Towards confidence estimates in Cascade networks

Ismaïl Aouadi, Nicolas Jung, Raphael Carapito, Laurent Vallat, Seiamak Bahram, Myriam Maumy-Bertrand, Frédéric Bertrand

27th January, 2019

#Extending results from the Cascade package: reverse engineering with selectboost to  
#compute confidence indices on the inferred links.  
#Code to reproduce the datasets saved with the package and the figures of the  
#article Aouadi et al. (2018), <arXiv:1810.01670>  
  
#Reference for the Cascade modelling  
# Vallat, L., Kemper, C. a., Jung, N., Maumy-Bertrand, M., Bertrand, F.,  
# Meyer, N., Pocheville, A., Fisher, J. W., Gribben, J. G. et Bahram, S.  
# (2013). Reverse-engineering the genetic circuitry of a cancer cell with predicted  
# intervention in chronic lymphocytic leukemia. Proceedings of the National  
# Academy of Sciences of the United States of America, 110(2), 459-64.  
  
#Reference for the Cascade package  
# Jung, N., Bertrand, F., Bahram, S., Vallat, L. et Maumy-Bertrand, M. (2014).  
# Cascade : A R package to study, predict and simulate the diffusion of a signal  
# through a temporal gene network. Bioinformatics. ISSN 13674803.  
  
library(Cascade)  
  
#We change the array with the F matrices  
T<-4  
F<-array(0,c(T-1,T-1,T\*(T-1)/2))  
  
for(i in 1:(T\*(T-1)/2)){diag(F[,,i])<-1}  
F[,,2]<-F[,,2]\*0.2  
F[2,1,2]<-1  
F[3,2,2]<-1  
F[,,4]<-F[,,2]\*0.3  
F[3,1,4]<-1  
F[,,5]<-F[,,2]  
  
#We set the seed to make the results reproducible  
set.seed(1)  
Net<-Cascade::network\_random(  
 nb=100,  
 time\_label=rep(1:4,each=25),  
 exp=1,  
 init=1,  
 regul=round(rexp(100,1))+1,  
 min\_expr=0.1,  
 max\_expr=2,  
 casc.level=0.4  
)  
Net@F<-F  
  
#We simulate gene expression according to the network Net  
M <- Cascade::gene\_expr\_simulation(  
 network=Net,  
 time\_label=rep(1:4,each=25),  
 subject=5,  
 level\_pic=200)

## Loading required package: VGAM

## Loading required package: stats4

## Loading required package: splines

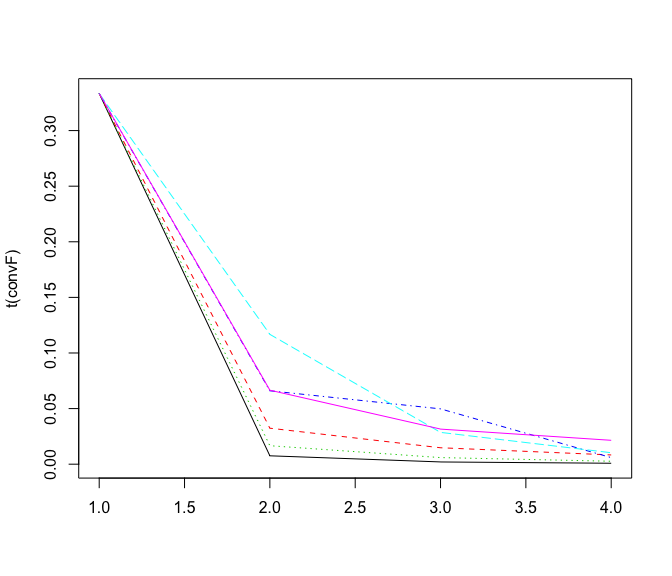
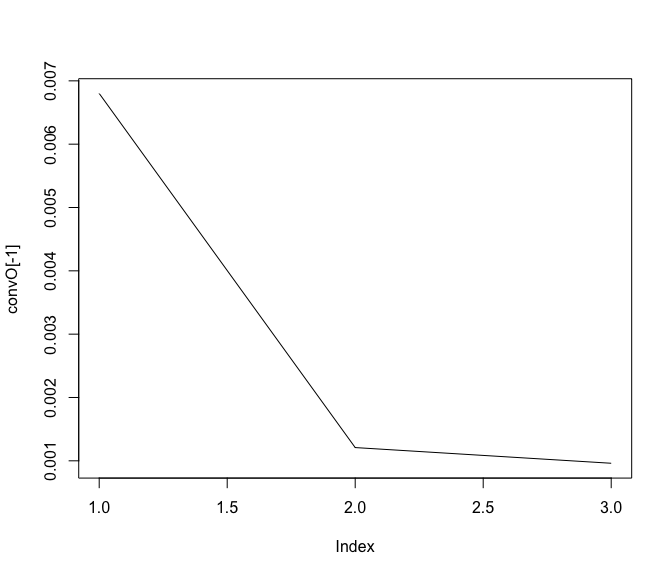
## Loading required package: magic

