

# Task04\_Compare\_Relative\_IC50

*Andrew Stein*

## Contents

Setup	1
load in the data and find common naming in the files	1
Plot distribution of IC50s	1
Load in one dataset and start to explore correlations for	2
Create a function for doing the correlation	3
Perform the cross correlation	3
Correlate the cross-correlations	5

## Setup

Prior toward running this code, you need to first install PharmacoGx (Task01) and download the relevant datasets (Task02)

```
library(PharmacoGx)
library(tidyverse)

## -- Attaching packages ----- tidyverse
## v ggplot2 3.1.0      v purrr  0.3.0
## v tibble  2.0.1      v dplyr  0.7.8
## v tidyr   0.8.2      v stringr 1.3.1
## v readr   1.3.1      v forcats 0.3.0

## -- Conflicts ----- tidyverse_conflicts()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

## load in the data and find common naming in the files

```
load("PSets/CCLE.RData")
load("PSets/GDSC.RData")
common <- intersectPSet(list('CCLE'=CCLE,
                             'GDSC'=GDSC),
                        intersectOn=c("cell.lines", "drugs"),
                        strictIntersect=TRUE)

## Intersecting large PSets may take a long time ...
```

## Plot distribution of IC50s

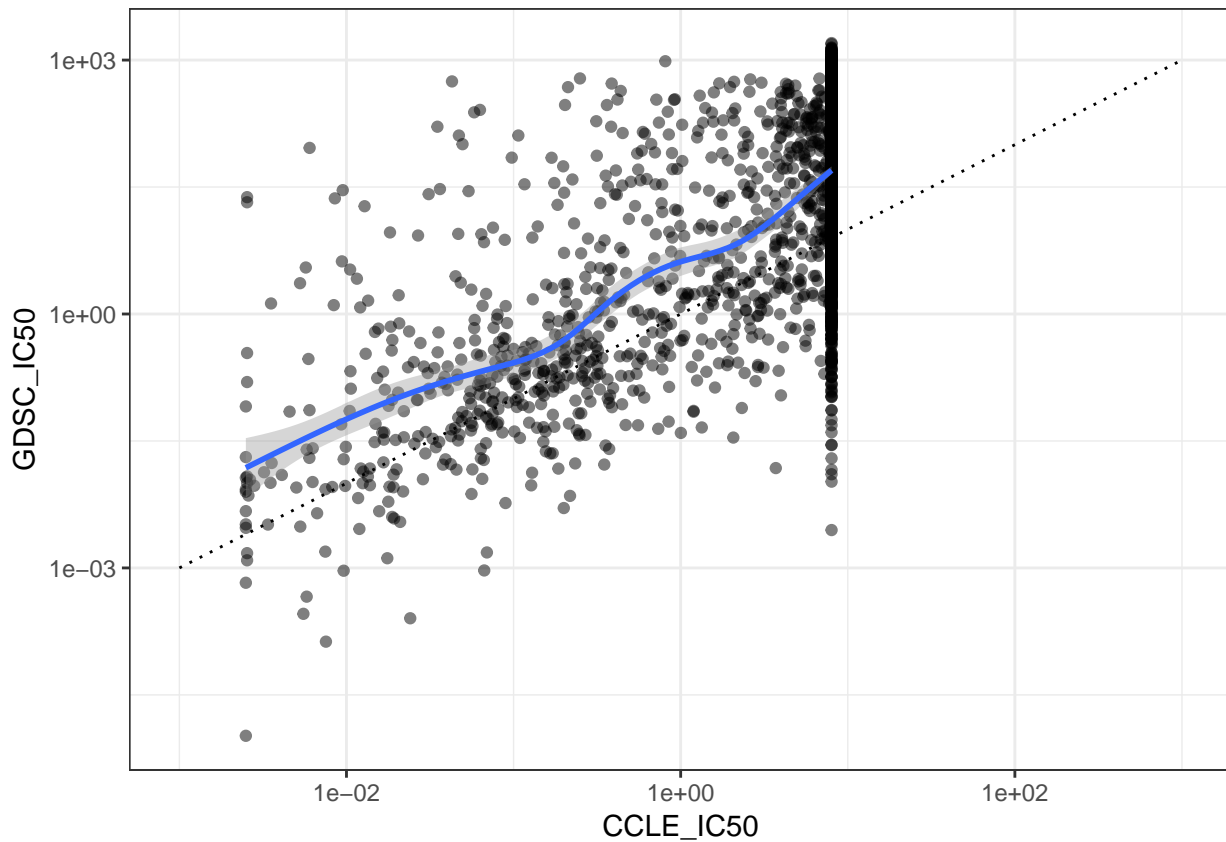
```
measure = "ic50_published"
ic50s = data.frame(
  CCLE_IC50 = as.vector(summarizeSensitivityProfiles(common$CCLE,sensitivity.measure = measure)),
  GDSC_IC50 = as.vector(summarizeSensitivityProfiles(common$GDSC,sensitivity.measure = measure)))
```

