in the literature.

```
applyLimma <- function(jdata, ctrl, pr = 0.01) {
    require(limma)
    cn <- colnames(jdata)
    # colnames(jdata) <- make.names(cn)
    exps <- factor(colnames(jdata))
    design <- model.matrix(~0 + exps)
    colnames(design) <- unique(cn)
    cn <- setdiff(cn, ctrl)
    fit1 <- lmFit(jdata, design)
    contrast.matrix <- makeContrasts(contrasts = paste(cn, "-", ctrl, sep = ""),
        levels = design)
    fit2 <- contrasts.fit(fit1, contrast.matrix)
    WNTfit <- eBayes(fit2, pr)
    res <- list(WNTfit, names = cn)
    return(res)
}
```

Scoring function for nem models

The scores are evaluated on the basis of continuous data (the posterior probabilities) according to the likelihood expression defined in the main text, at the end of the section *Quantifying the evidence for NEM structural features via Bayes factors*

```
netScore <- function(jPhi, jD) {
    if (!all(diag(jPhi) == 1))
        diag(jPhi) <- 1
    numbSgene <- ncol(jD)
    Escore <- exp(log(jD) %*% jPhi + log(1 - jD) %*% (1 - jPhi))
    jScRes <- sum(log(rowSums(Escore)/numbSgene))
}
```

Evaluation of Bayes factors between different topology classes

```
fullSimpleBF <- function(sNew, sStd) {
    sCor <- log(length(sStd)) - log(length(sNew))
    # size correction (accounts for different number of models/complexity, a
    # typical feature of Bayes Factors
    Mn <- max(sNew)
    Ms <- max(sStd)
    logBF <- sCor + Mn - Ms + log(sum(exp(sNew - Mn))) - log(sum(exp(sStd -
        Ms)))
    c(logBF = logBF, BF = exp(logBF))
}
```

Perform differential expression analysis

Obtain the posterior probabilities for downstream effects via limma analysis

```
subJdata <- D4nem
subJlim <- applyLimma(subJdata, ctrl = "CTRL", pr = 0.004)
subJodd <- exp(subJlim[[1]]$lods)  # log-odds
colnames(subJodd) <- subJlim$names
subJodd <- subJodd/(1 + subJodd)
# posterior probabilities of differential expression
postProbDE <- subJodd
```

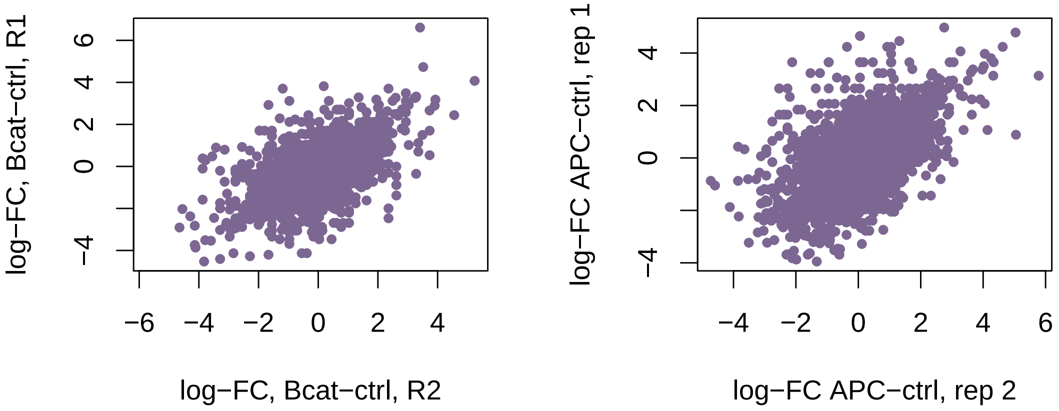
```
# global chunk options
opts_chunk$set(cache = FALSE, autodep = TRUE)
```

Make a log-fold change reproducibility plot from the log count per million reads for β-catenin and ”APC” with respect to CTRL. It is similar to what obtained from the voom transformed data `D4nem$E`

```
b <- ((1:4) * 2 - 1)/7
r <- b * 0.85
g <- b * 0.7
jpurp <- rgb(4 * 0.85/7, 4 * 0.7/7, 4/7)
```

