Drug prediction tutorial

A. Gabor & A. V. Ponce

1/16/2020

Contents

Introduction

The goal of this tutorial is to show an example where dynamic logic model is used to predict drug response.

The tutorial is based on the paper

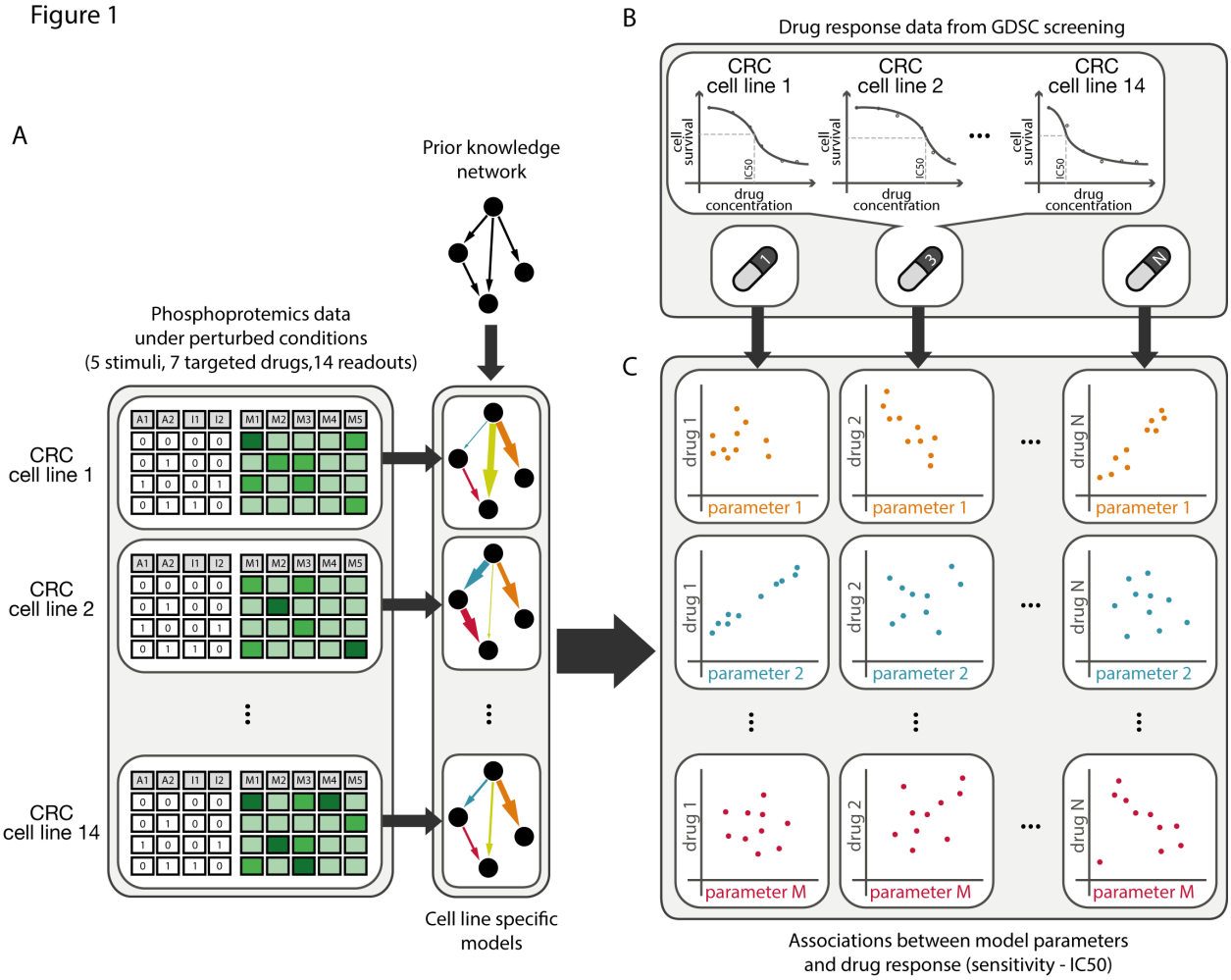
> Eduati et al (2017) Drug resistance mechanisms in colorectal cancer dissected with cell type-specific dynamic logic models. *Cancer Research*. DOI: 10.1158/0008-5472.CAN-17-0078

On Github: https://github.com/saezlab/CRC-pathway-biomarkers

We investigate here the drug-response of colorectal cancer cell lines. For this, we use drug response data and a signaling dataset.

```
knitr::include_graphics("./data/tutorial_3/Eduatietal_Figure1.png")
```

Figure 1



The Genomics of Drug Sensitivity in Cancer (GDSC), https://www.cancerrxgene.org/ offers drug response data for more than a 1000 human cancer cell lines, for houndreds of drugs. A small part of these data is can be found in `"./data/IC50_GDSC.csv"`.

The perturbation dataset contains the short time signaling response of 14 colorectal cancer cell lines, where 14 phosphoproteins are measured under 43 perturbation conditions (5 stimuli, 7 inhibitors).

First, we construct signaling models based on the perturbation data to the cell lines, here we use the CNORode modelling package. In the next step, we will associate model features to drug response to see why certain cell lines respond to certain drugs and others do not. Here we use a linear modeling framework.

CNORode

CNORode is a member of the CellNOptR logic based tool family. It can translate the network to ordinary differential equation (ODE) model, fit the model parameters to data and make predictions.

Dependencies

These should be already installed from previous tutorial.

```
# installs devtools package if not already installed
if(!require("devtools")) install.packages('devtools')

# installs CellNOptR and CNORode from GitHub:
if(!require("CellNOptR")) devtools::install_github('saezlab/CellNOptR')
if(!require("CNORode")) devtools::install_github('saezlab/CNORode')

if(!require("dplyr")) install.packages('dplyr')
if(!require("readr")) install.packages('readr')
if(!require("tidyr")) install.packages('tidyr')
```

If you dont have `devtools` and cannot install it, then

1. please visit the https://github.com/saezlab/CellNOptR and https://github.com/saezlab/CNORode websites,
2. download the toolboxes by clicking "Clone or download" then "Download Zip"
3. Unzip the files
4. In RStudio run:
   `install.packages("../CellNOptR-master", repos = NULL, type = "source")`
   `install.packages("../CNORode-master", repos = NULL, type = "source")`

Make sure to import the libraries

```
library(CellNOptR)
library(CNORode)
library(MEIGOR)
```

```
## Loading required package: Rsolnp
```

```
## Loading required package: snowfall
```

```
## Loading required package: snow
```

```
##
## Attaching package: 'snow'
```

```
## The following objects are masked from 'package:BiocGenerics':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, clusterSplit, parApply, parCapply,
##     parLapply, parRapply, parSapply
```

```
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, clusterSplit, makeCluster,
##     parApply, parCapply, parLapply, parRapply, parSapply,
##     splitIndices, stopCluster
```

```
## Loading required package: deSolve
```

```
library(dplyr)
library(tidyr)
library(ggplot2)
```

PART I: DRUG response exploration

```
IC50 <- readr::read_csv("./data/tutorial_3/IC50_GDSC.csv") %>% rename("cell_line" = "X1")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   X1 = col_character()
## )
```

