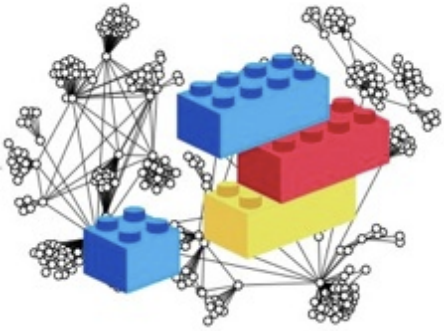
Iterative Network-Guided Connectivity mapping

This website and the linked web pages contain functions, scripts and data objects used in the software enclosed to the paper entitled *A semi-supervised approach for refining transcriptional signatures of drug response and repositioning predictions*, by Francesco Iorio et al, submitted as research paper to *PLoS ONE*.



Source code and supplementary data

Supplementary Dataset DS1: cMap Drugs prototype ranked lists
Compressed tab delimited txt file containing the 'prototype ranked lists' of genes for all the drug contained in the connectivity map dataset, computed as described in Iorio et al, PNAS 2010.
SuppDataset_SD1_DRUG_PRLS_txt.zip

How to reproduce results and figures presented in the manuscript?

To start:

Make sure you have R installed. You can download it from http://cran.ma.imperial.ac.uk/

We strongly recommend to install and use the RStudio interface to R, downloadable from: http://www.rstudio.com

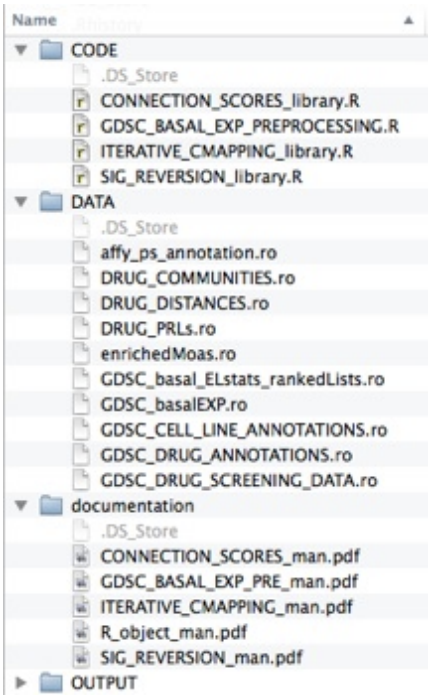
Required libraries:
Make sure you have the following libraries installed (all available on the CRAN repository):
- mixtools
- sROC
- pheatmap
- beeswarm

To install them use the following command from the RStudio console:
`install.packages("[library.name]")`
replacing `[library.name]` with each of the library names listed above, in turn.

Working directory creation:

Download and unzip the following compressed folder:
IorioEtAl_R_code_and_objects.zip

Once uncompressed, the content of this folder and its sub-folders should not be changed. Files in the `OUTPUT` subfolder (initially empty) can be moved and/or deleted.



Working directory setup:
To set the working directory to IorioEtAl_R_code_and_objects use the following command from the RStudio console:
`setwd('[path]/IorioEtAl_R_code_and_objects')`
replacing `[path]` with the path of the `IorioEtAl_R_code_and_objects` directory.

Ready to go! 

To reproduce results and figure presented in our manuscript execute the commands contained in the following pipelines:

Network guided iterative connectivity mapping pipeline

Pipeline for predictive ability validation through the signature reversion paradigm

Iterative network guided connectivity mapping pipeline

A semi-supervised approach for refining transcriptional signatures of drug response and repositioning predictions

(Supplementary Material and Methods: Supplementary Code)

Francesco Iorio - 24 Aprile 2014

Importing libraries of functions needed to compute connectivity scores and to run the iterative network guided connectivity mapping pipeline:

```
options(warn = -1)
source("CODE/CONNECTION_SCORES_library.R")
```

```
## Loading required package: boot
## Loading required package: MASS
## Loading required package: segmented
## mixtools package, version 1.0.1, Released January 2014
## This package is based upon work supported by the National Science Foundation under Grant No. SES-0518772.
```

```
source("CODE/ITERATIVE_CMAPPING_library.R")
```

Querying the drug network described in Iorio et al (PNAS 2010) using paclitaxel as seed compound:

```
paclitaxelNeighborhood <- DNquery(seed = "paclitaxel", distTh = 0.8065, printToFile = FALSE)
```

Analysing the paclitaxel neighborhood in the drug network (main figure 2 and supplementary table 1):

```
print(paclitaxelNeighborhood[, c("D", "quantile %", "C id", "Adj p-val")])
```

```
##                         D        quantile % C id            Adj p-val
## demecolcine       0.70572 0.449954794261324   48
## 5252917           0.73307  0.82631885114346   48 0.000236989288084179
## pararosaniline    0.75068  1.26062101237493   62
## MG-132            0.75678  1.46398843106884   40
## parbendazole      0.75768  1.49517688643431   90
## celastrol         0.75776  1.49891482865039   40   0.0182249280466375
## 5224221           0.76625  1.84537534780384   40  0.00169840589958832
## splitomicin       0.76715  1.88836168328883   24
## diltiazem         0.77064  2.05528416537591   73
## cytochalasin_B    0.77165  2.10878346334364  100
## fenbendazole      0.77847   2.5058230131085   69
## gefitinib         0.78277  2.79411180652411   60
## suloctidil        0.78504  2.95157262237672   34
## chlortetracycline 0.78788  3.16907413507521   42
## PHA-00665752       0.7887   3.2295820746981   19
## rotenone          0.78983  3.32606770815082   62   0.0694962846867603
## promethazine       0.7901  3.34697682242205   90    0.331175834577628
## ionomycin         0.79515  3.77742423074317   40  0.00251447929397367
## cyproheptadine    0.79774  4.02120814964852   40 0.000400440318057552
## lynestrenol       0.80095  4.35166560368935   40 9.79332311504551e-05
## perhexiline       0.80485   4.7877199253346  100    0.375201411612571
## terfenadine        0.8058  4.90114310945396   34   0.0559034992586342
```