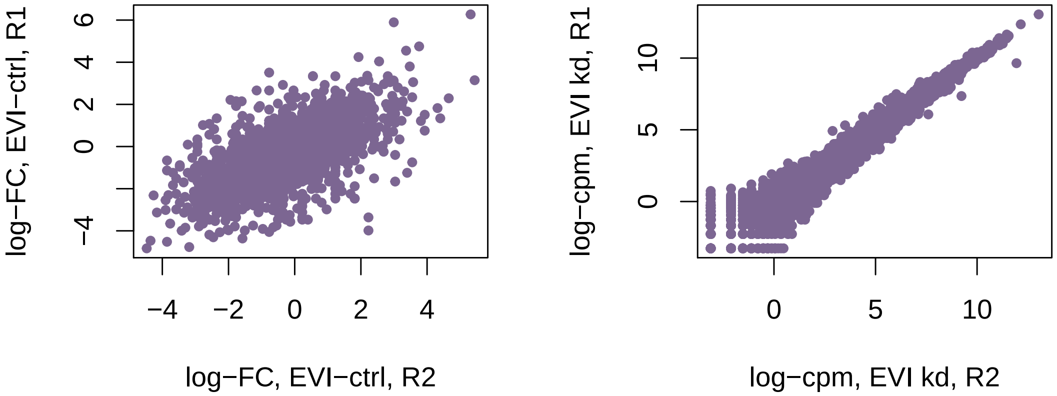
```
jpurp2 <- rgb(5 * 0.85/7, 5 * 0.7/7, 5/7)
cpmD <- cpm(DsubJ)
cpmDlog <- log2(cpmD[isexpr, ])
```

```
par(mfrow=c(1,2), cex=1.2)
plot(cpmDlog[,"CTNNB1_1"]-cpmDlog[,"CTRL_1"],
     cpmDlog[,"CTNNB1_2"]-cpmDlog[,"CTRL_2"],
     ylab="log-FC, Bcat-ctrl, R1",
     xlab="log-FC, Bcat-ctrl, R2",
     pch=20, col=jpurp)
plot(cpmDlog[,"APC_1"]-cpmDlog[,"CTRL_1"],
     cpmDlog[,"APC_2"]-cpmDlog[,"CTRL_2"],
     ylab="log-FC APC-ctrl, rep 1",
     xlab="log-FC APC-ctrl, rep 2",
     pch=20, col=jpurp)
```



Reproducibility of EVI and its fold changes with respect to the CTRL

```
par(mfrow=c(1,2), cex=1.2)
plot(cpmDlog[,"EVI_1"]-cpmDlog[,"CTRL_1"],
     cpmDlog[,"EVI_2"]-cpmDlog[,"CTRL_2"],
     ylab="log-FC, EVI-ctrl, R1",
     xlab="log-FC, EVI-ctrl, R2",
     pch=20, col=jpurp)
plot(cpmDlog[,"EVI_1"], cpmDlog[,"EVI_2"],
     ylab="log-cpm, EVI kd, R1",
     xlab="log-cpm, EVI kd, R2",
     pch=20, col=jpurp)
```



Plot the siRNA efficiencies observed in the data

```
knockdown <- c("CTNNB1", "EVI", "APC", "TCF7L2")
geneTarget <- c("CTNNB1", "WLS", "APC", "TCF7L2") ### WLS is the EVI genen symbol
genePick <- c(geneTarget,"AXIN2", "ACTB")
postProbDE[genePick, knockdown]

##         Contrasts
##              CTNNB1       EVI      APC    TCF7L2
##   CTNNB1 1.0000000 0.0108482 0.026336 0.0003116
##   WLS    0.9093016 0.9999991 0.001961 0.0397116
##   APC    0.0009051 0.5449705 0.999917 0.9853499
##   TCF7L2 0.9997415 0.5592880 0.998739 0.9999973
##   AXIN2  0.9986680 0.9897556 0.999897 0.0019609
##   ACTB   0.0015294 0.0005056 0.323049 0.0074923

knockEff <- topTable(subJlim[[1]], adjust="BH",
                     n=length(postProbDE[,1]))[genePick,]
knockNames <- paste0(knockdown,".","CTRL")
knockObs <- cbind(geneTarget, knockNames)
knockPlot <- c(1,2^as.numeric(knockEff[knockObs]))
postProbDE[geneTarget,]

##         Contrasts
##                APC     AXIN1      BCL9    CTNNB1     EVI    TCF7L2
##   CTNNB1 0.026336 0.0020398 0.0082530 1.0000000 0.01085 0.0003116
##   WLS    0.001961 0.0032370 0.0004244 0.9093016 1.00000 0.0397116
##   APC    0.999917 0.0017990 0.0019270 0.0009051 0.54497 0.9853499
##   TCF7L2 0.998739 0.0003446 0.9978946 0.9997415 0.55929 0.9999973

## effectively the posterio probabilities are estimated as being all 1
topTable(subJlim[[1]], coef=4, adjust="BH", n=3)

##         logFC AveExpr      t   P.Value adj.P.Val     B
## CTNNB1 -3.981   9.034 -30.95 7.343e-14 9.212e-10 20.80
## PDE4B  -2.047   8.095 -25.91 7.671e-13 4.812e-09 18.79
## KITLG  -2.876  10.581 -24.25 1.830e-12 7.652e-09 18.14
## coef=4 corresponds to "CTNNB1" knockdown, and actually it comes top in the effects
BetaCatRank <- topTable(subJlim[[1]], coef=4, adjust="BH",
                        n=length(postProbDE[,1]))
BetaCatRank[which(BetaCatRank$ID %in% genePick),]

## [1] logFC     AveExpr   t         P.Value   adj.P.Val B
## <0 rows> (or 0-length row.names)
```

vspace-3ex

```
par(cex=1.2)
barplot(knockPlot, width=.2, space=2, names.arg=c("CTRL",knockdown),
        xlim=c(0,3), cex.axis=1.1, axis.lty=1, ylab="knockdown efficiencies",
        col=jpurp)
```

