# Applications and Statistical Analyses of Proteogenomic Data Using R

Larry Phouthavongsy

April 20 2018

# Contents

# Chapter 1

# Introduction

Pharmacogenomics is an emerging field that combines pharmacology with genomics to elucidate how genetic variations influence responses to drug therapy. Clinical trials provide a rich source of data that need further analysis to guide the drug development process. The potential for the widespread use of pharmacogenomics merits the examination of its fundamental impact on clinical-trial design and practice.[1] With that comes many ethical issues that surround clinical trial design. Researchers hope to use an individuals genome to select the drug along with the correct dosage that is most beneficial for treatment. This area of study is aware that no two patients are alike. Personalized medicine has gained traction in recent years due to the accessibility of genetic data. Drugs are designed for a ”one size fits all” approach but if health care practitioners implement a personalized treatment plan there is a chance that a tailored approach will be of greater benefit (Figure 1.1). In fact, the practice of curating genetic data from clinical trials is currently implemented, Phase II and III trials that are undertaken at present involve taking samples of genomic DNA for pharmacogenomics analysis.[1]

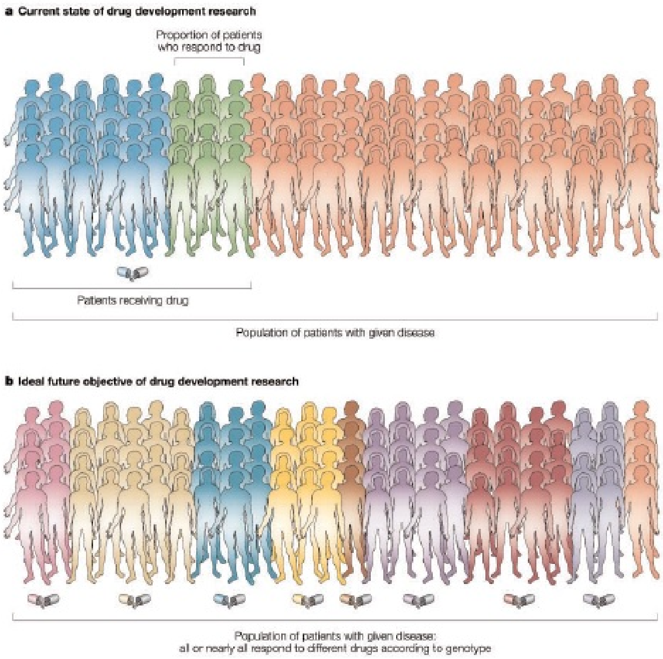
a Current state of drug development research
Proportion of patients who respond to drug
Patients receiving drug
Population of patients with given disease
b Ideal future objective of drug development research
Population of patients with given disease:
all or nearly all respond to different drugs according to genotype

Figure 1.1: (Top) A fraction of patients that receive treatment respond. (Bottom) The goal is to personalize therapies so that all patients respond to treatment.[1]

The main benefit of a personalized drug plan would be to reduce side effects. In turn, this will streamline treatment selection since doctors will understand the genetic makeup of their patients. For example, doctors are now able to screen HIV patients for a genetic variant to lessen side effects of certain antiretroviral drugs.[5] In another example, doctors screened breast cancer patients for a specific gene profile to treat them with the drug trastuzumab which was found to only work with a unique genetic profile.[5] Treating mental illnesses can also benefit from pharmacogenomics since bioaccumulation is often required to show a response. Advances in genome interrogation technology and analytical approaches have evolved the discovery paradigm from candidate gene studies to more agnostic genome-wide analyses of populations of patients who have been characterized for specific drug response phenotypes (e.g., toxicity or desired pharmacologic effects).[2]

# Chapter 2

# Clinical Applications

Drug development through clinical trials will benefit from pharmacogenomic study due to the ability of pharmaceutical companies targeting subgroups with specific genetic profiles. Researchers can use pharmacogenomic data analysis tools to target specific molecular pathways. A pharmacogenomic investigation can also revive old drugs that were once abandoned during the development process. For example, development for the beta-blocker drug bucindolol (Gencaro) was canceled after competing drug companies won FDA approval for their drug.[5] Gencaro has been of interest to researchers lately after it was found to benefit patients with a specific genetic variant.[5] When studying drug action within individuals, researchers often focus on the pharmacokinetic and pharmacodynamic elements of the drug. (Figure 3.1).