Listing drug communities enriched in the paclitaxel neighborhood (adjusted p-value < 0.05):

```
enriched_cid <- unique(paclitaxelNeighborhood[which(as.numeric(as.character(paclitaxelNeighborhood[,
    "Adj p-val"])) < 0.05), "C id"])
print(enriched_cid)
```

```
## [1] 48 40
## Levels: 100 19 24 34 40 42 48 60 62 69 73 90
```

Listing modes-of-action/Drug-features over-represented in the drug communities enriched in the paclitaxel neighborhood:

```
print(as.character(unique(paclitaxelNeighborhood[which(is.element(paclitaxelNeighborhood[,
    "C id"], enriched_cid)), "MOAs"])))
```

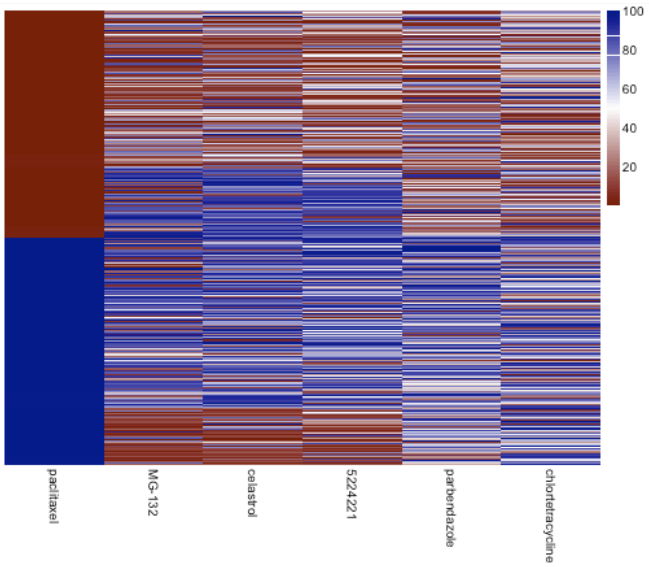
```
## [1] "plant alkaloids // alkaloid"
## [2] "Proteasome inhibitors and UPS modulators // protein synthesis inhibitors (elongation inhibitors) // calcium signal modulators"
```

Deriving paclitaxel/Proteasome-inhibitors consistent/inconsistent signatures (supplementary table 3)

```
P_PI_consistentSig <- DeriveConsistentSignature(seed = "paclitaxel", otherCompounds = c("MG-132",
    "celastrol", "5224221"), PTH = 30, FUZZYNESS = 2, printToFile = FALSE)
P_PI_inconsistentSig <- DeriveInConsistentSignature(seed = "paclitaxel", otherCompounds = c("MG-132",
    "celastrol", "5224221"), PTH = 30, FUZZYNESS = 2, printToFile = FALSE)
```

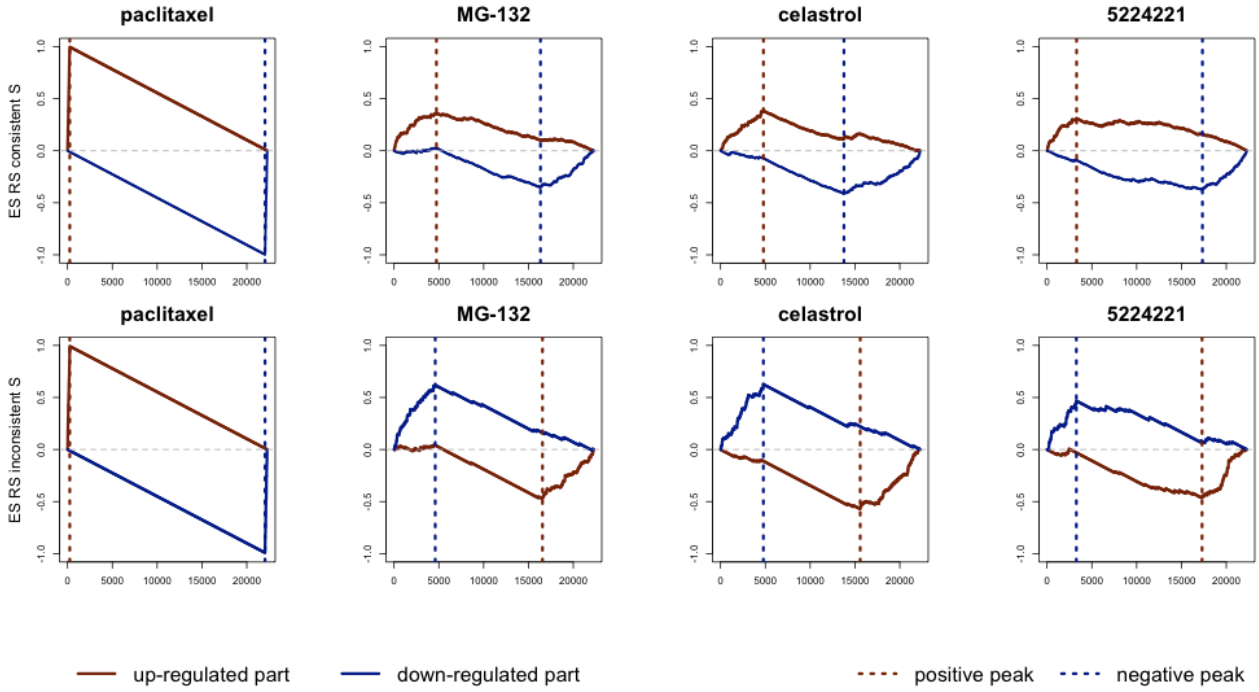
Visualising a heatmap of expression percentiles of the computed signatures along the prototype ranked lists of paclitaxel, the protasome inhibitors contained in its neighbourhood, and 2 microtubule pertubing drugs included for reference (main figure 3 (A)):

```
percHeatMaps(c(as.character(P_PI_consistentSig$seedUPreg$ProbeSets), as.character(P_PI_inconsistentSig$seedUPreg$ProbeSets),
    as.character(P_PI_consistentSig$seedDOWNreg$ProbeSets), as.character(P_PI_inconsistentSig$seedDOWNreg$ProbeSets)),
    seed = "paclitaxel", otherCompounds = c("MG-132", "celastrol", "5224221",
        "parbendazole", "chlortetracycline"), printToFile = FALSE)
```