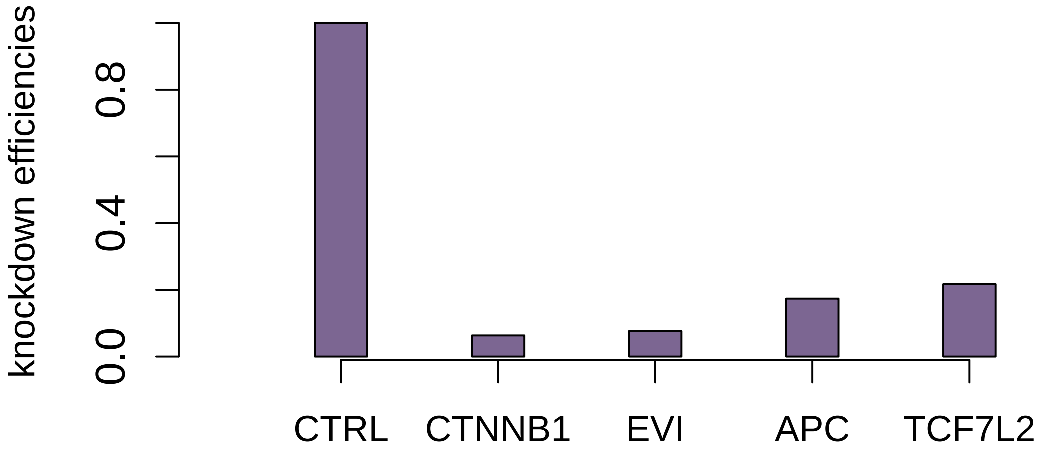


Figure 1: The posterior probabilities of the knockdown targets being differentially expressed are effectively estimated all as being 1.

Plot effects observed in the data on typical wnt targets

```
# moreBetaTargets <- c("MET", "EDN1", "HES1", "LAMC2", "PLAU", "SP5")
knockdown <- c("CTNNB1", "EVI", "APC", "TCF7L2")
wntTargets <- c("AXIN2", "RUNX2", "SMAD7")
postProbDE[wntTargets, knockdown]

##        Contrasts
##          CTNNB1    EVI    APC   TCF7L2
##   AXIN2 0.99867 0.9898 0.9999 0.001961
##   RUNX2 0.02812 0.1168 0.1843 0.999737
##   SMAD7 0.38008 0.5950 0.7904 0.000694

wntEff <- topTable(subJlim[[1]], adjust="BH",
                      n=length(postProbDE[,1]))[wntTargets,]
knockNames <- paste0(knockdown,".","CTRL")
wntObs <- cbind(rep(wntTargets, each=length(knockdown)),
                rep(knockNames, length(wntTargets)) )
# knockCol <- rainbow(length(knockdown), start=.6, end=1, alpha = .4)
knockCol <- rgb(r,g,b)
names(knockCol) <- knockdown
```

```
# global chunk options
opts_chunk$set(cache = FALSE, autodep = TRUE)
```

Define models and classes

Define a function building the nem models and the topology classes to consider, it defines the node names, enumerates all the models and selects those satisfying the desired contraints encoded by `rnaiDisc`.

```
buildTopologyClasses <- function(postPde, iOut, iDisc) {
    # Define node names corresponding to the knockdowns to model and select
    # corresponding data
    nodeNames <- unique(colnames(postPde)[which(!colnames(postPde) %in% iOut)])
    postPde <- postPde[, which(colnames(postPde) %in% nodeNames)]
    nNode <- length(nodeNames)
    nodeNames

    ### Enumerate all models
    StdMods <- enumerate.models(nodeNames, verbose = TRUE)
    numbStd <- length(StdMods)
    StdModels <- list(AdjMats = StdMods, numb = numbStd, Names = nodeNames)

    ### Define the set of constrained models, with no links between given nodes
    ### (for example ``EVI, APC'' on one side and ``CTNNB1, TCF7L2'' on the other)
    E1 <- which(StdModels$Names %in% iDisc)
    E2 <- which(!StdModels$Names %in% iDisc)
    stdEl <- c()
    ### A numerical vector with the indeces of the models satisfying the
    ### contraints
    for (i in 1:StdModels$numb) {
        if (!(sum(StdModels$AdjMats[[i]][E2, E1]) + sum(StdModels$AdjMats[[i]][E1,
            E2])))
            stdEl <- c(stdEl, i)
    }
    list(allModels = StdModels, constModels = stdEl, selData = postPde)
}
```