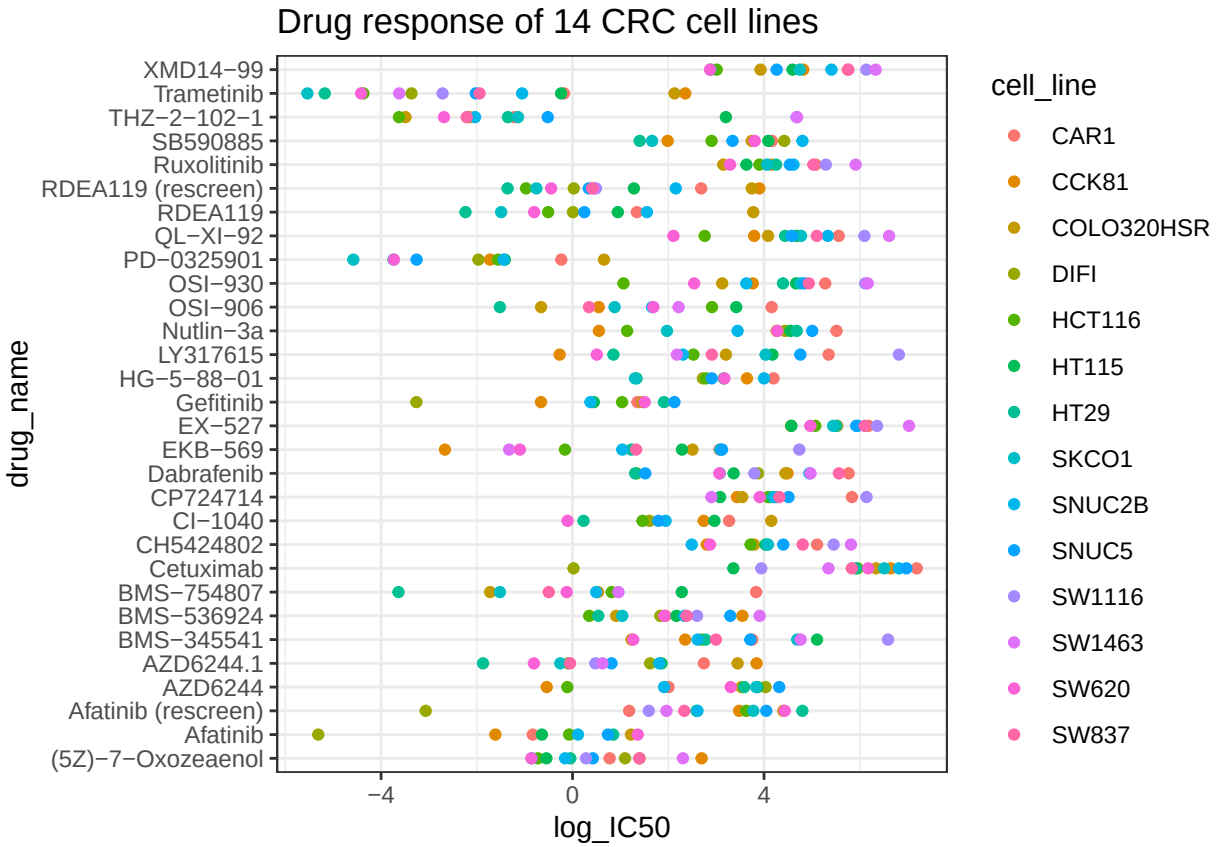
```
## See spec(...) for full column specifications.
```

```
print(IC50)
```

```
## # A tibble: 14 x 31
##    cell_line Gefitinib  RDEA119 `CI-1040` Afatinib `Nutlin-3a` `PD-0325901`
##    <chr>         <dbl>    <dbl>     <dbl>    <dbl>       <dbl>        <dbl>
##  1 CAR1           1.36   1.34        3.27   -0.831        5.51        -0.235
##  2 CCK81         -0.659 NA           2.74   -1.61         0.551       -1.72
##  3 COLO320H~      1.46   3.78        4.15    1.22         4.26         0.658
##  4 DIFI          -3.26   0.00908     1.61   -5.31         4.44        -1.97
##  5 HCT116         1.03  -0.510       1.46   -0.0704       1.14        -1.56
##  6 HT115          0.444  0.946       2.96   -0.640        4.55        -1.41
##  7 HT29           1.91  -2.24        0.230   0.849        4.68        -3.74
##  8 SKCO1         NA     -1.49       NA      NA            1.97        -4.58
##  9 SNUC2B         0.375  1.55        1.94    0.113        3.45        -1.44
## 10 SNUC5          2.13   0.246       1.79    0.739        5.01        -3.26
## 11 SW1116        NA     NA          NA      NA           NA           NA
## 12 SW1463        NA     NA          NA      NA           NA           NA
## 13 SW620          1.51  -0.802      -0.104   1.36         4.28        -3.73
## 14 SW837         NA     NA          NA      NA           NA           NA
## # ... with 24 more variables: SB590885 <dbl>, AZD6244 <dbl>,
## #   `BMS-536924` <dbl>, Cetuximab <dbl>, `HG-5-88-01` <dbl>,
## #   `(5Z)-7-Oxozeaenol` <dbl>, Trametinib <dbl>, Dabrafenib <dbl>,
## #   `Afatinib (rescreen)` <dbl>, AZD6244.1 <dbl>, `RDEA119
## #   (rescreen)` <dbl>, `BMS-754807` <dbl>, `OSI-906` <dbl>,
## #   `BMS-345541` <dbl>, Ruxolitinib <dbl>, LY317615 <dbl>,
## #   `XMD14-99` <dbl>, CP724714 <dbl>, CH5424802 <dbl>, `EKB-569` <dbl>,
## #   `OSI-930` <dbl>, `QL-XI-92` <dbl>, `EX-527` <dbl>, `THZ-2-102-1` <dbl>
```

```
IC50 %>% gather(drug_name, log_IC50, -cell_line) %>%
  ggplot() +
  geom_point(aes(drug_name, log_IC50,col=cell_line)) +
  coord_flip() +
  theme_bw() +
  ggtitle("Drug response of 14 CRC cell lines")
```

```
## Warning: Removed 49 rows containing missing values (geom_point).
```



Form the raw IC50 values we can see that there are some drugs that are more effective (Trametinib) than others, like XMD14-99. There are also cell-line differences, for example, DIFI shows stronger sensitivity to Afatinib than any other cell lines. What could be the reason for this?

PART II: cell-line models

The goal of part II is to build a cell-line specific model from the perturbation data using CNORode.

This model is an ordinary differential equation (ODE) model, where the equation for each state (x_A) can be written as

$$\frac{dx_A}{dt} = \tau_A(B(f_1(x), f_2(x), ...) - x_A)$$

here

- $f_i(x)$ represents the incoming edges on node A with a transfer function. This transfer function typically has an S-shape.

$$f(x) = \frac{x^n}{x^n + k^n}$$

- B is a Boolean homologue function. This is responsible to combine the incoming edges with the OR and AND gates. For example, an OR gate is represented by $x_1 \cdot x_2$.
- τ is a time parameter, that tells how fast node A adapts to the input.
- the model has free parameters: a τ for each node, and (k,n) for each edge. These are found by optimisation.

The main differences are that in ODE models the states are continuous values, therefore it is quantitative, not only qualitative like a Boolean model. Further, here we hace to find the specific edge and node parameters.