Visualising the enrichment score running sums for of the paclitaxel/proteasome-inhibitors consistent/inconsistent signatures along the prototype ranked lists of paclitaxel and the protasome inhibitors (supplementary figure SF1):

```
plotRunningSums(P_PI_consistentSig, P_PI_inconsistentSig, seed = "paclitaxel",
    otherCompounds = c("MG-132", "celastrol", "5224221"), printToFile = FALSE)
```



Computing connectivity scores between the prototype ranked lists of all the cMap drugs and the paclitaxel/proteasome-inhibitors consistent/inconsistent signatures (this may take a while):

```
P_PI_consistent_CS <- CS(P_PI_consistentSig, RANKED_LISTS = DRUG_PRLs, show_progress = FALSE)
```

```
## simulating null model
## number of iterations= 123
## done!
## computing connectivity scores
## Done!
```

```
P_PI_inconsistent_CS <- CS(P_PI_inconsistentSig, RANKED_LISTS = DRUG_PRLs, show_progress = FALSE)
```

```
## simulating null model
## number of iterations= 26
## done!
## computing connectivity scores
## Done!
```

Combining the obtained connectivity scores to refine the paclitaxel neighbourhood:

```
first_nb <- combine_2CS(P_PI_consistent_CS, P_PI_inconsistent_CS, printToFile = FALSE,
    fn = "")
```

Visualising paclitaxel 1st refined neighbourhood (main figure 3 (B) and supplementary table 4):

```
id <- which(first_nb[, "cons S fdr %"] < 5 & first_nb[, "incons S fdr %"] <
    5 & first_nb[, "cons S CS"] > 0 & first_nb[, "incons S CS"] > 0)
print(first_nb[id[2:length(id)], c(4, 8, 9)])
```

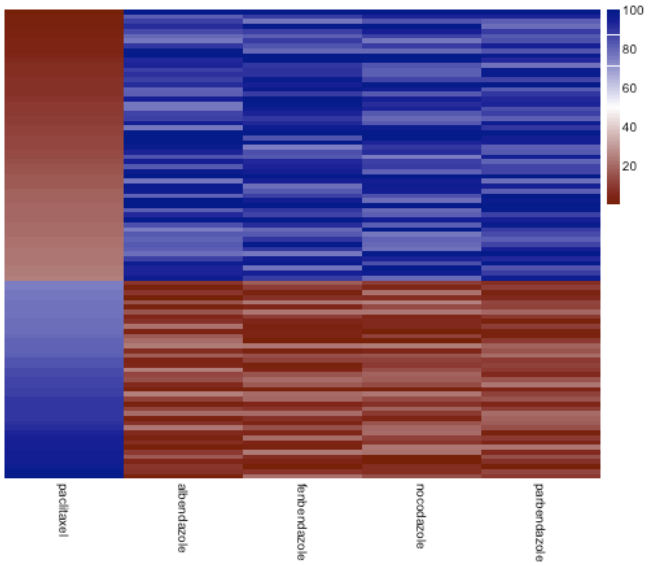
```
##                          cons S NCS incons S NCS avg NCS
## 5252917                        4.306          2.719   3.513
## parbendazole                   4.380          1.771   3.076
## splitomicin                    3.657          2.451   3.054
## fenbendazole                   3.916          1.927   2.922
## gefitinib                      3.532          2.228   2.880
## chlortetracycline              3.429          2.226   2.828
## rotenone                       3.850          1.705   2.778
## glipizide                      3.290          2.185   2.738
## albendazole                    3.240          2.115   2.678
## bromocriptine                  3.167          2.142   2.655
## diltiazem                      3.559          1.739   2.649
## cyproheptadine                 3.620          1.666   2.643
## moroxydine                     2.614          2.471   2.542
## naloxone                       2.936          2.132   2.534
## perhexiline                    3.425          1.596   2.511
## nilutamide                     2.929          1.886   2.407
## hesperetin                     3.169          1.632   2.400
## nocodazole                     2.627          1.980   2.303
## danazol                        2.584          1.788   2.186
## betulinic_acid                 2.712          1.616   2.164
## fluoxetine                     2.722          1.603   2.162
## metolazone                     2.159          2.146   2.153
## hydrastinine                   2.517          1.770   2.144
## practolol                      2.385          1.764   2.074
## genistein                      1.984          2.035   2.010
## monastrol                      2.091          1.779   1.935
## methoxamine                    1.958          1.876   1.917
## primidone                      1.874          1.815   1.844
## 3-hydroxy-DL-kynurenine        1.759          1.921   1.840
## dehydrocholic_acid             1.798          1.854   1.826
## clozapine                      1.933          1.496   1.714
## thioridazine                   1.882          1.533   1.707
## phenazone                      1.739          1.641   1.690
## epiandrosterone                1.744          1.531   1.638
## dihydroergotamine              1.681          1.591   1.636
## chlorphenesin                  1.574          1.531   1.552
```

Deriving a microtubule stabilising signature (supplementary table 5):

```
MS_sig <- DeriveMSTSignature(seed = "paclitaxel", otherCompounds = c("albendazole",
    "fenbendazole", "nocodazole", "parbendazole"), FUZZYNESS = 4, printToFile = FALSE)
```