```
ggplot(ic50s,aes(x=CCLE_IC50,y=GDSC_IC50)) +
  geom_point(alpha=0.5) +
  geom_smooth() +
  scale_x_log10() +
  scale_y_log10() +
  theme_bw() +
  annotate("segment",x=1e-3,xend=1e3,y=1e-3,yend=1e3,linetype="dotted")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

```
## Warning: Removed 8033 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 8033 rows containing missing values (geom_point).
```



Load in one dataset and start to explore correlations for

Commenting out the plot because now using the function further below

```
pSet = CCLE

info      = sensitivityInfo(pSet)
profiles  = sensitivityProfiles(pSet)
measures  = sensitivityMeasures(pSet)

log_ic50  = summarizeSensitivityProfiles(pSet,sensitivity.measure = "ic50_published") %>%
  log()
```

```
x = log_ic50 %>%
  t() %>%
  cor(use="pairwise")

#corrplot::corrplot(x)

#keep only rows and columns
```

## Create a function for doing the correlation

```
#look at cross correlations
pSet_corr = function(pSet) {

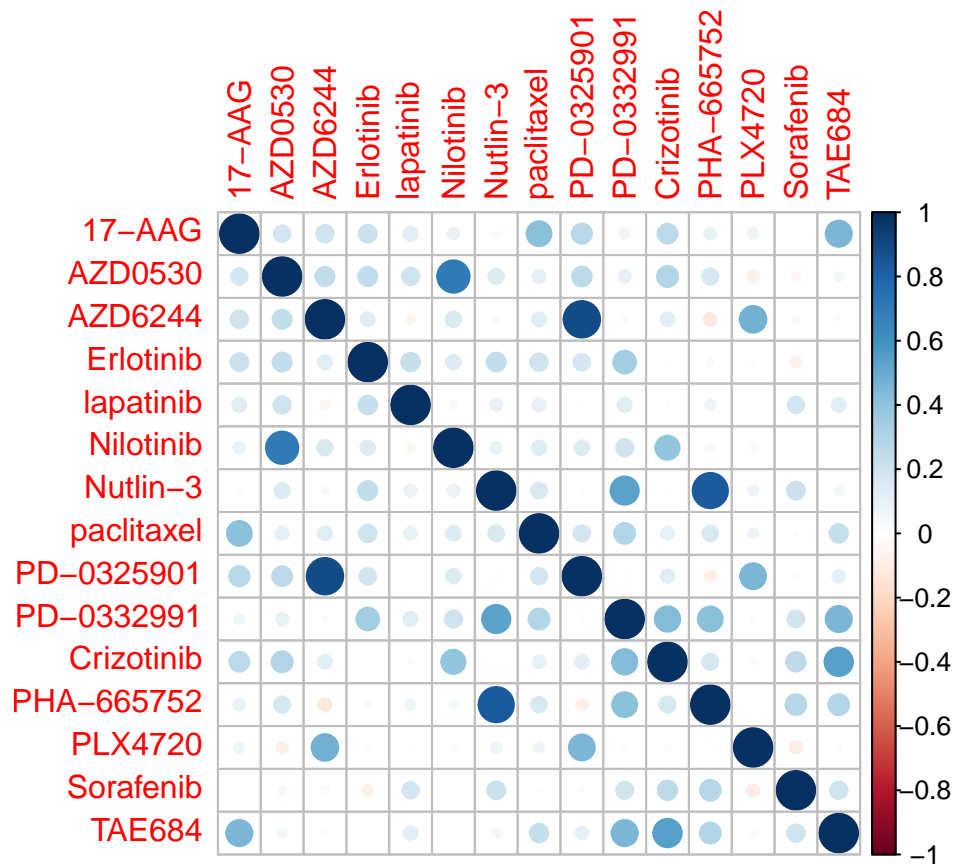
  #get the log_ic50 dataset
  log_ic50 = summarizeSensitivityProfiles(pSet,sensitivity.measure = "ic50_published") %>%
    log()

  x = log_ic50 %>%
    t() %>%
    cor(use="pairwise")

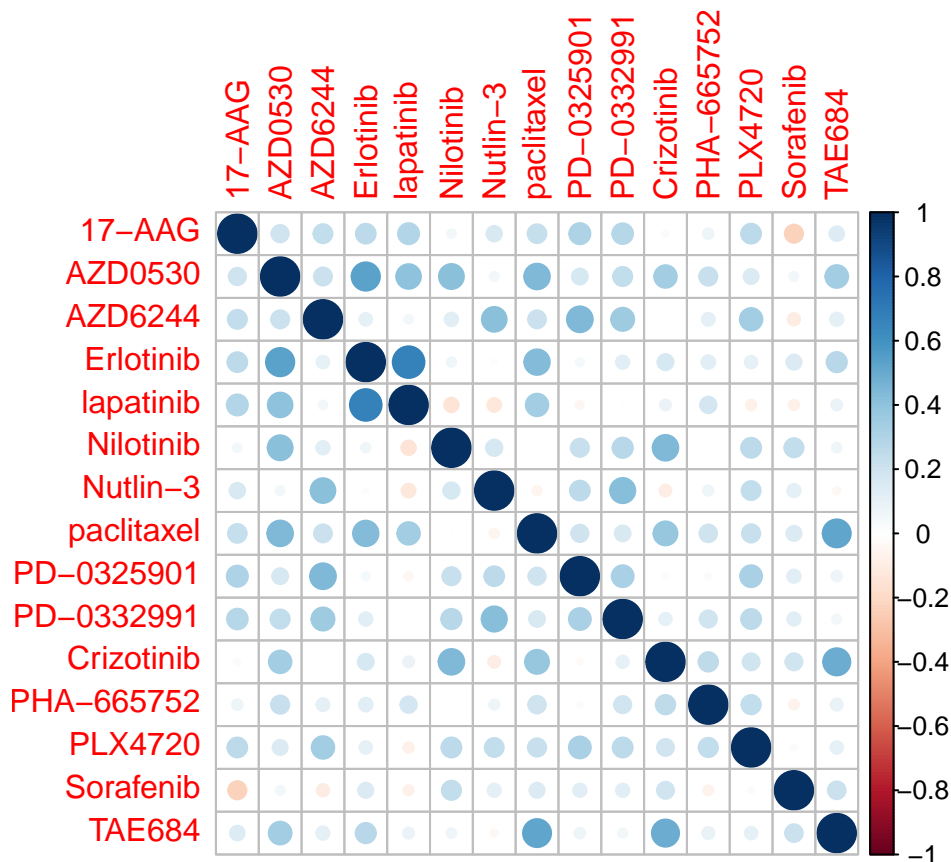
  corrplot::corrplot(x)
  return(x)
}
```