Evaluate Bayes factors between selected models

Define a function evaluating the Bayes factors for the selected data according to the definition in equation (**??**), between the classes of topologies corresponding to the activation by sensitization

```
layout(matrix(c(1,2)), heights=c(1,3))
par(mar = c(1,4,2,4), cex=1.5)
mid_bar <- barplot(t(postProbDE[wntTargets, knockdown]),
        width=.1, space=c(4,14), beside=TRUE,
        col="white", xlim=c(0,11),
        legend.text=knockdown,
        args.legend=list(x=12, y=.9, fill=knockCol, bty="n"),
        ylab="pp DE",
        border=NA, xaxt="n")
segments(as.vector(mid_bar)-.2, rep(0, length(mid_bar)),
         as.vector(mid_bar)-.2, as.vector(t(postProbDE[wntTargets, knockdown])),
         col=knockCol, lwd=5, lend=2)
# abline(h=.5, col=rgb(1, .4, 0), lwd=2, lty=2)
segments(-1, .5, 9, .5, lwd=2, lty=2, col=rgb(1, .4, 0))
abline(h=0)
par(mar = c(4,4,1,4))
barplot(t(as.matrix(wntEff[,knockNames])),
        width=.5, space=c(0,2), beside=TRUE,
        col=knockCol, xlim=c(0,11), xpd=FALSE,
        ylab="Log Fold Change",
#        legend.text=knockdown,
#        args.legend=list(x=17, y=-1, fill=knockCol, bty="n"),
        names.arg=wntTargets)
abline(h=0)
```

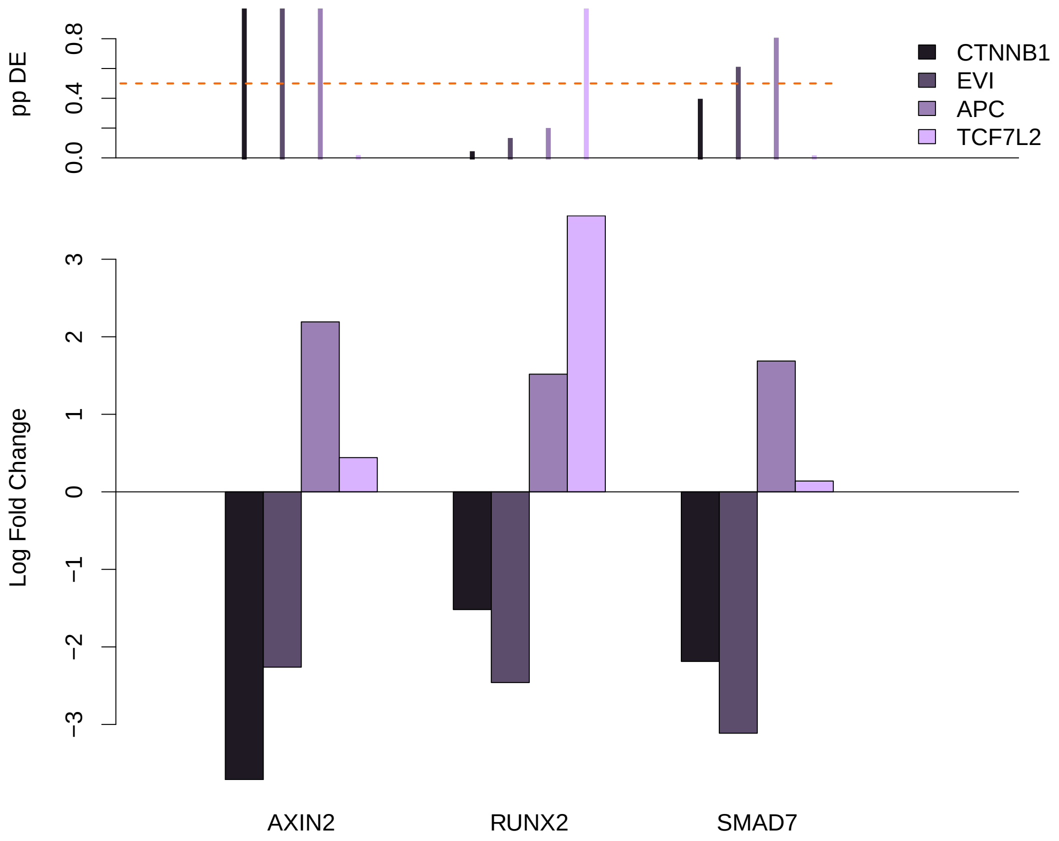


Figure 2: Fold changes observed on well known typical wnt targets and their posterior probability of differential expression.

and the direct activation models. When considering *EVI, APC, CTBNN1* and *TCF7L2* for example no links are allowed in the direct activation model between the subnets including the pairs *EVI, APC* and *CTBNN1, TCF7L2* respectively. Bayes factors are calculated between a topology class and its complement, rather than between classes included in each other, as the cut-off for the probability of a downstream effect (or differential expression) increases, in a continuous data approach.

```
dataBFtopologyClasses <- function(postPde, allModels, constModels) {
    ### Define some working variables
    myp <- seq(0.01, 0.99, 0.01)
    # posterior probability cut-off for differential expression
    listScores <- list()  # all scores
    subScores <- list()  # scores of constrained models, a subset of the whole set
    nG <- c()  # number of genes selected for each cut-off
    vecBF <- c()

    for (tmpp in myp) {
        tmp_sel <- which(apply(postPde > tmpp, 1, any))
        D_sel <- postPde[tmp_sel, ]
        nG <- c(nG, dim(D_sel)[1])
        tmpScore <- sapply(allModels$AdjMats, netScore, D_sel)
        listScores <- c(listScores, list(tmpScore))
        tmpSub <- tmpScore[constModels]
        subScores <- c(subScores, list(tmpSub))
        vecBF <- c(vecBF, fullSimpleBF(tmpScore[-constModels], tmpSub))
    }
    matBF <- matrix(vecBF, ncol = 2, byrow = T)
    matBF <- cbind(matBF, myp)
    colnames(matBF) <- c("logBF", "BF", "podd")
    list(allScores = listScores, constScores = subScores, BayesFactors = matBF,
        nSelGenes = nG, cutoffP = myp)
}
```