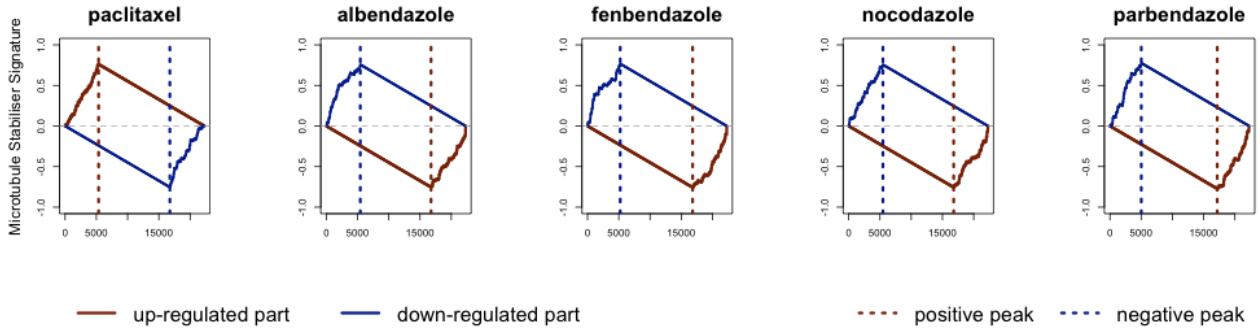
Visualising a heatmap of expression percentiles of the microtubule stabilising signature along the prototype ranked lists of paclitaxel and the three benzimidazoles contained in his first refined neighbourhood (figure 3 (C)):

```
percHeatMaps(c(as.character(MS_sig$seedUPreg$ProbeSets), rev(as.character(MS_sig$seedDOWNreg$ProbeSets))),
    seed = "paclitaxel", otherCompounds = c("albendazole", "fenbendazole", "nocodazole",
        "parbendazole"), printToFile = FALSE)
```



Visualising the enrichment score running sums of the microtubule stabilising signature along the prototype ranked lists of paclitaxel and the recovered benzimidazoles (supplementary figure SF1):

```
plotRunningSumsMST(MS_sig, seed = "paclitaxel", otherCompounds = c("albendazole",
    "fenbendazole", "nocodazole", "parbendazole"), printToFile = FALSE)
```



Computing connectivity scores between the prototype ranked lists of all the cMap drugs and the microtubule stabilising signature (this may take a while):

```
MST_CS <- CS(MS_sig, RANKED_LISTS = DRUG_PRLs, show_progress = FALSE)
```

```
## simulating null model
## number of iterations= 34
## done!
## computing connectivity scores
## Done!
```

Combining the obtained connectivity scores to finally refine the paclitaxel neighbourhood:

```
previousConnections <- rownames(first_nb)[which(as.numeric(first_nb[, "avg NCS"]) >
    0 & as.numeric(first_nb[, "cons S NCS"]) > 0 & as.numeric(first_nb[, "incons S NCS"]) >
    0 & as.numeric(first_nb[, "cons S fdr %"]) < 5 & as.numeric(first_nb[, "incons S fdr %"]) <
    5)]

final_nb <- combine_3CS(P_PI_consistent_CS, P_PI_inconsistent_CS, MST_CS, previousNeighBr = previousConnections,
    printToFile = TRUE, fn = "final")
```

Visualising the final refined neighbourhood of paclitaxel (main figure 3 (D), table 1 and supplementary table 4):

```
print(final_nb[2:nrow(final_nb), 5])
```

```
##               glipizide             splitomicin              fluoxetine
##       "2.19989686915533"       "2.04683202514754"       "1.96842993269957"
##               metolazone                  5252917                diltiazem
##        "1.8779827209263"       "1.80526353389624"       "1.79016488605669"
##           betulinic_acid              nilutamide              moroxydine
##       "1.77586391018833"       "1.75347275374926"       "1.61659219321536"
##              perhexiline                gefitinib           bromocriptine
##       "1.57596141207419"       "1.54301919561025"       "1.52192398075313"
##                rotenone                  danazol          epiandrosterone
```

```
##       "1.4943673200471"      "1.47509920854964"      "1.46695129207344"
##      chlortetracycline            thioridazine          cyproheptadine
##      "1.43213750357412"      "1.38190858522147"      "1.36622967073937"
##             hydrastinine                genistein                naloxone
##      "1.35856721388798"      "1.34265532302282"      "1.31001661205284"
##               monastrol              hesperetin                primidone
##      "1.27449217019113"      "1.26817333429531"      "1.26195005404283"
##              phenazone               clozapine      dehydrocholic_acid
##      "1.21488168491939"      "1.02664929830309"     "0.982238982351692"
##            methoxamine 3-hydroxy-DL-kynurenine               practolol
##     "0.961273540522456"     "0.914417894562068"     "0.885856416275826"
##          chlorphenesin       dihydroergotamine           parbendazole
##     "0.729772935806824"     "0.550035080493067"   "-0.0926467043760503"
##           fenbendazole              albendazole             nocodazole
##    "-0.159860724573494"    "-0.306185924261493"    "-0.555568320480004"
```

Pipeline for predictive ability validation through the signature reversion paradigm

A semi-supervised approach for refining transcriptional signatures of drug response and repositioning predictions

(Supplementary Material and Methods: Supplementary Code)

Francesco Iorio - 24 Aprile 2014

Importing libraries of functions needed to compute connectivity scores and to run the signature reversion pipeline:

```
options(warn = -1)
source("CODE/ITERATIVE_CMAPPING_library.R")
source("CODE/CONNECTION_SCORES_library.R")
```

```
## Loading required package: boot
## Loading required package: MASS
## Loading required package: segmented
## mixtools package, version 1.0.1, Released January 2014
## This package is based upon work supported by the National Science Foundation under Grant No. SES-0518772.
```

```
source("CODE/SIG_REVERSION_library.R")
```

Loading AffyMetrix probe-set annotations:

```
load("DATA/affy_ps_annotation.ro")
```

Loading the GDSC drug screening data and drug annotations for docetaxel, vinorelbine and paclitaxel:

```
load("DATA/GDSC_DRUG_SCREENING_DATA.ro")
load("DATA/GDSC_DRUG_ANNOTATIONS.ro")
```

Loading the annotation file for the GDSC cell lines:

```
load("DATA/GDSC_CELL_LINE_ANNOTATIONS.ro")
```

Load the basal expression statistic ranked lists:

```
load("DATA/GDSC_basal_ELstats_rankedLists.ro")
```