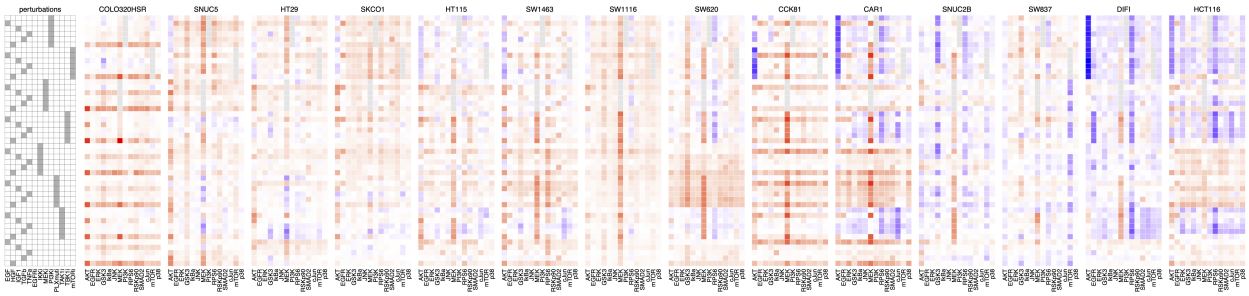
What do we need to build and simulate a differential equation model?

- the equations are derived from the network graph
- inputs: given in the MIDAS description
- Initial conditions for each state in each experiment

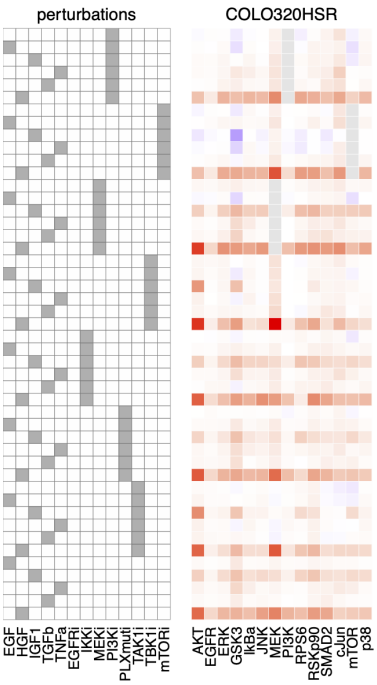
In this example, the baseline is set to 0.5. A value of 1 means full activation and 0 means full inhibition of the node.

Perturbation data

The following heatmap shows an overview on the perturbation data. The first block outlines the combinations of treatment. Then each other block represents the response of a cell line. Different columns within a block shows the different phosphoprotein markers.



In the tutorial we make a single model for the first cell line *COLO320HSR*.



Model a single cell line

Similarly to the previous tutorial with CellNopt, here we also start by importing a prior knowledge network and the perturbation data in MIDAS format.

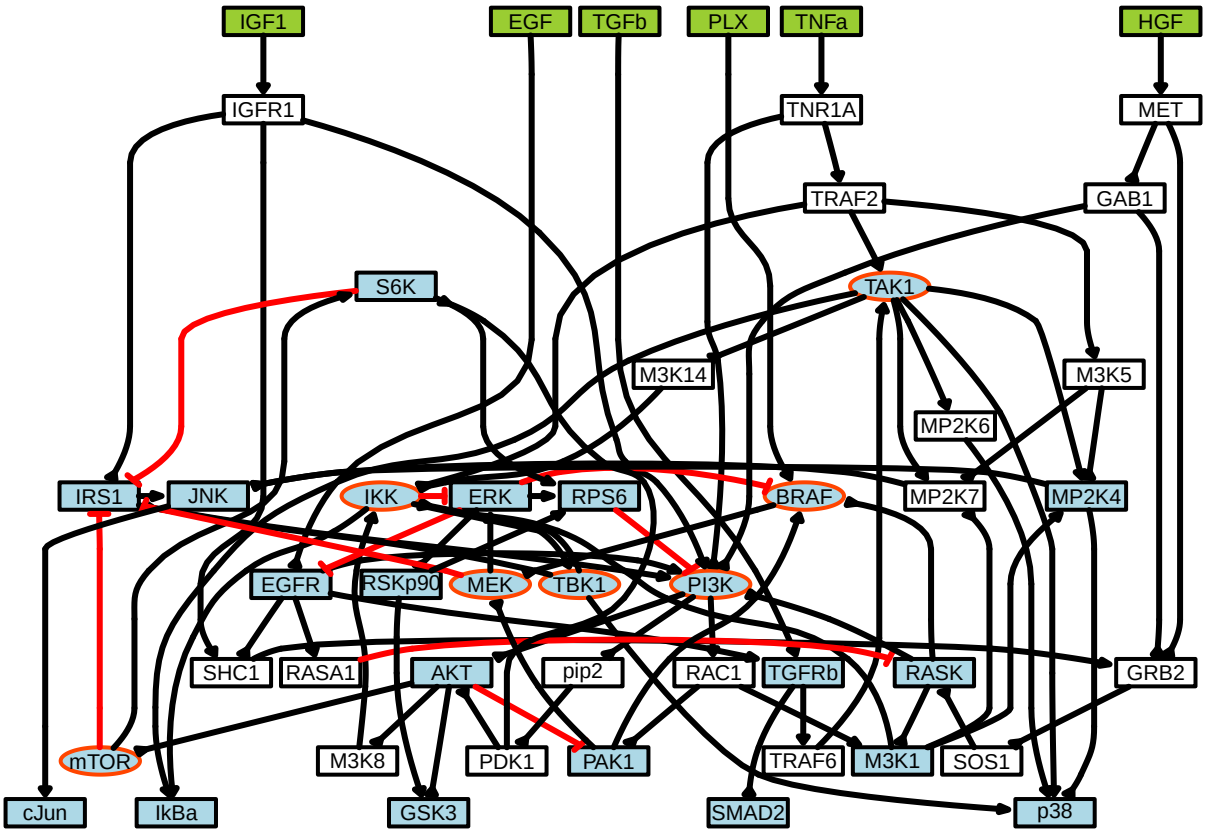
```
# load Prior Knowledge Network (PKN)
pknmodel<-readSIF("./data/tutorial_3/PKN.sif")

# load normalised perturbation data
# select MIDAS file for the desired cell line
MIDASfile <- "./data/tutorial_3/processed/MIDAS/MD-COLO320HSR_Ktuned_v1_n4_all_noEGFRi_CNORode.csv"

Mydata<-readMIDAS(MIDASfile=MIDASfile,verbose = FALSE)
cnolist<-makeCNOlist(Mydata, subfield=F)
```

```
## [1] "Please be aware that if you only have some conditions at time zero (e.g.only inhibitor/no inhibi
```

```
cnolist$valueStimuli[cnolist$valueStimuli==0]=0.5
```