## Loading required package: abind

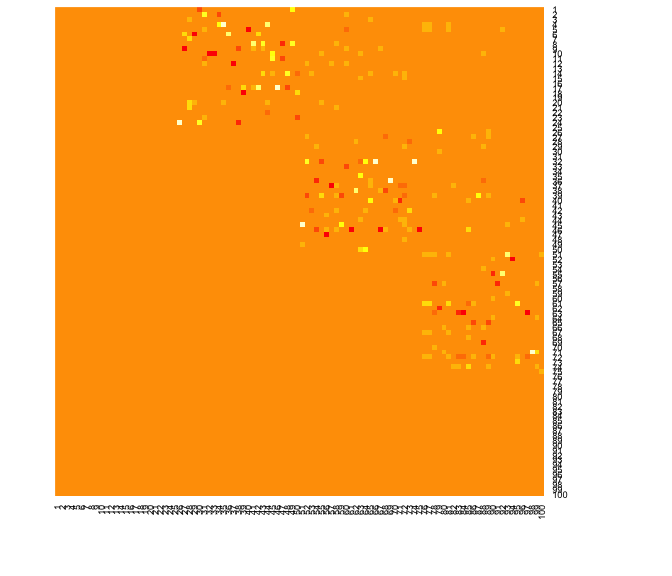
#We infer the new network  
Net\_inf\_C <- Cascade::inference(M,cv.subjects=TRUE)

## Loading required package: nnls

## We are at step : 1  
## The convergence of the network is (L1 norm) : 0.0068  
## We are at step : 2  
## The convergence of the network is (L1 norm) : 0.00121  
## We are at step : 3  
## The convergence of the network is (L1 norm) : 0.00096



#heatmap of the coefficients of the Omega matrix of the network  
stats::heatmap(Net\_inf\_C@network, Rowv = NA, Colv = NA, scale="none", revC=TRUE)



Fab\_inf\_C <- Net\_inf\_C@F  
  
  
library(SelectBoost)

##   
## Attaching package: 'SelectBoost'

## The following object is masked from 'package:Cascade':  
##   
## compare

set.seed(1)  
#by default the crossvalidation is made subjectwise according to a leave one out  
#scheme and the resampling analysis is made at the .95 c0 level  
net\_confidence <- selectboost(M, Fab\_inf\_C)

##   
## We are at peak : 2  
## .........................  
## We are at peak : 3  
## .........................  
## We are at peak : 4  
## .........................

net\_confidence\_.5 <- selectboost(M, Fab\_inf\_C, c0value = .5)

##   
## We are at peak : 2  
## .........................  
## We are at peak : 3  
## .........................  
## We are at peak : 4  
## .........................

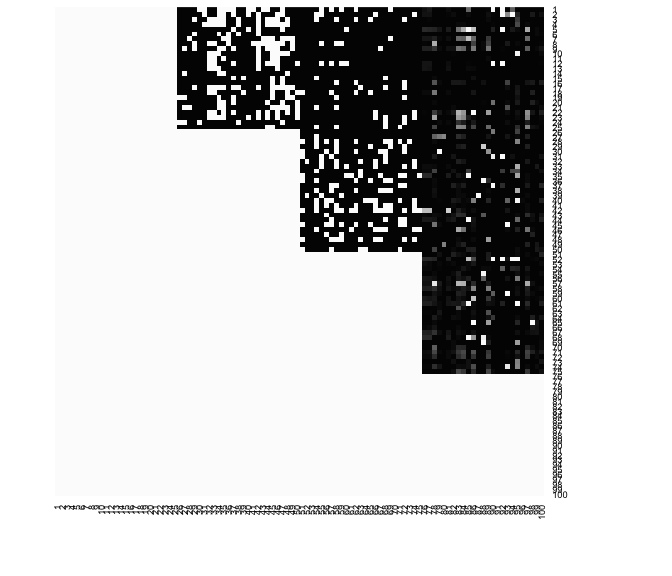
net\_confidence\_thr <- selectboost(M, Fab\_inf\_C, group = group\_func\_1)