Or, **alternatively**, recomputing them by executing the following commands:

```
source("DATA/GDSC_basal_ELstats_rankedLists.ro")
ELstats <- EL_statistics(basalEXP)
gdsc_basal_ELstats_rankedLists <- basalRanked_lists(ELstats)
```

Generating the gene signatures to be tested individually and in combinations (note that the functions are called with the default parameters):

1. the paclitaxel optimal signature (supplementary table 2)

2. the paclitaxel/protasome-inh. consistent signature

3. the paclitaxel/protasome-inh. inconsistent signature (supplementary table 3)

4. the microtubule stabilising signature (supplementary table 5)

```
paclitaxel_opt_sig <- DeriveSingleSignature()
paclitaxel_PI_con_sig <- DeriveConsistentSignature()
paclitaxel_PI_incon_sig <- DeriveInConsistentSignature()
MI_stab_sig <- DeriveMSTSignature()
```

Converting microarray probe-sets to gene symbols:

```
paclitaxel_opt_sig$seedUPreg$ProbeSets <- affy_ps_annotation[as.character(paclitaxel_opt_sig$seedUPreg$ProbeSets),
    1]
paclitaxel_opt_sig$seedDOWNreg$ProbeSets <- affy_ps_annotation[as.character(paclitaxel_opt_sig$seedDOWNreg$ProbeSets),
    1]
paclitaxel_PI_con_sig$seedUPreg$ProbeSets <- affy_ps_annotation[as.character(paclitaxel_PI_con_sig$seedUPreg$ProbeSets),
    1]
paclitaxel_PI_con_sig$seedDOWNreg$ProbeSets <- affy_ps_annotation[as.character(paclitaxel_PI_con_sig$seedDOWNreg$ProbeSets),
    1]
paclitaxel_PI_incon_sig$seedUPreg$ProbeSets <- affy_ps_annotation[as.character(paclitaxel_PI_incon_sig$seedUPreg$ProbeSets),
    1]
paclitaxel_PI_incon_sig$seedDOWNreg$ProbeSets <- affy_ps_annotation[as.character(paclitaxel_PI_incon_sig$seedDOWNreg$ProbeSets),
    1]
MI_stab_sig$seedUPreg$ProbeSets <- affy_ps_annotation[as.character(MI_stab_sig$seedUPreg$ProbeSets),
    1]
MI_stab_sig$seedDOWNreg$ProbeSets <- affy_ps_annotation[as.character(MI_stab_sig$seedDOWNreg$ProbeSets),
    1]
```

Computing connectivity scores for all the GDSC cell lines and the individual signatures (this may take a while):

```
PACLITAXEL <- CS(paclitaxel_opt_sig, gdsc_basal_ELstats_rankedLists, show_progress = FALSE)
```

```
## simulating null model
## number of iterations= 64
## done!
## computing connectivity scores
## Done!
```

```
P_PI_CON <- CS(paclitaxel_PI_con_sig, gdsc_basal_ELstats_rankedLists, show_progress = FALSE)
```

```
## simulating null model
## number of iterations= 686
## done!
## computing connectivity scores
## Done!
```

```
P_PI_INCON <- CS(paclitaxel_PI_incon_sig, gdsc_basal_ELstats_rankedLists, show_progress = FALSE)
```

```
## simulating null model
## number of iterations= 80
## done!
## computing connectivity scores
## Done!
```

```
MI <- CS(MI_stab_sig, gdsc_basal_ELstats_rankedLists, show_progress = FALSE)
```

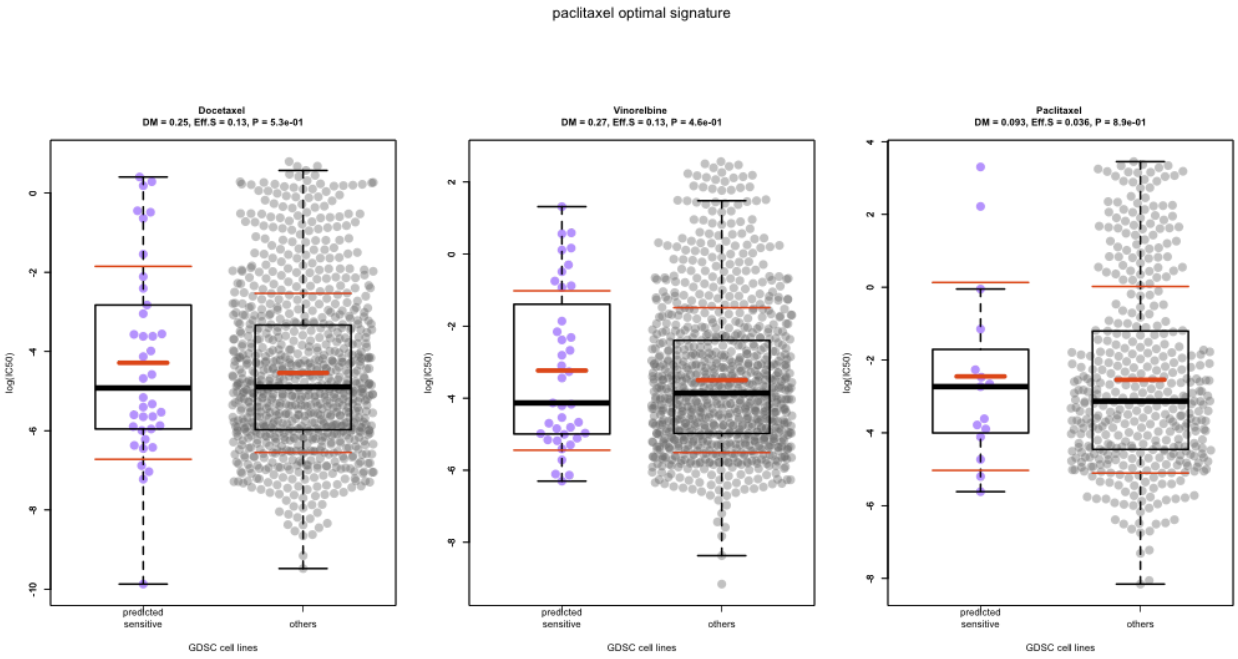
```
## simulating null model
## number of iterations= 33
## done!
## computing connectivity scores
## Done!
```