Show the network first

```
plotModel(pknmodel,cnolist)
```

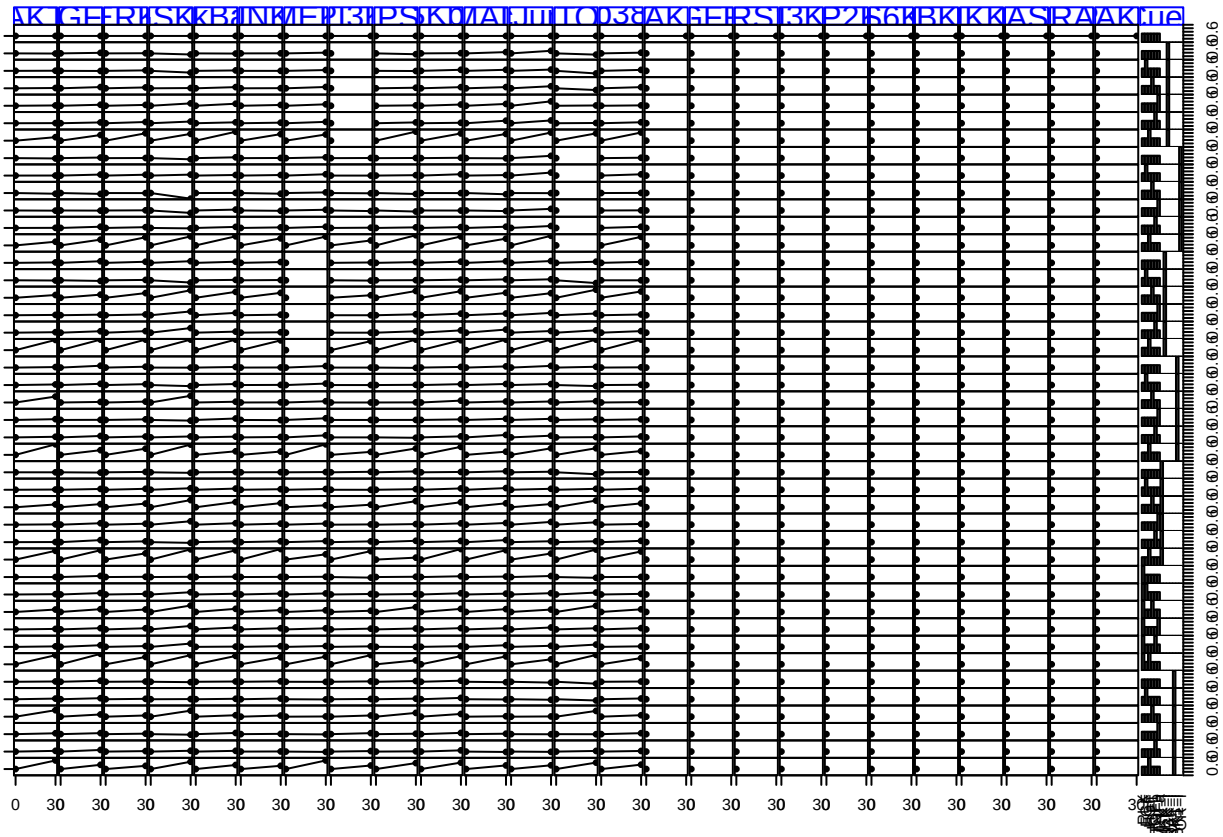


As in CellNOPt:

- green nodes are stimulated in some experiments
- blue nodes are measured
- white nodes are modelled, but not measured
- red nodes or red bordered nodes are occasionally inhibited
- black edges represents activation, red T-shaped arrows represents inhibition

Then the data in CellNOpt format:

```
plotCNOlist(cnolist)
```



The data is very large, therefore is hard to see the details, but we can notice as some nodes increases their activity at the final time (30 mins).

```
# compress the network (no expansion, only OR gates are considered)
model<-preprocessing(data=cnolist, model=pknmodel, compression=TRUE, expansion=FALSE)
```

```
## [1] "The following species are measured: AKT, EGFR, ERK, GSK3, IkBa, JNK, MEK, PI3K, RPS6, RSKp90, S
## [1] "The following species are stimulated: PLX, EGF, HGF, IGF1, TGFb, TNFa"
## [1] "The following species are inhibited: IKK, MEK, PI3K, BRAF, TAK1, TBK1, mTOR"
## [1] "The following species are not observable and/or not controllable: "
```

```
# set initial parameters (here parameters 'k' and 'tau' are optimised and 'n' fixed to 3)
ode_parameters <- createLBodeContPars(model,
                                      LB_n = 1, LB_k = 0, LB_tau = 0,
                                      UB_n = 3, UB_k = 1, UB_tau = 1,
                                      default_n = 3,
                                      default_k = 0.5,
                                      default_tau = 0.01,
                                      opt_n = FALSE, opt_k = TRUE, opt_tau = TRUE,
                                      random = TRUE)

# PLX -> BRAF is an artificial regulation used to model paradoxical effect of PLX4720,
# which works as selective BRAF inhibitor in cell-lines where BRAF is mutated in
# V600E (i.e. HT29 and SNUC5 in our panel), but induces a paradoxical activation
# of wild type BRAF cells (modeled as stimulus on those cell lines)
```

```
ode_parameters$parValues[which(ode_parameters$parNames=="PLX_k_BRAF")]<-0.5
ode_parameters$index_opt_pars<- setdiff(ode_parameters$index_opt_pars,
                                        which(ode_parameters$parNames=="PLX_k_BRAF"))

## Parameter Optimization
# essm
paramsSSm=defaultParametersSSm()
paramsSSm$local_solver = "DHC"
paramsSSm$maxtime = 30; #36000;
paramsSSm$transfer_function = 4;
```

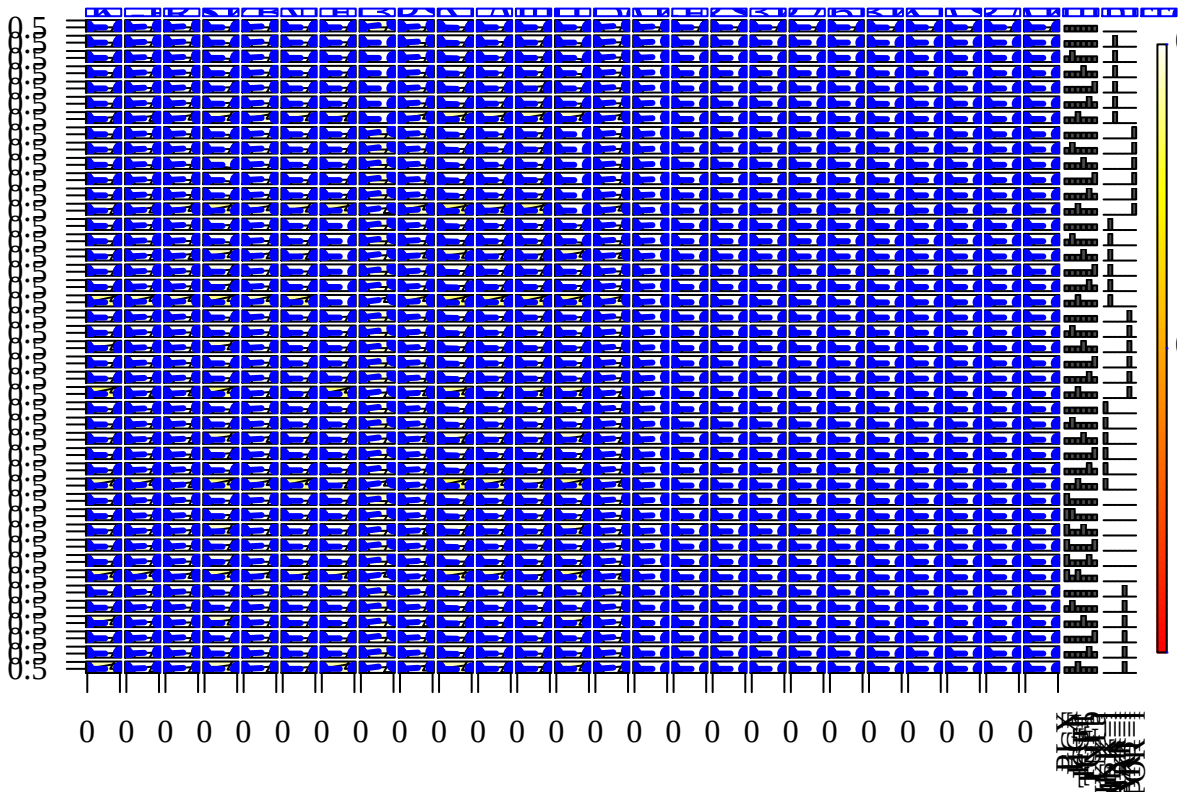
The actual optimisation takes around 10 mins, instead of the 30 sec. So, instead of running it here, we just load the results:

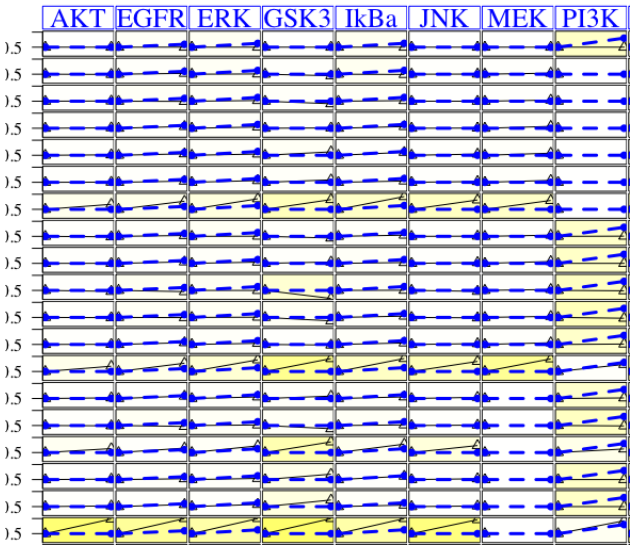
```
opt_pars = parEstimationLBode(cnolist, model, method="essm",
                              ode_parameters=ode_parameters,
                              paramsSSm=paramsSSm)
#write_rds(opt_pars,"data/tutorial_3/opt_pars_30sec.RDS")
```

```
opt_pars <- read_rds("data/tutorial_3/opt_pars_30sec.RDS")
```

Plot the fit of the model:

```
sim_res <- CNORode::plotLBodeFitness(cnolist,model,ode_parameters = opt_pars)
```





The fit is a bit better than a random model, but these optimisations should be run for around 10 hours.

We are interested in the optimised model parameters of this model.

```
opt_par_values <- opt_pars$parValues
names(opt_par_values) <- opt_pars$parNames
opt_par_values
```

```
##  TGFRb_n_TAK1  TGFRb_k_TAK1   TNFa_n_TAK1   TNFa_k_TAK1      tau_TAK1
##  3.000000e+00  6.290662e-01  3.000000e+00  0.000000e+00  7.787088e-03
##  TGFb_n_TGFRb  TGFb_k_TGFRb  EGFR_n_TGFRb  EGFR_k_TGFRb     tau_TGFRb
##  3.000000e+00  2.802733e-01  3.000000e+00  1.542701e-01  0.000000e+00
##    EGF_n_EGFR    EGF_k_EGFR    ERK_n_EGFR    ERK_k_EGFR      tau_EGFR
##  3.000000e+00  0.000000e+00  3.000000e+00  2.007754e-07  7.015971e-03
##    S6K_n_IRS1    S6K_k_IRS1   TBK1_n_IRS1   TBK1_k_IRS1   mTOR_n_IRS1
##  3.000000e+00  0.000000e+00  3.000000e+00  0.000000e+00  3.000000e+00
##   mTOR_k_IRS1   IGF1_n_IRS1   IGF1_k_IRS1    MEK_n_IRS1    MEK_k_IRS1
##  0.000000e+00  3.000000e+00  0.000000e+00  3.000000e+00  4.060043e-07
##      tau_IRS1    PI3K_n_AKT    PI3K_k_AKT       tau_AKT   PI3K_n_M3K1
##  0.000000e+00  3.000000e+00  4.100319e-02  0.000000e+00  3.000000e+00
##   PI3K_k_M3K1   RASK_n_M3K1   RASK_k_M3K1      tau_M3K1    ERK_n_RPS6
##  4.432509e-01  3.000000e+00  5.075054e-08  1.568557e-09  3.000000e+00
##    ERK_k_RPS6    S6K_n_RPS6    S6K_k_RPS6 RSKp90_n_RPS6 RSKp90_k_RPS6
##  2.498247e-01  3.000000e+00  0.000000e+00  3.000000e+00  8.162135e-01
##      tau_RPS6     IKK_n_ERK     IKK_k_ERK     MEK_n_ERK     MEK_k_ERK
##  1.027507e-02  3.000000e+00  3.065270e-01  3.000000e+00  1.271568e-01
##       tau_ERK  TAK1_n_MP2K4  TAK1_k_MP2K4  M3K1_n_MP2K4  M3K1_k_MP2K4
##  9.911941e-03  3.000000e+00  1.000000e+00  3.000000e+00  0.000000e+00
##  TNFa_n_MP2K4  TNFa_k_MP2K4     tau_MP2K4    mTOR_n_S6K    mTOR_k_S6K
##  3.000000e+00  0.000000e+00  0.000000e+00  3.000000e+00  5.503381e-01
##    PI3K_n_S6K    PI3K_k_S6K       tau_S6K    IKK_n_TBK1    IKK_k_TBK1
##  3.000000e+00  0.000000e+00  1.462364e-05  3.000000e+00  6.452658e-01
##      tau_TBK1    AKT_n_mTOR    AKT_k_mTOR      tau_mTOR    TAK1_n_IKK
##  9.931725e-03  3.000000e+00  0.000000e+00  0.000000e+00  3.000000e+00
##    TAK1_k_IKK     AKT_n_IKK     AKT_k_IKK    M3K1_n_IKK    M3K1_k_IKK
##  8.246211e-01  3.000000e+00  0.000000e+00  3.000000e+00  0.000000e+00
##    TNFa_n_IKK    TNFa_k_IKK    TBK1_n_IKK    TBK1_k_IKK       tau_IKK
```

```
##  3.000000e+00  0.000000e+00  3.000000e+00  8.707054e-05  0.000000e+00
##   EGFR_n_PI3K   EGFR_k_PI3K   IRS1_n_PI3K   IRS1_k_PI3K   RPS6_n_PI3K
##  3.000000e+00  1.599331e-01  3.000000e+00  0.000000e+00  3.000000e+00
##   RPS6_k_PI3K   TNFa_n_PI3K   TNFa_k_PI3K   IGF1_n_PI3K   IGF1_k_PI3K
##  1.000000e+00  3.000000e+00  0.000000e+00  3.000000e+00  0.000000e+00
##    HGF_n_PI3K    HGF_k_PI3K   RASK_n_PI3K   RASK_k_PI3K      tau_PI3K
##  3.000000e+00  4.751120e-01  3.000000e+00  3.009077e-02  3.042961e-02
##   EGFR_n_RASK   EGFR_k_RASK   IGF1_n_RASK   IGF1_k_RASK    HGF_n_RASK
##  3.000000e+00  0.000000e+00  3.000000e+00  0.000000e+00  3.000000e+00
##    HGF_k_RASK      tau_RASK    BRAF_n_MEK    BRAF_k_MEK    PAK1_n_MEK
##  5.156353e-07  0.000000e+00  3.000000e+00  7.505019e-03  3.000000e+00
##    PAK1_k_MEK       tau_MEK    ERK_n_BRAF    ERK_k_BRAF   RASK_n_BRAF
##  0.000000e+00  0.000000e+00  3.000000e+00  5.244096e-02  3.000000e+00
##   RASK_k_BRAF   PAK1_n_BRAF   PAK1_k_BRAF    PLX_n_BRAF    PLX_k_BRAF
##  0.000000e+00  3.000000e+00  9.269049e-02  3.000000e+00  3.603077e-06
##      tau_BRAF  ERK_n_RSKp90  ERK_k_RSKp90    tau_RSKp90    TAK1_n_JNK
##  0.000000e+00  3.000000e+00  9.381897e-01  1.471951e-02  3.000000e+00
##    TAK1_k_JNK    IRS1_n_JNK    IRS1_k_JNK    M3K1_n_JNK    M3K1_k_JNK
##  0.000000e+00  3.000000e+00  3.617738e-05  3.000000e+00  0.000000e+00
##    TNFa_n_JNK    TNFa_k_JNK   MP2K4_n_JNK   MP2K4_k_JNK       tau_JNK
##  3.000000e+00  8.352255e-02  3.000000e+00  0.000000e+00  0.000000e+00
##    AKT_n_PAK1    AKT_k_PAK1   PI3K_n_PAK1   PI3K_k_PAK1      tau_PAK1
##  3.000000e+00  5.496838e-01  3.000000e+00  0.000000e+00  0.000000e+00
##    TAK1_n_p38    TAK1_k_p38   MP2K4_n_p38   MP2K4_k_p38    TBK1_n_p38
##  3.000000e+00  2.582269e-01  3.000000e+00  0.000000e+00  3.000000e+00
##    TBK1_k_p38       tau_p38 TGFRb_n_SMAD2 TGFRb_k_SMAD2     tau_SMAD2
##  8.704423e-01  1.299676e-02  3.000000e+00  0.000000e+00  0.000000e+00
##    AKT_n_GSK3    AKT_k_GSK3 RSKp90_n_GSK3 RSKp90_k_GSK3      tau_GSK3
##  3.000000e+00  0.000000e+00  3.000000e+00  0.000000e+00  0.000000e+00
##   TAK1_n_IkBa   TAK1_k_IkBa    IKK_n_IkBa    IKK_k_IkBa      tau_IkBa
##  3.000000e+00  4.080694e-01  3.000000e+00  7.350341e-01  1.539245e-02
##    JNK_n_cJun    JNK_k_cJun      tau_cJun
##  3.000000e+00  0.000000e+00  0.000000e+00
```