##   
## We are at peak : 2  
## .........................  
## We are at peak : 3  
## .........................  
## We are at peak : 4  
## .........................

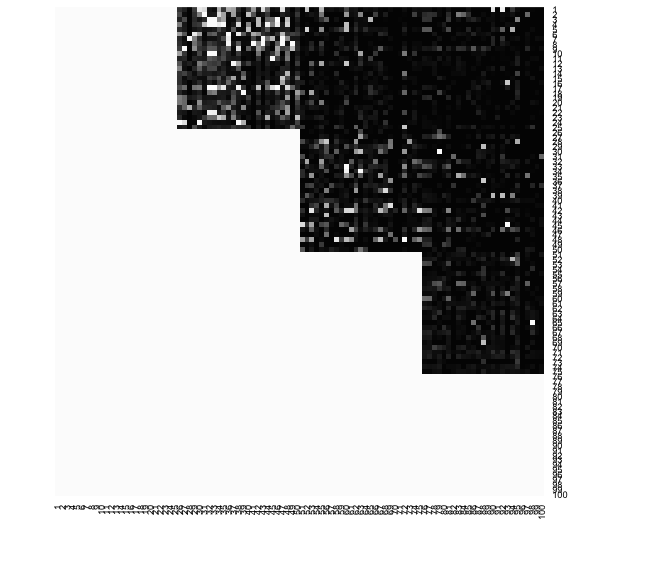
#use cv.subject=FALSE to use default crossvalidation  
net\_confidence\_cv <- selectboost(M, Fab\_inf\_C, cv.subject=FALSE)

##   
## We are at peak : 2  
## .........................  
## We are at peak : 3  
## .........................  
## We are at peak : 4  
## .........................

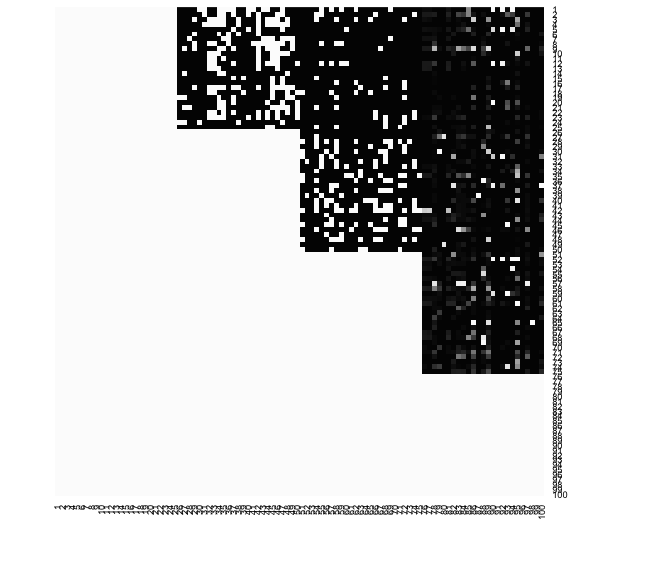
#use plot to display the result of the confidence analysis  
plot(net\_confidence)