Selecting an fdr threshold and the drugs to be tested:

```
th <- 0.3
DRUGS <- rownames(DRUG_PROPS)
```

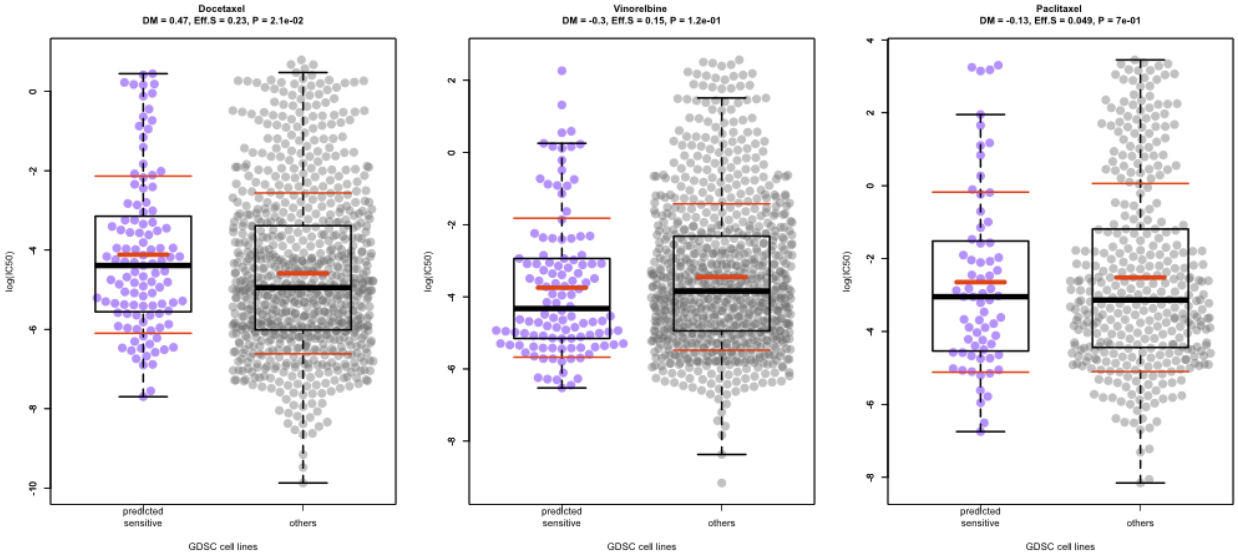
Testing the predictive ability of the individual signatures and storing performance scores (supplementary figure SF6) :

```
totRES <- test_pred_ability(list(PACLITAXEL), DRUGS = DRUGS, mainTitle = "paclitaxel optimal signature")
```



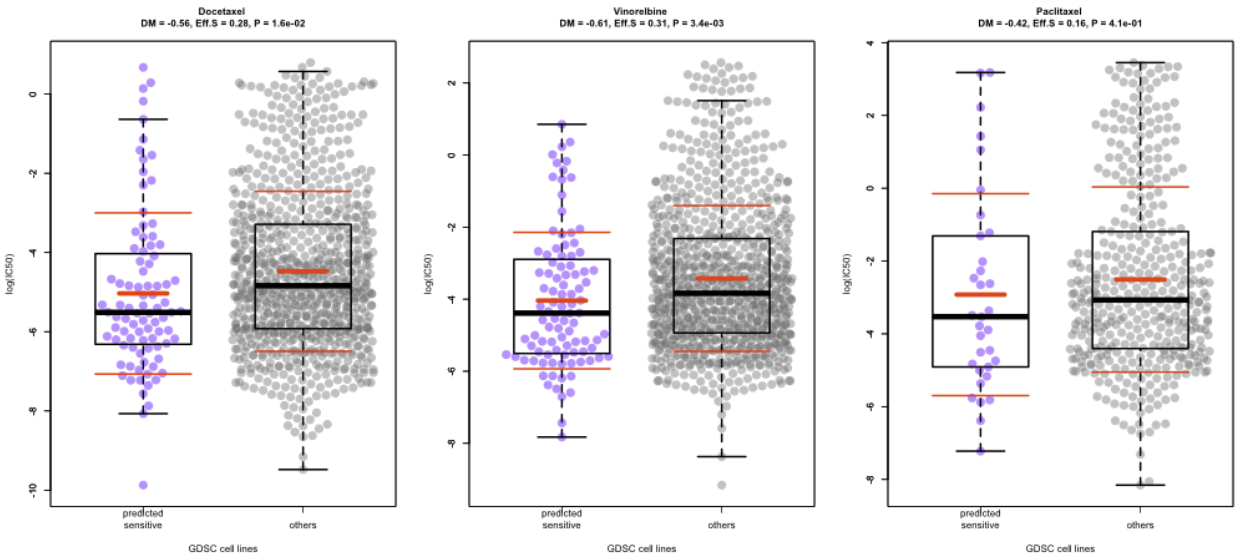

```
totRES <- rbind(totRES, test_pred_ability(list(P_PI_CON), DRUGS = DRUGS, mainTitle = "paclitaxel/proteasome-inh. consistent signature"))
```