Similar to the above cell-line, we can build a model for each of the cell lines. This is very time consuming, therefore we just load the optimised parameters from the paper.

```
optimised_parameters <- read_delim("./data/tutorial_3/allModelsParameters.txt",delim = "\t")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   cell_line = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

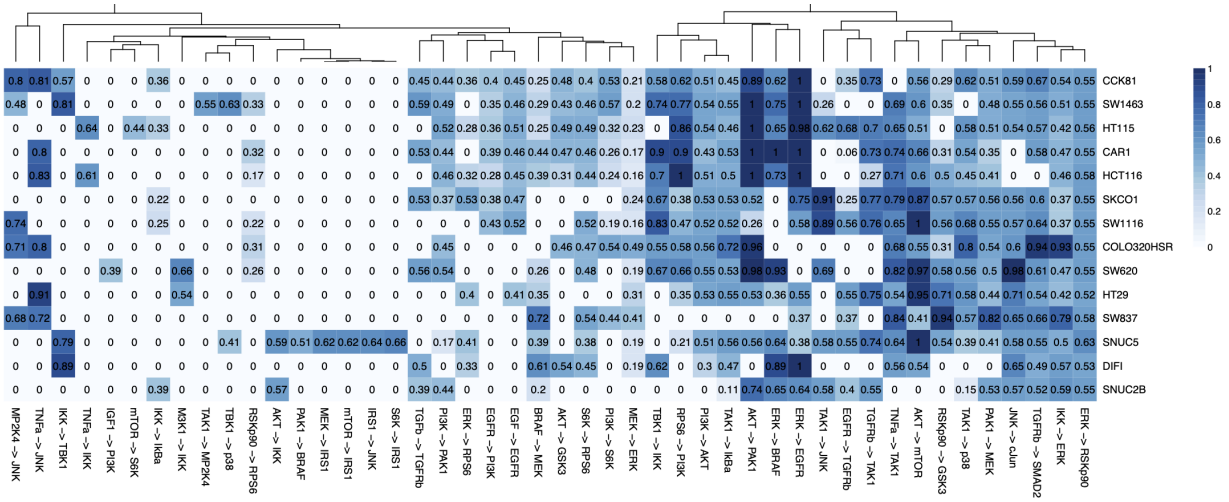
Let's check edge and node parameters.

```
edge_id <- grep("_k_",colnames(optimised_parameters))
edge_parameters_HM <-  optimised_parameters[edge_id]%>% as.matrix()
rownames(edge_parameters_HM) <- optimised_parameters$Cell_line
```

```
## Warning: Unknown or uninitialised column: 'Cell_line'.
```

```
# heatmap(edge_parameters_HM, main = "Cell line edge parameters")
```

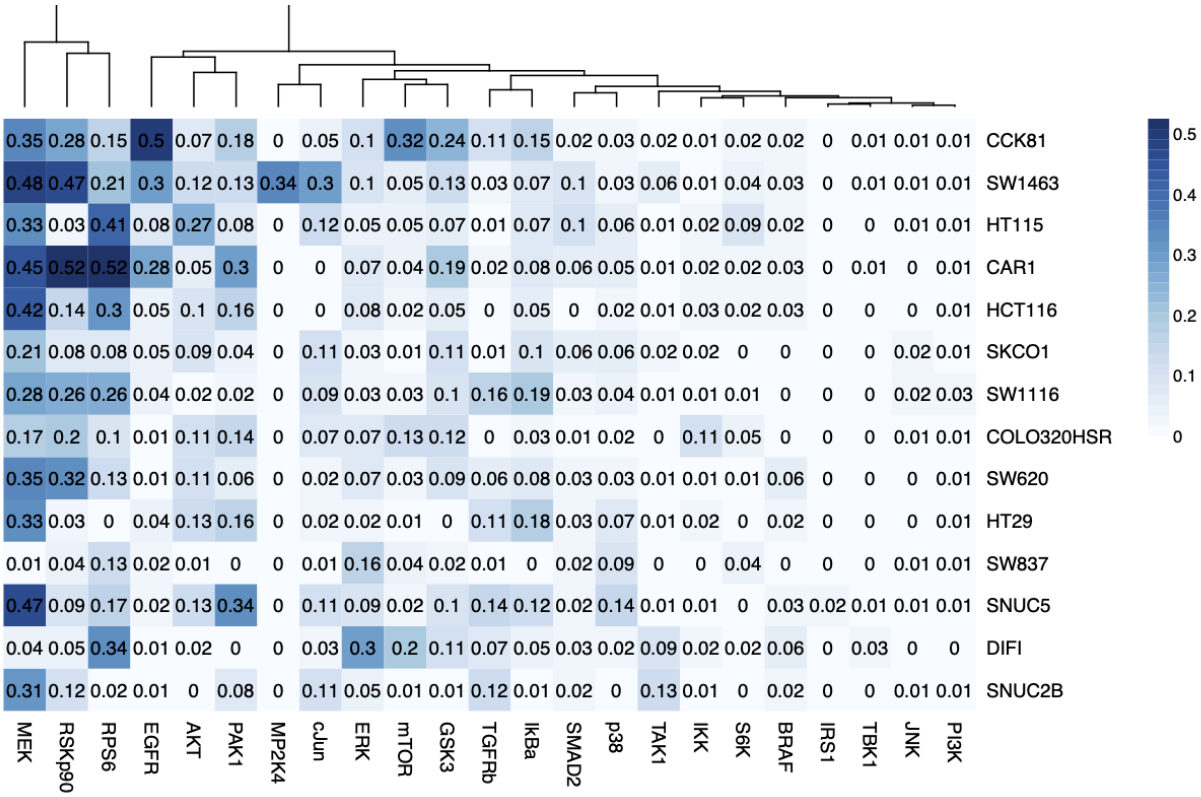
```
knitr::include_graphics("./data/tutorial_3/parHeatmap_k.png")
```