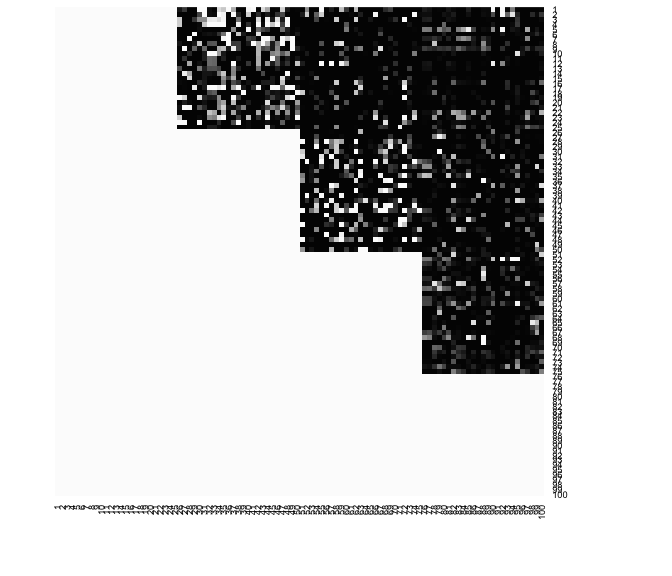
plot(net\_confidence\_.5)



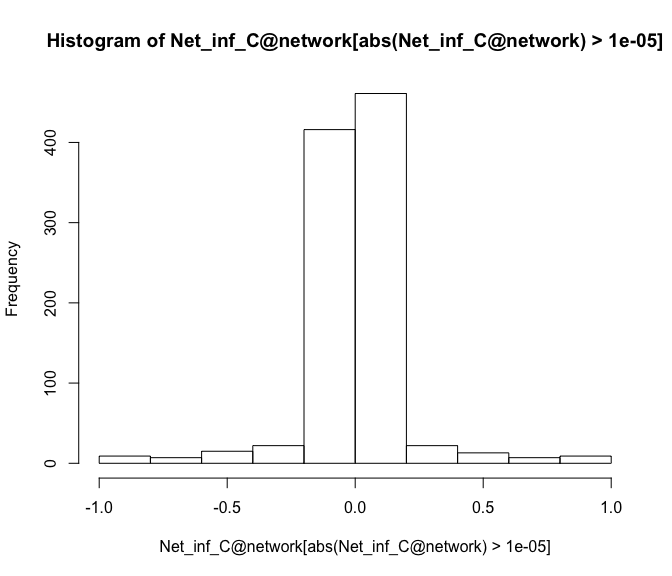
plot(net\_confidence\_thr)



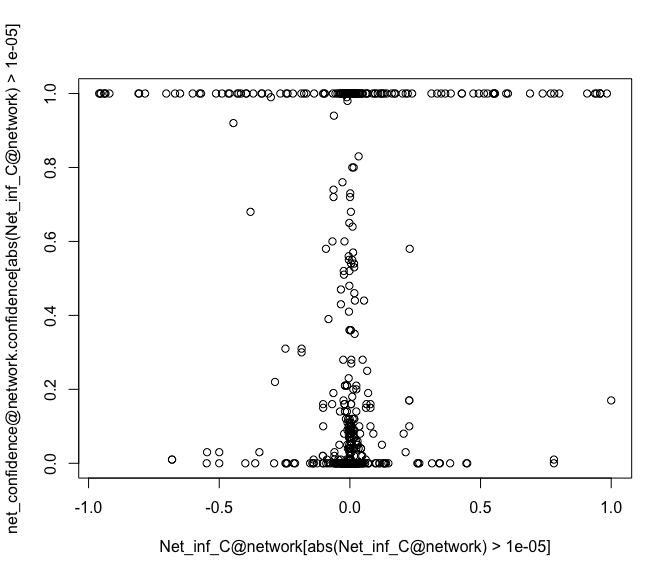
plot(net\_confidence\_cv)



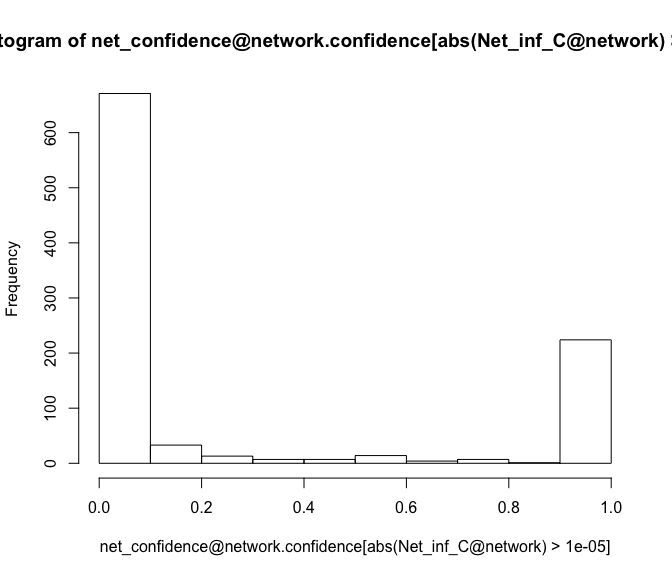
#Distribution of non-zero (absolute value > 1e-5) coefficients  
hist(Net\_inf\_C@network[abs(Net\_inf\_C@network)>1e-5])