paclitaxel/proteasome-inh. consistent signature

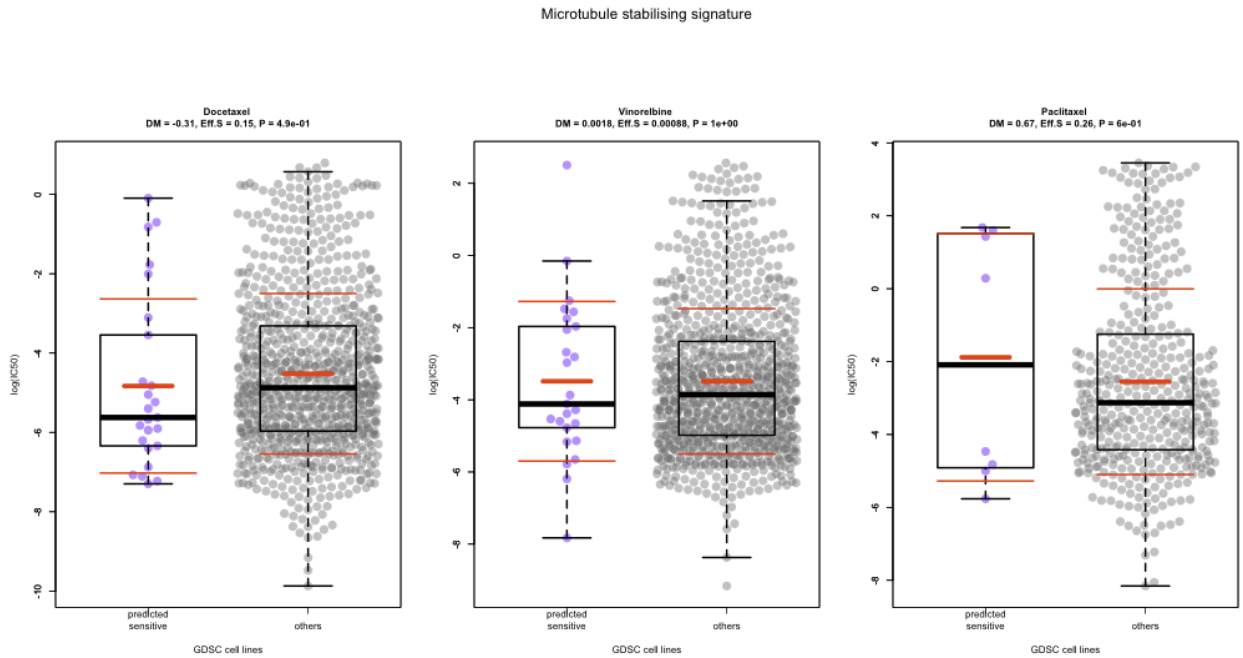



```
totRES <- rbind(totRES, test_pred_ability(list(P_PI_INCON), DRUGS = DRUGS, mainTitle = "paclitaxel/proteasome-inh. inconsistent signature"))
```

paclitaxel/proteasome-inh. inconsistent signature

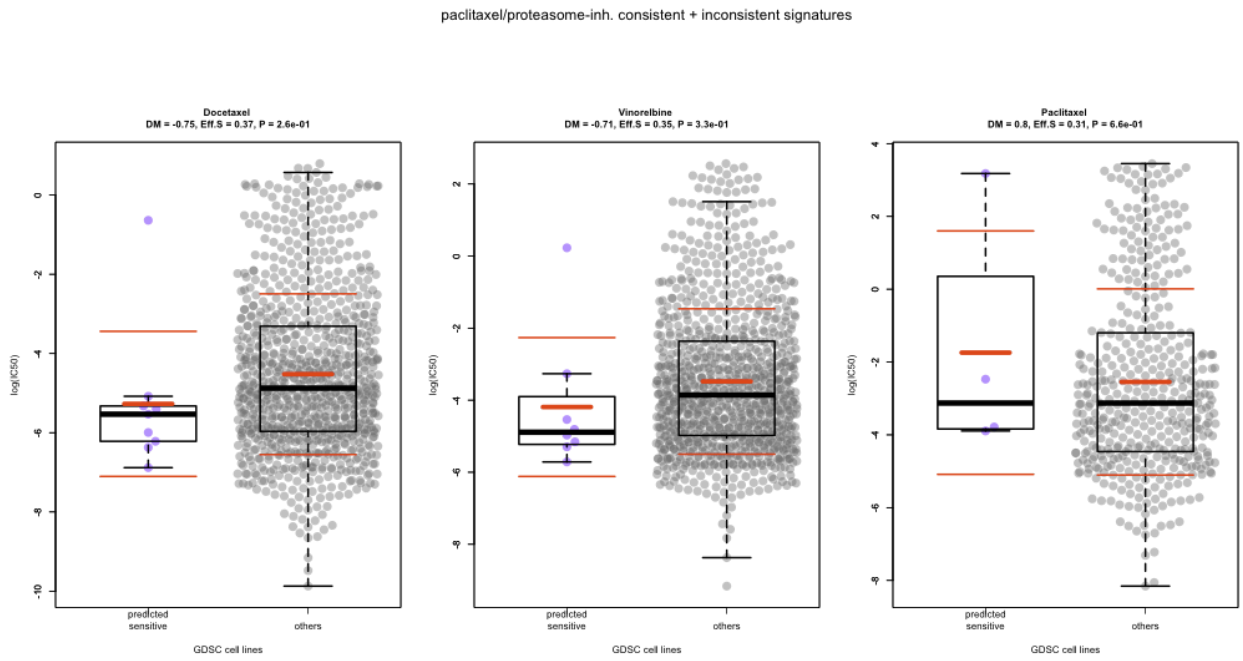



```
totRES <- rbind(totRES, test_pred_ability(list(MI), DRUGS = DRUGS, mainTitle = "Microtubule stabilising signature"))
```



Testing the predictive ability of the combined signatures and storing performance scores (main figure 5 (B) and supplementary figure SF6):

```
totRES <- rbind(totRES, test_pred_ability(list(P_PI_CON, P_PI_INCON), DRUGS = DRUGS,
    mainTitle = "paclitaxel/proteasome-inh. consistent + inconsistent signatures"))
```