```
node_id <- grep("tau_",colnames(optimised_parameters))
node_parameters_HM <-  optimised_parameters[node_id]%>% as.matrix()
rownames(node_parameters_HM) <- optimised_parameters$Cell_line
```

```
## Warning: Unknown or uninitialised column: 'Cell_line'.
```

```
# heatmap(node_parameters_HM, main = "Cell line node parameters")
knitr::include_graphics("./data/tutorial_3/parHeatmap_tau.png")
```



The level of edge and node parameters differs across the cell-lines.

PART III: Associate model parameters and drug response

Which model parameters correlates with drug response IC50 ?

```
# first we need to remove the parameters, that are zero across all the models
zero_pars <- names(which(colMeans(optimised_parameters[,-1]) == 0))

# join the IC50 data and network model parameters based on cell_lines
drug_model_data <- optimised_parameters %>% select(-zero_pars) %>%
  gather(parameter,par_value,-cell_line) %>%
  left_join(IC50 %>% gather(drug, IC50,-cell_line),by = "cell_line")


# for each drug and each parameter compute the correlation coefficient

corr_data <- drug_model_data %>% group_by(drug, parameter) %>%
  summarise(corr_par_drug = cor(par_value,IC50,use = "complete.obs"))
```

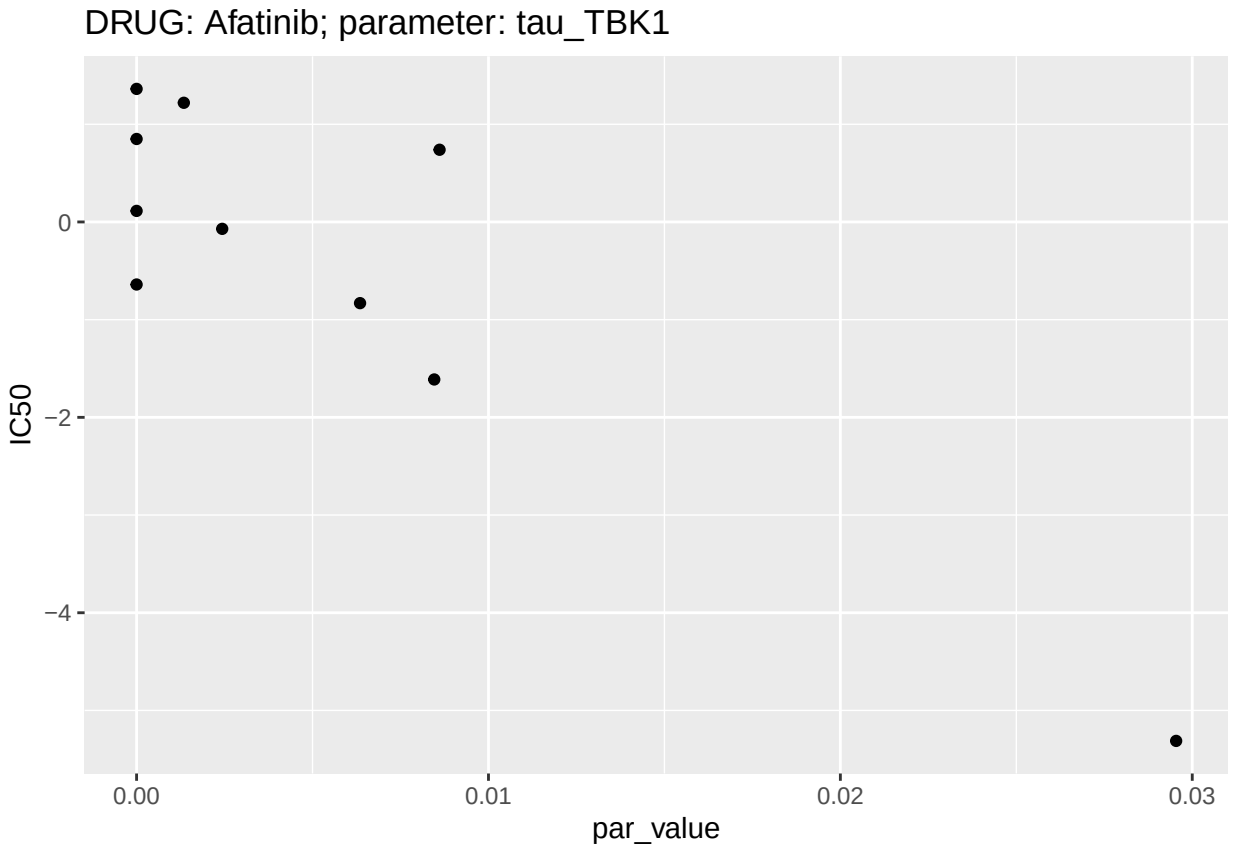
Let's show some correlation

```
corr_data %>% arrange(desc(abs(corr_par_drug))) %>% print(.,n=25)
```

```
## # A tibble: 2,070 x 3
## # Groups:   drug [30]
##    drug                parameter     corr_par_drug
##    <chr>               <chr>                 <dbl>
##  1 Afatinib            tau_TBK1             -0.896
##  2 Afatinib            tau_ERK              -0.875
##  3 BMS-754807          tau_RPS6              0.861
##  4 Gefitinib           tau_ERK              -0.859
##  5 OSI-906             tau_RPS6              0.846
##  6 Afatinib            PAK1_k_MEK            0.834
##  7 RDEA119             IKK_k_ERK             0.818
##  8 Gefitinib           tau_TBK1             -0.813
##  9 SB590885            tau_IkBa             -0.799
## 10 Afatinib (rescreen) tau_TBK1             -0.784
## 11 CI-1040             PI3K_k_S6K            0.777
## 12 CI-1040             M3K1_k_IKK           -0.776
## 13 Afatinib (rescreen) tau_ERK              -0.762
## 14 HG-5-88-01          ERK_k_BRAF            0.756
## 15 HG-5-88-01          ERK_k_RPS6           -0.754
## 16 PD-0325901          AKT_k_GSK3            0.732
## 17 RDEA119             PI3K_k_S6K            0.731
## 18 PD-0325901          PI3K_k_S6K            0.721
## 19 SB590885            RSKp90_k_GSK3        -0.717
## 20 (5Z)-7-Oxozeaenol   PI3K_k_S6K            0.717
## 21 RDEA119             MP2K4_k_JNK           0.712
## 22 OSI-906             ERK_k_BRAF            0.708
## 23 Gefitinib           tau_MEK               0.708
## 24 (5Z)-7-Oxozeaenol   MP2K4_k_JNK           0.707
## 25 THZ-2-102-1         tau_SMAD2             0.706
## # ... with 2,045 more rows
```