Selection of knockdown experiments modelled in the analysis

The nem analysis is performed including 4 knockdown experiments in the modelling, namely EVI, APC, TCF7L2 and CTNNB1.

```
library(nem)
rnaiOut <- c("AXIN1", "BCL9", "CTRL")
## set of rnai to leave out
rnaiDisc <- c("EVI", "APC")
## set of rnai assumed disconnected from the others (CTNNB1 and TCF7L2)
```

NEM analysis

Perform nem analysis and evaluate Bayes factors

```
topClasses <- buildTopologyClasses(subJodd, rnaiOut, rnaiDisc)

## Generated 355 unique models ( out of 4096 )
```

```r
topClassBF <- dataBFtopologyClasses(topClasses$selData,
                                    topClasses$allModels,
                                    topClasses$constModels)
```

Plot Bayes factors

Plot the log Bayes factors between the two classes of topologies corresponding to activation by sensitization and direct activation model as a function of the cut-off posterior probabilities of differential expression.

```r
library(plotrix)
matBFplot <- topClassBF$BayesFactors[, 1]
matBFplot <- topClassBF$BayesFactors[, 1]
biggerHalf <- which(topClassBF$cutoffP > 0.485)
matBFplot <- topClassBF$BayesFactors[biggerHalf, 1]
cutsPlot <- topClassBF$cutoffP[biggerHalf]
```

Plot genes with highest posterior probability of differential expression in at least one intervention

Visualize in a quilt plot the posterior probability of differential expression for the genes which have a posterior probability larger than .5 of showing an effect in at least one of the knockdown experiments included in the model, order by the minimum between the probabilities of differential expression in the knockdown of EVI, CTNNB1, TCF7L2 and the probability of no differential expression in the knockdown of APC.

```r
library(fields)
library(RColorBrewer)
```

Gene ordering.

```r
# set of rnai's modelled
subNet1 <- c("EVI", "APC")
subNet2 <- c("CTNNB1", "TCF7L2")
modelPde <- topClasses$selData[,c(subNet1, subNet2)]
modelPde <- modelPde[which(apply(modelPde>.5, 1, any )),]
ordMatrix <- modelPde
ordMatrix[,"APC"] <- 1-ordMatrix[,"APC"]
minProbDE <- apply(ordMatrix,1,min)
orderPs <- order(minProbDE, decreasing=TRUE)
nrcolors <- 100
half <- 1+nrcolors/2
colpal <- c(brewer.pal(9, "Blues")[9:1], brewer.pal(9,"Reds")[1:9])
colorpalette <- colorRampPalette(colpal)(nrcolors)
quantileBreaks <- c(quantile(postProbDE[which(postProbDE < .5)],
                             probs = seq(0, 1, length = half)),
                    quantile(postProbDE[which(postProbDE >= .5)],
                             probs = seq(0, 1, length = half-1)))
```

```r
par(cex.lab=1.4, cex.axis=1.1)
plot(cutsPlot, matBFplot, type="l", col=4, lwd=3,
ylab="log Bayes Factor", xlab="Probability cutoff for downstream effects",
xaxt="n", yaxt="n", ylim=extendrange(r=range(matBFplot), f=.1))
ticks2 <- axTicks(2); ticks2 <- ticks2[which(ticks2>=0)]
axis(2, at=c(range(matBFplot)[1],ticks2),
labels=c(round(range(matBFplot)[1],2),ticks2))
axis(1, at=topClassBF$cutoffP[which(seq(1:100)%%5 == 0)],
labels=topClassBF$cutoffP[which(seq(1:100)%%5 == 0)])
axis(3, at=topClassBF$cutoffP[which(seq(1:100)%%5 == 0)],
labels=topClassBF$nSelGenes[which(seq(1:100)%%5 == 0)])
mtext("Number of Targets", side=3, line=3, cex=1.4) # margin text
abline(h=0, col="firebrick3", lwd=4)
abline(v=0.95, col="goldenrod2", lwd=3, lty=4)
abline(v=topClassBF$cutoffP[min(which(topClassBF$BayesFactors[,2]>1))],
col="orchid3", lwd=3, lty=4)
text(.01,300, pos=4, labels="Activation by Sensitization Model")
text(.01,-300, pos=4, labels="Direct Activation Model")
```