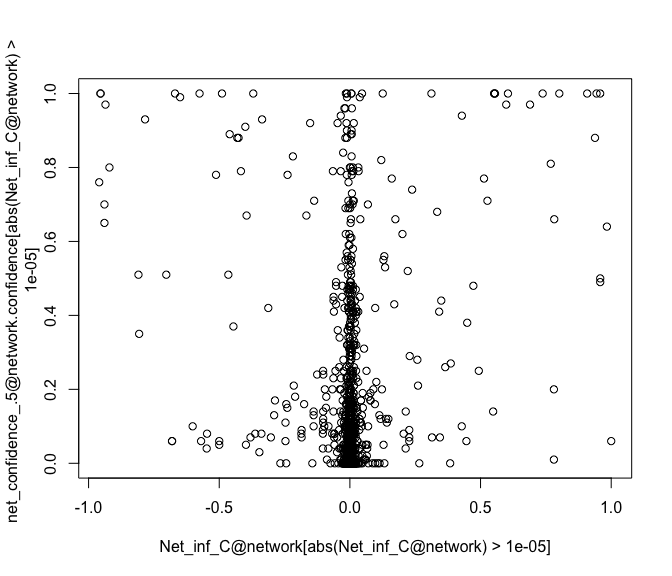
#Plot of confidence at .95 resampling level versus coefficient value for non-zero (absolute value > 1e-5) coefficients  
plot(Net\_inf\_C@network[abs(Net\_inf\_C@network)>1e-5],net\_confidence@network.confidence[abs(Net\_inf\_C@network)>1e-5])



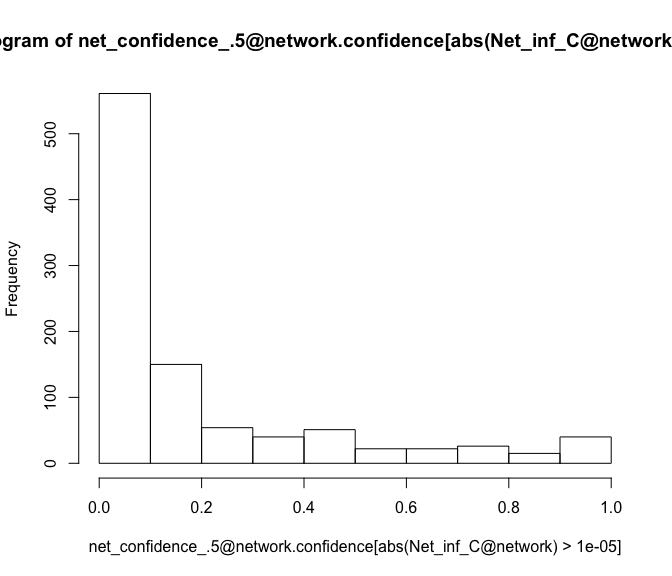
hist(net\_confidence@network.confidence[abs(Net\_inf\_C@network)>1e-5])



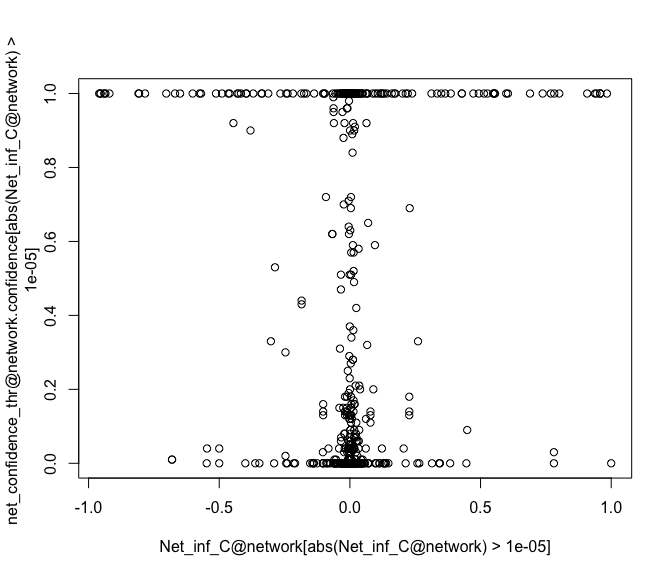
#Plot of confidence at .5 resampling level versus coefficient value for non-zero (absolute value > 1e-5) coefficients  
plot(Net\_inf\_C@network[abs(Net\_inf\_C@network)>1e-5],net\_confidence\_.5@network.confidence[abs(Net\_inf\_C@network)>1e-5])