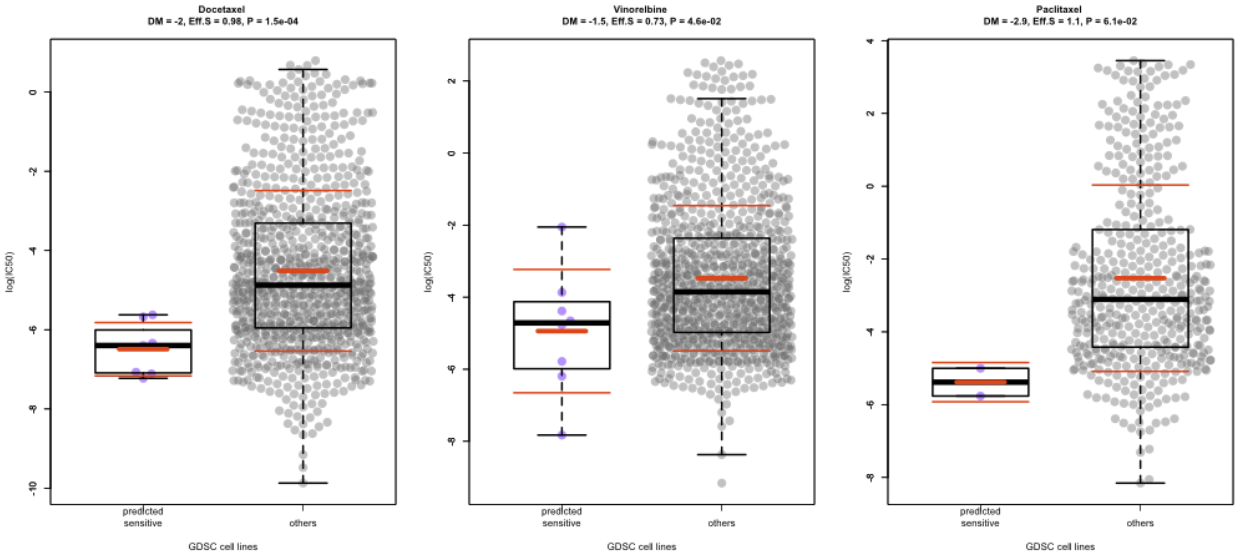



```
totRES <- rbind(totRES, test_pred_ability(list(P_PI_CON, MI), DRUGS = DRUGS,
    mainTitle = "paclitaxel/proteasome-inh. consistent + microtubule stabilising signature"))
```

paclitaxel/proteasome-inh. consistent + microtubule stabilising signature

```
totRES <- rbind(totRES, test_pred_ability(list(P_PI_INCON, MI), DRUGS = DRUGS,
    mainTitle = "paclitaxel/proteasome-inh. inconsistent + microtubule stabilising signature"))
```

paclitaxel/proteasome-inh. inconsistent + microtubule stabilising signature




```
totRES <- rbind(totRES, test_pred_ability(list(P_PI_CON, P_PI_INCON, MI), DRUGS = DRUGS,
    mainTitle = "paclitaxel/proteasome-inh. consistent + inconsistent + microtubule stabilising signature"))
```

paclitaxel/proteasome-inh. consistent + inconsistent + microtubule stabilising signature

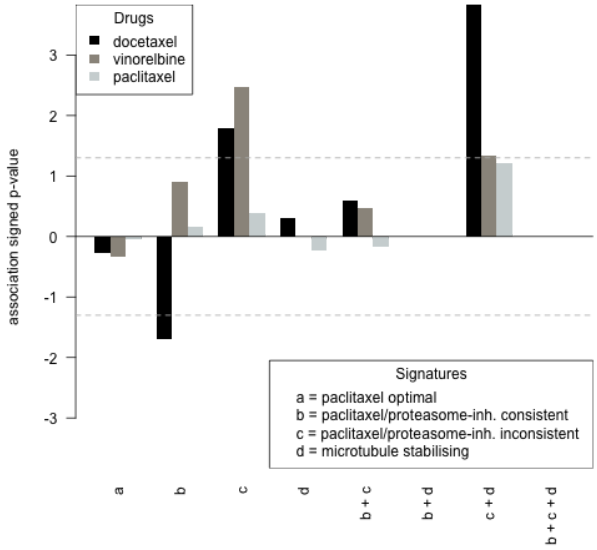
Summarising the predictive performances for all the tested signatures and signature combinations:

```
performanceMatrix <- matrix(NA, nrow = length(unique(totRES$"used signature(s)")),
    ncol = 3, dimnames = list(unique(totRES$"used signature(s)"), unique(totRES$drug)))
signatures <- unique(totRES$"used signature(s)")


for (i in 1:length(signatures)) {
    idxs <- which(totRES$"used signature(s)" == signatures[i])
    performanceMatrix[i, ] <- log10(as.numeric(as.character(totRES[idxs, "p-val"]))) *
        sign(as.numeric(as.character(totRES[idxs, "deltaMean"])))
}
```

Plotting the performance summary (supplementary figure SF7):

```
par(las = 2)
barplot(t(performanceMatrix), ylab = "association signed p-value", names.arg = c("a",
    "b", "c", "d", "b + c", "b + d", "c + d", "b + c + d"), beside = TRUE, col = c("#000000",
    "#8B8378", "#C1CDCD"), ylim = c(-max(abs(c(performanceMatrix)), na.rm = TRUE),
    max(abs(c(performanceMatrix)), na.rm = TRUE)), border = NA)
abline(h = -log10(0.05), lty = 2, col = "darkgray")
abline(h = log10(0.05), lty = 2, col = "darkgray")
abline(h = 0, lty = 1, col = "black")
legend("topleft", c("docetaxel", "vinorelbine", "paclitaxel"), fill = c("#000000",
    "#8B8378", "#C1CDCD"), border = NA, title = "Drugs")
legend("bottomright", c("a = paclitaxel optimal", "b = paclitaxel/proteasome-inh. consistent",
    "c = paclitaxel/proteasome-inh. inconsistent ", "d = microtubule stabilising"),
    title = "Signatures")
```


Drugs
docetaxel
vinorelbine
paclitaxel
association signed p-value
3
2
1
0
-1
-2
-3
Signatures
a = paclitaxel optimal
b = paclitaxel/proteasome-inh. consistent
c = paclitaxel/proteasome-inh. inconsistent
d = microtubule stabilising
a
b
c
d
b + c
b + d
c + d
b + c + d