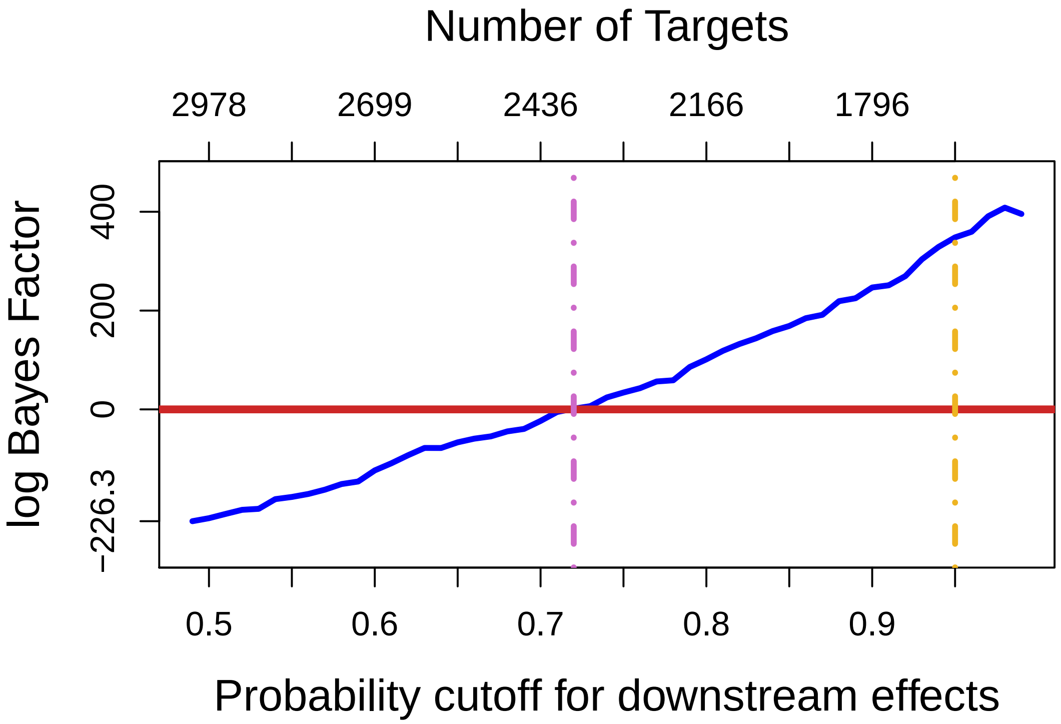


Figure 3: The blue line shows log Bayes Factors (y-axis) between the class of topologies corresponding to activation by sensitization models to the class of topologies corresponding to direct activation models. Log Bayes Factors were obtained from nested effect models for different numbers of potential target genes (x-axis, top), which were included according to a cut-off on the posterior probability that a gene is affected for at least one perturbation (among those included in the model). Positive log Bayes factor reflect evidence in favour of the activation by sensitization model. Log Bayes Factors above 10 can be considered strong evidence. The dashed purple line indicates the smallest cutoff yielding a model that favours the activation by sensitization model. A conservative cutoff of 95% posterior probability that an observed expression difference is a true response to perturbation is marked by the yellow dashed line.

```
par( mar = par( "mar" ) + c( 0, 2, 0, 4 ) )
imgOdd <- modelPde[orderPs,rev(knockdown)]
sep <- 40
sepLine <- matrix(runif(sep*4), nrow=sep, ncol=4)
colnames(sepLine) <- colnames(imgOdd) ### check that the order is right!!!
left <- 1:750
imgLeft <- imgOdd[left,]
imgRight <- imgOdd[-left,]
# imgOdd <- rbind(sepLine, imgLeft, sepLine, imgRight)
imgOdd <- rbind(imgLeft, sepLine, imgRight)
x <- 1:nrow(imgOdd)
y <- 1:ncol(imgOdd)
image(x,y,imgOdd, xaxt="n", yaxt="n", ylab="", xlab="",
col=colorpalette, breaks=quantileBreaks)
# axis( 1, labels=c(1, nrow(imgOdd)-2*sep), at=seq(1,nrow(imgOdd),length.out=2))
axis( 1, labels=c(1, nrow(imgOdd)-sep), at=seq(1,nrow(imgOdd),length.out=2))
axis( 2, at=seq(1,ncol(imgOdd),length.out=ncol(imgOdd) ),
labels= colnames( imgOdd ), las= 2 )
image.plot(imgOdd, xaxt="n", yaxt="n", legend.only=TRUE,
col=colorpalette)
colLine <- 751:(750+sep)
for(i in colLine) abline(v=i, col=rgb(.5,.9,0))
```