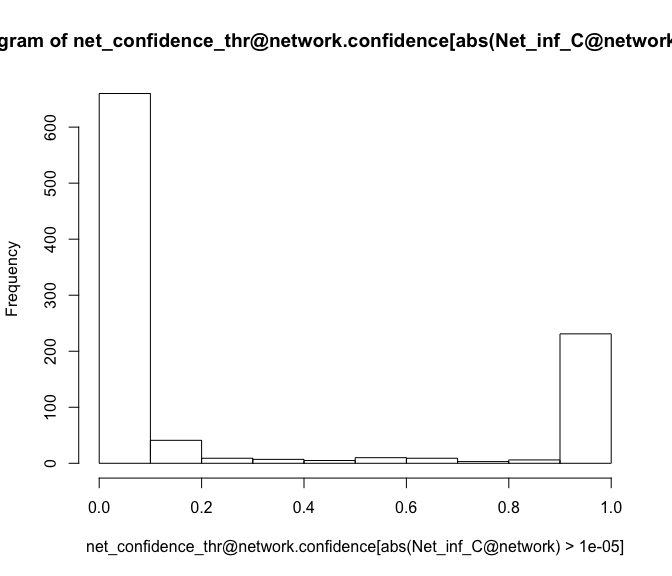
hist(net\_confidence\_.5@network.confidence[abs(Net\_inf\_C@network)>1e-5])



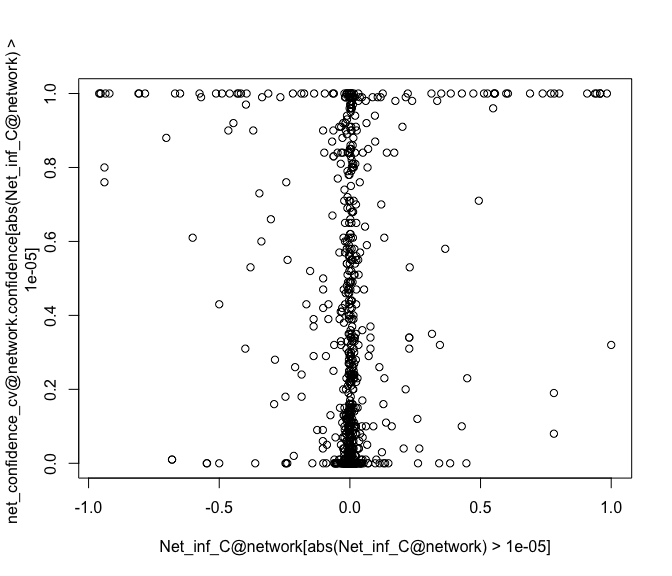
#Plot of confidence at .95 resamling level with groups created by thresholding the correlation matrix  
#versus coefficient value for non-zero (absolute value > 1e-5) coefficients  
plot(Net\_inf\_C@network[abs(Net\_inf\_C@network)>1e-5],net\_confidence\_thr@network.confidence[abs(Net\_inf\_C@network)>1e-5])



hist(net\_confidence\_thr@network.confidence[abs(Net\_inf\_C@network)>1e-5])



#Plot of confidence at .95 resampling level versus coefficient value for non-zero (absolute value > 1e-5) coefficients using standard cross-validation  
plot(Net\_inf\_C@network[abs(Net\_inf\_C@network)>1e-5],net\_confidence\_cv@network.confidence[abs(Net\_inf\_C@network)>1e-5])



hist(net\_confidence\_cv@network.confidence[abs(Net\_inf\_C@network)>1e-5])