```
drug_model_data %>% filter(drug=="Afatinib",parameter=="tau_TBK1") %>%
  ggplot() + geom_point(aes(par_value,IC50)) +
  ggtitle("DRUG: Afatinib; parameter: tau_TBK1")
```

```
## Warning: Removed 4 rows containing missing values (geom_point).
```



Think what the problem might be here?
We have only 14 cell-lines, therefore each of the correlations between model parameter and drug IC50 is based on 14 data points. There are 31 drugs and 89 model parameters, which results in `31*89=2759` tests.

Also this is only a single parameter - single drug association. It is possible, that the existence of multiple edges makes a cell-line sensitive/resistant. Therefore (Eduati et al) derived linear models, that funds multiple parameters at the same time.

```
knitr::include_graphics("./data/tutorial_3/Eduatietal_Figure5.png")
```

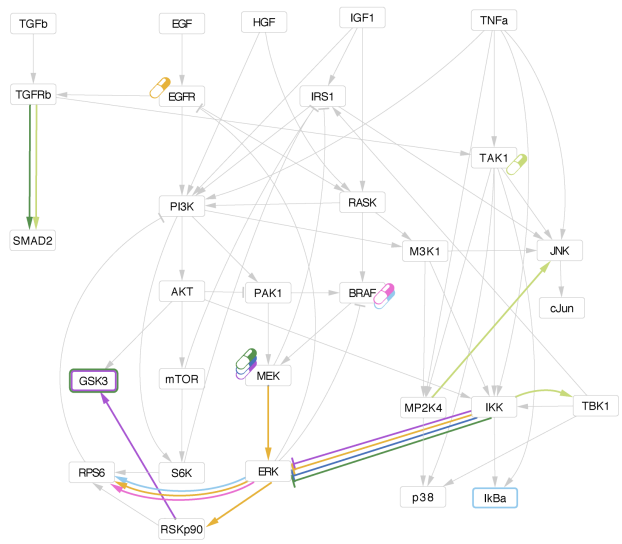
Figure 5

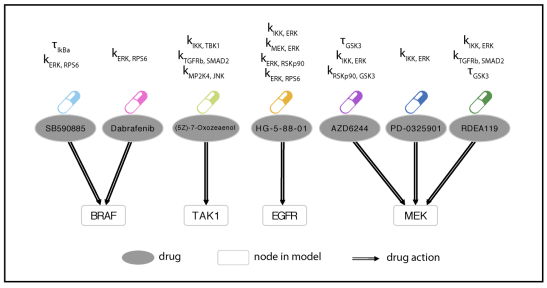
A

Drug	Associated model parameter	Targets	Associated genomic alterations
OSI-906	k ERK,RSKp90	IGF1R	MYC->PDK1,SMAD2
OSI-906	k MEK,ERK	IGF1R	MYC->PDK1,SMAD2
BMS-345541	τ cJun	IKBKB	PARK2-|cJun; MYC->PDK1,SMAD2
HG-5-88-01	k IKK,ERK	EGFR	
OSI-906	τ RPS6	IGF1R	MYC->PDK1,SMAD2
BMS-345541	k EGFR,PI3K	IKBKB	PARK2-|cJun; MYC->PDK1,SMAD2
SB590885	τ IkBa	BRAF	
AZD6244	τ GSK3	MEK	
HG-5-88-01	k MEK,ERK	EGFR	
PD-0325901	k IKK,ERK	MEK	
HG-5-88-01	k ERK,RSKp90	EGFR	
BMS-754807	τ RPS6	IGF1R	MYC->PDK1,SMAD2
SB590885	k ERK,RPS6	BRAF	
(5Z)-7-Oxozeaenol	k IKK,TBK1	MAP3K7 (TAK1)	
BMS-345541	k PI3K,PAK1	IKBKB	PARK2-|cJun; MYC->PDK1,SMAD2
AZD6244	k IKK,ERK	MEK	
AZD6244	k RSKp90,GSK3	MEK	
(5Z)-7-Oxozeaenol	k TGFRb,SMAD2	MAP3K7 (TAK1)	
RDEA119	k IKK,ERK	MEK	
RDEA119	k TGFRb,SMAD2	MEK	
(5Z)-7-Oxozeaenol	k MP2K4,JNK	MAP3K7 (TAK1)	
Dabrafenib	k ERK,RPS6	BRAF	
THZ-2-102-1	τ cJun	CDK7->MEK,PAK1	MYC->PDK1,SMAD2
RDEA119	τ GSK3	MEK	
HG-5-88-01	k ERK,RPS6	EGFR	
SB590885	k RSKp90,GSK3	BRAF	
Afatinib	τ ERK	EGFR; ERBB2->EGFR,SHC1,PI3K,GRB2,PAK1	MYC->PDK1,SMAD2
(5Z)-7-Oxozeaenol	k PI3K,S6K	MAP3K7 (TAK1)	
Dabrafenib	k TGFRb,SMAD2	BRAF	
BMS-345541	τ mTOR	IKBKB	PARK2-|cJun; MYC->PDK1,SMAD2
BMS-345541	k EGFR,TGFRb	IKBKB	PARK2-|cJun; MYC->PDK1,SMAD2
PD-0325901	k ERK,RPS6	MEK	
BMS-345541	τ RPS6	IKBKB	PARK2-|cJun; MYC->PDK1,SMAD2
(5Z)-7-Oxozeaenol	τ EGFR	MAP3K7 (TAK1)	
PD-0325901	k AKT,GSK3	MEK	
CI-1040	k PI3K,S6K	MEK	
HG-5-88-01	k ERK,BRAF	EGFR	
(5Z)-7-Oxozeaenol	k TAK1,JNK	MAP3K7 (TAK1)	
Afatinib	τ RPS6	EGFR; ERBB2->EGFR,SHC1,PI3K,GRB2,PAK1	MYC->PDK1,SMAD2
PD-0325901	k PI3K,S6K	MEK	
Dabrafenib	k PI3K,AKT	BRAF	
BMS-345541	k EGF,EGFR	IKBKB	PARK2-|cJun; MYC->PDK1,SMAD2
Trametinib	k IKK,ERK	MEK	
RDEA119	k IKK,IkBa	MEK	
BMS-345541	k AKT,PAK1	IKBKB	PARK2-|cJun; MYC->PDK1,SMAD2
AZD6244	k AKT,GSK3	MEK	
AZD6244	τ mTOR	MEK	
SB590885	k ERK,BRAF	BRAF	
PD-0325901	τ IkBa	MEK	
Trametinib	k TGFRb,SMAD2	MEK	

Weight

B





drug-parameter association weight