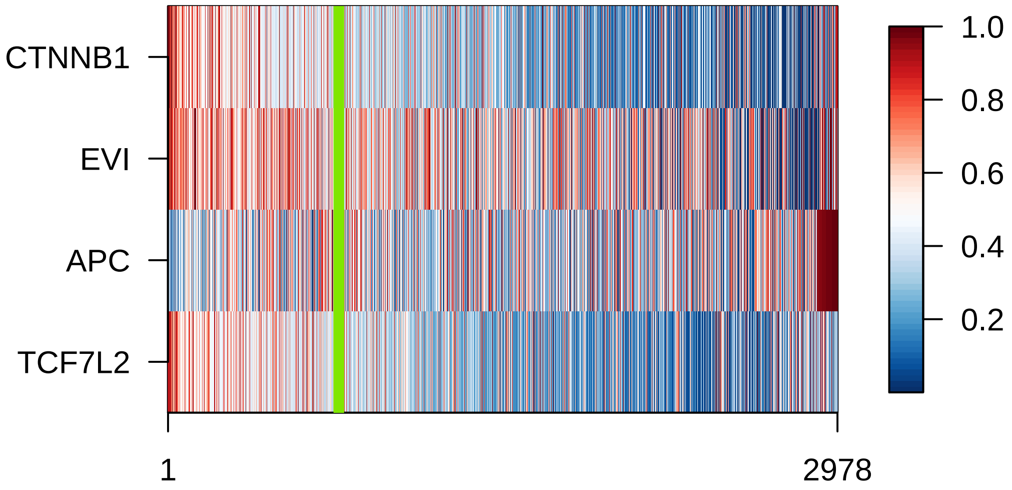


Figure 4: **Posterior probabilities of differential expression.** Heatmap of the posterior probabilities of differential expression in the silencing experiments. Only genes which have a posterior probability larger than .5 of showing an effect in at least one of the knockdown experiments are shown. The green line leaves about 750 genes on its left. The pattern there shows that the majority of those genes respond not only to intervention on β-catenin, but also to EVI, while most of them do not respond to the APC.

No-Conan on RNAseq expression data from the Wnt pathway in colorectal cancer

Hypothesis tests on the nested structure of the observed effects yield a partial learning of a signalling network

Recently, Sadeh et al. [2013] developed an algorithm called No-CONAN for non confoundable network analysis, based on the idea of using a statistical test to decide whether pairs of variables should be connected or not. This type of approach is well established in the context of Gaussian graphical models and Bayesian networks for the reconstruction of a graph skeleton, where conditional independence tests are routinely used in constraint based learning methods. Connections which do not explain the observed data sufficiently well are rejected.

In order to gather insights about the directional flow and possibly the causal relations of the components of a pathway however it is useful to think of the gene signalling networks in terms of nested effects. No-CONAN builds on the idea of applying statistical tests to learn the structure of networks encoding nesting relationships between the effects E of the signalling nodes S. Assuming a deterministic nature of the signalling pathways and a binary measure of the effects (1 for present and 0 for absent), each topology defines a set of possible patterns which can be observed for the E-genes, under each intervention on the signal components. In the No-CONAN terminology configurations of the observed E-genes which are not compatible with the hypothesised underlying network are called alien patterns and can only be explained by noise.

No-CONAN focuses on pairs of pathway genes S_i and S_j and tests every possible up-/downstream relation between them, e.g. S_i is upstream of S_j, S_j is upstream of S_i, they are in a feedback loop, or they are unconnected. The tests are based on the nesting of downstream events in perturbation assays of S_i and S_j. Depending on its attachment position, each E-gene can show four different patterns, responding to silencing of either S-gene, to only one (one or the other) or to both at the same time. For example if we assume that S_i is upstream of S_j, observing that the targets of S_j do not show an effect when intervening on S_i would constitute an alien pattern. Similarly observing an effect on targets of S_i when intervening on S_j, and so on. If a test is significant it means that under the hypothesis that the tested connection is true, the observed pattern is unlikely to be explained by noise alone, therefore the relation is discarded. The tests are applied to all possible pairs of perturbed S-genes.

Our RNAseq data covers only knockdown assays from 5 genes in the Wnt pathway. One might argue that this gives an incomplete account of this complex pathway and all its cross talk with other pathways. Although we do not make global claims about how the pathway works, one might still argue that the pathway needs to be modelled in its entirety, even if one only wants to decide on a single detail. Mechanisms beyond the modelled genes affect the data and might confound our interpretation of the data. Moreover, we only assess changes in the coding transcriptome of cells thus missing out on changes on other levels of cellular regulation like the non-coding transcriptome, chromatin structure, the proteome, or the metabolome. However, even in a huge project that collects all this information there could still be concerns that there are cellular mechanisms affecting our data and hence its interpretation that to date we do not even know of: the unknown unknowns of the pathway.

Sadeh et al. [2013] have systematically addressed the problem of deciding which properties of an estimated network topology can be affected by unobserved or even unknown mechanisms and which cannot. Key to the algorithm is that hidden confounders do not influence the type of E-gene patterns which can be observed in response to each silencing experiment, leaving only the observational noise to explain the observation of alien patterns.

Every test has then the unique property that it cannot be confounded by hidden variables. The relation between a pair of S-genes would be fully resolved, if all but one relation were rejected. However, in many cases this is not possible due to the possible effects of hidden confounders that we need to account for. As explained in detail in Sadeh et al. [2013], that rejection of the disconnected relationship between S_i and S_j is assumed in a strict sense, where no interaction between them are allowed even further downstream. In other words when the possibility of no connection is rejected for a pair of nodes, not only it is intended that there is no direct link between them, but also that they cannot have a common child. When considering two known signalling nodes S_i and S_j the relation $S_i \to H \leftarrow S_j$ of two nodes with a common child can explain every possible configuration of E-genes, therefore it is assumed a priori possible in the No-CONAN approach and can never be excluded. Likewise the only relation that can be fully resolved between two nodes is the topology with two parents and a common hidden child, in the case where the data reject every other possible relation. Interestingly the possibility to fully resolve networks of two parents with a common child means that the No-CONAN approach has the potential to identify the position of hidden nodes in a network. On the other hand it is clear that feedback loops of the type $S_1 \leftrightarrow S_2$ cannot be clearly identified since they include both relations $S_1 \to S_2$ and $S_2 \leftarrow S_1$, neither of which should be rejected if the feedback loop is the true underlying network. Nevertheless by excluding certain of the possible up/down-stream relations the No-CONAN approach provides us with a partially identified network topology a pNEM and important insights into which relations may be worth investigating further.