## Perform the cross correlation

```
cor.CCLE = pSet_corr(common$CCLE)
```



```
cor.GDSC = pSet_corr(common$GDSC)
```



## Correlate the cross-correlations

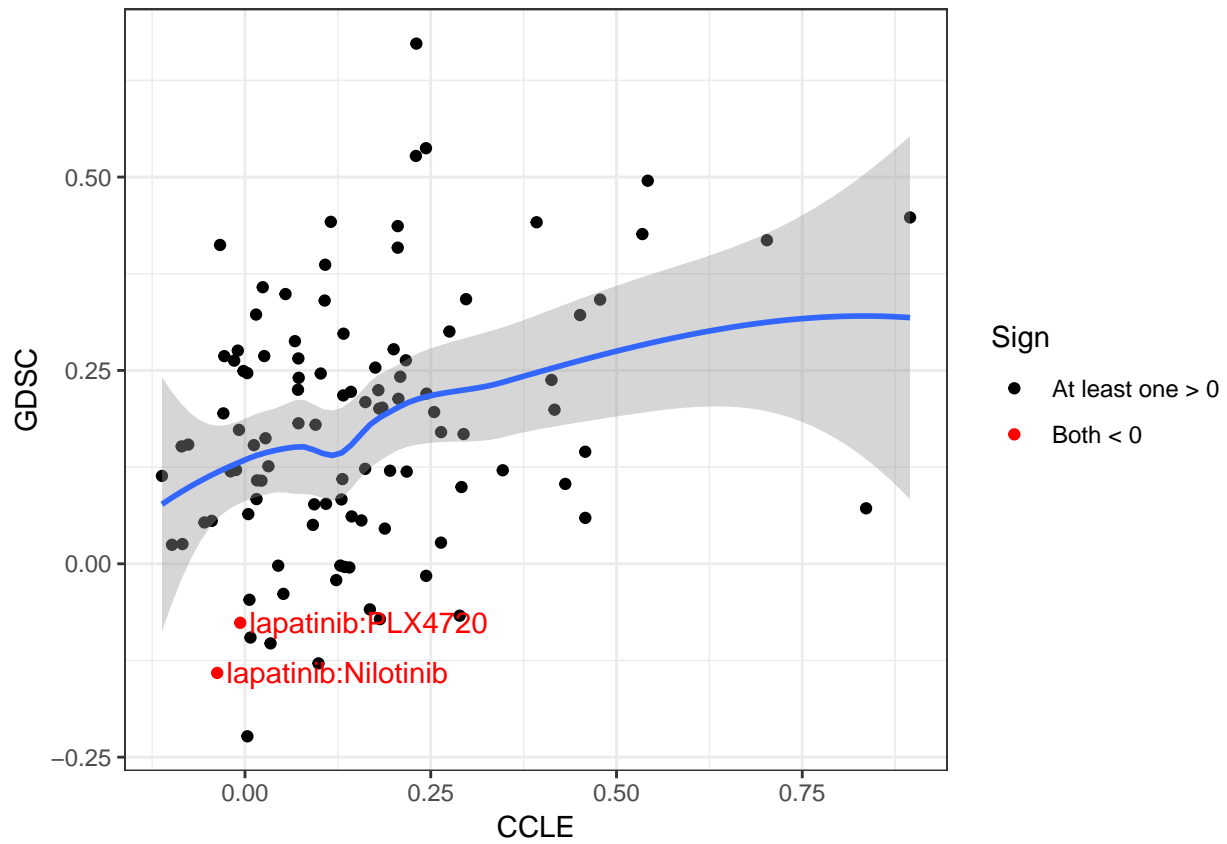
Red highlight the case where both cross-correlations were less than zero

```
n      = nrow(cor.CCLE)
drugs  = row.names(cor.CCLE)
cor.cor = data.frame(CCLE = as.vector(cor.CCLE),
                     GDSC = as.vector(cor.GDSC),
                     drug1= rep(drugs,times=n),
                     drug2= rep(drugs,each=n),
                     num1 = rep(1:n,times=n),
                     num2 = rep(1:n,each=n)) %>%
  mutate(drugs= paste0(drug1,":",drug2),
         Sign = ifelse(CCLE<0 & GDSC<0,
                       "Both < 0",
                       "At least one > 0")) %>%
  filter(num1<num2) #remove the self-correlations

neg.cor = cor.cor %>%
  filter(CCLE<0 & GDSC<0)

ggplot(cor.cor,aes(x=CCLE,y=GDSC,color=Sign)) +
  geom_point() +
  geom_smooth(aes(color=NULL),show.legend = FALSE,method='loess') +
  ggrepel::geom_text_repel(data=neg.cor,aes(label=drugs),show.legend = FALSE) +
```

```
scale_color_manual(values=c("black","red")) +  
theme_bw()
```



## Properties of R session sessionInfo()