In general it is true for pNEM that the higher the number of relations which cannot be rejected the higher is the degree of uncertainty in the partially learned network. The limitation in learning is accepted in order to gain confidence in certain features which can be identified in a non confoundable manner. The edges in a pNEM describe relations which cannot be rejected from the observed data [Sadeh et al., 2013].

Technically No-CONAN performs a statistical test in order to detect unusually high numbers of observed alien patterns, which for a given connection are unlikely to be only due to noise. Let γ_R be a bound on the probability that alien patterns for relationship R are observed, it has been shown [Sadeh et al., 2013] that for each possible relation R between S_i and S_j the probability of observing at least a alien patterns out of L can be bound as

$$P(A \geq a|R) = \sum_{l=a}^{L} \binom{L}{l} \gamma_R^l (1-\gamma_R)^{L-l}$$

and used to conduct a test, where a relationship R is rejected if

$$P(A \geq a|R) < \kappa$$

where κ is a calibration parameter typically set to .05.

Non confoundable network analysis rejects the dominant activation model using a statistical test that cannot be confounded by missing data or unknown unknowns of Wnt signalling

The difference between the DAM and the ASM is mainly captured by the type of edge between and . Dominant activation assumes that this edge does not exist. It postulates that in CRC no information is flowing through the upstream parts of the Wnt signalling pathway and hence is not on the path from the mutated proteins to the Wnt target genes. In contrast, activation by sensitisation postulates the autocrine stimulation of the intact Wnt pathway, and here mediated secretion of Wnt ligands is an event that occurs upstream of the activity in the nucleus. Hence, in this model there must be an edge from to . Our strategy is to use No-CONAN to test whether this edge exists or not. Since the No-CONAN tests are not confoundable by hidden data (no matter whether it was unobserved or observed but excluded from the analysis), we could confine the analysis to the RNAseq profiles from the and knockdowns only. Since we are interested to learn whether in CRC affects targets downstream of in the signalling we only included in the analysis E-genes that responded to the knockdowns.

Parameter calibration is critical to a No-CONAN analysis. The algorithm requires the input data to be in the form of a binary matrix = (d_{lk}) with rows D_l corresponding to E-genes and NoCONAN repetitionsHow are repetitions handled? Simply as more E-genes? columns D_k corresponding to silenced S-genes. An element $d_{lk} = 1$ indicates that the expression of E-gene l

was affected by the knockdown of S-gene k in the RNAi experiment. When testing the DAM model, it is assumed that there is no connection between and . An unexpected or alien pattern is observed for E-genes which shows an effect both when perturbing and β-catenin. No-CONAN tests whether the number of targets responding to both interventions is significantly higher than it would be expected by noise alone. No-CONAN requires an estimate of the size of the binary noise α to be expected in the data. This parameter is the probability of a false call and needs to be set by the user. i.e. the probability that the data holds a 1 although a 0 was correct and vice versa. The calibration of α can greatly affect the results and is therefore critical. For each knockdown the data were binarised on the basis of the posterior probability of differential expression. Values of the posterior probabilities above a certain threshold λ were set to 1 and values below λ were set to 0. Again the choice of an appropriate cutoff λ is critical to the analysis. Like in the previous analysis our strategy was to use multiple settings of the parameters to assess the robustness of the analysis. The discretising threshold was varied between .5 and .99 in steps of .01. Noise levels ranging from .02 and .32 were considered in the simulations. The p-value for rejecting the DAM, or more formally for rejecting the relation "unconnected" for the pair of genes and β-catenin, was found to be virtually zero for all noise levels, and no matter how many genes were included in the analysis.

The inclusion of all β-catenin responsive genes as E-genes might lead to the inclusion of genes that are not Wnt targets but are affected by β-catenin functions outside the pathway, e.g. in cell adherence mediated signalling. To filter those out, we rerun the analysis including only genes whose posterior probability of differential expression was above the cutoff λ for both the β-catenin and the knockdowns. Although the number of E-genes included in the analysis was greatly reduced, the No-CONAN again rejected the DAM over the full range of cutoff values and even for very high noise levels.

References

Charity W Law, Yunshun Chen, Wei Shi, and Gordon K Smyth. voom: precision weights unlock linear model analysis tools for rna-seq read counts. *Genome Biology*, 15:R29, 2014.

Mohammad J. Sadeh, Giusi Moffa, and Rainer Spang. Considering unknown unknowns - reconstruction of non-confoundable causal relations in biological networks. *Journal of Computational Biology*, 20:920–932, 2013.

Gordon K. Smyth. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3, 2004.