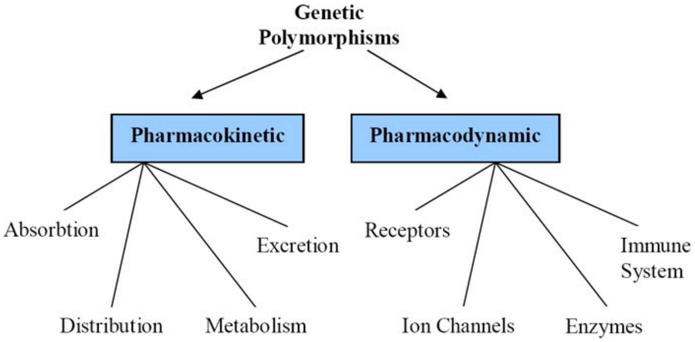

Figure 2.1: Areas of focus in drug action studies. https://www.nature.com/scitable/topicpage/pharmacogenomics-and-personalized-medicine-643

Pharmacokinetics revolves around the four processes of absorption, distribution, metabolism, and excretion. Absorption is how the drug enters the bloodstream, distribution describes in what way the drug travels in the body, metabolism is defined as how the drug is broken down in the body, and excretion is how the drug is eliminated from the body. Pharmacodynamics is the molecular action of a drug on a target. Targets include cellular receptors, ion channels, enzymes, and the immune system. Researchers believe that genetic variance is responsible for pharmacodynamic effects on the patient which can lead to negative side effects also known as adverse effects. These effects are an important consideration when developing drugs. The clinical trial process stresses the importance of reporting adverse drug reactions. Predicting adverse reactions are a priority for pharmacogenomic research due to the fact that it can lessen participant harm and save resources.[6]

# Chapter 3

# Statistical Methods

Similar to other areas of human genomics, pharmacogenomics is experiencing an exponential increase in available data. The vast amount of data brings new opportunities and challenges for the statistical analysis of big data. Statistical pharmacogenomics research by Fan *et al.* investigates the fundamental problems estimating two types of correlation structures: marginal correlation and conditional correlation.[3] Marginal correlation represents the correlation between two variables while ignoring the other variables, and can be estimated using the covariance matrix to introduce methods to estimate false discovery proportions for large-scale simultaneous tests and to select important molecules of SNPs that have significant marginal correlations with biological outcomes.[3] Conditional correlation demonstrates the correlation between two variables by conditioning the other variables to estimate an inverse covariance matrix to find conditional correlations with biomedical responses in the presence of many other molecules or SNPs.[3]

# Notation

Let $\mathrm{A} = (\mathrm{a}_{\mathrm{jk}})\epsilon\mathbb{R}^{\mathrm{dxd}}$, $|\mathrm{B} = \mathrm{b}_{\mathrm{jk}}\epsilon\mathbb{R}^{\mathrm{dxd}}$, and $v = (\mathrm{v}_1,...,\mathrm{v}_{\mathrm{d}})^{\mathrm{T}}\epsilon\mathbb{R}^{\mathrm{d}}$. Denote by $\lambda_{\min}(\mathrm{A})$ and $\lambda_{\max}(\mathrm{A})$ the smallest and largest eigenvalues of A. The inner product of A and B is defined as $\langle \mathrm{A},\mathrm{B}\angle=\mathrm{tr}(\mathrm{A}^{\mathrm{T}}\mathrm{B})$ Define the vector norms: $||\mathrm{v}||_1 = \Sigma_{\mathrm{j}}|\mathrm{v}_{\mathrm{j}},||\mathrm{v}||_2^2=\Sigma_{\mathrm{j}^{\mathrm{v}^2}\mathrm{j}}||\mathrm{v}||_\infty=\max_{\mathrm{j}}|\mathrm{v}_{\mathrm{i}}|$.We also define matrix operator and element-wise norms:$||\mathrm{A}||_2^2\lambda_{\max}(\mathrm{A}^{\mathrm{T}}\mathrm{A}),||\mathrm{A}||_F^2=\Sigma_{\mathrm{j,k}^{\mathrm{a}^2}\mathrm{jk.}}$ Notation A greater than 0 means that A is positive definite and $a_n\bullet b_n$ implies there are positive constants $c_1$ and $c_2$ independent of $n$ such that $c_1 b_n \leq a_n\leq c_2 b_n$.[3]

### 3.0.1 Estimating Large Covariance Matrix

Estimating a large covariance or correlation matrix under a small sample size is a fundamental problem which has many applications in phamacogenomics.[3] Let:

$$x_1, ...x_n \epsilon R^d$$

be n independent observations of a d-dimensional random vector:

$$X = (x_1...x_d)^{\mathrm{T}}$$

Without loss of generality, we assume: $\mathbb{E}\ X = 0$ We want to find a reliable estimate of the population covariance matrix: $\Sigma=\mathbb{E}XX^{\mathrm{T}}$. At the first sight, this problem does not seem to be challenging. In the literature, $\Sigma$ was traditionally estimated by the sample covariance matrix:[3]

$$S = \frac{1}{n}\sum_{i=1}^{n}(x_i\text{-}\overline{x})(x_i^T\text{-}\overline{x})^T \qquad with \qquad \overline{x} = \frac{1}{n}\sum_{i=1}^{n}x_i, \tag{3.1}$$

which has many good theoretical properties when the dimensionality $d$ is small.[3] However, in the more realistic settings where the dimensionality $d$ is comparable or even larger than $n$ (i.e., $d|n$ goes to a nonzero constant or infinity), the sample covariance matrix in (4.1) is no longer a good estimate of the population covariance matrix $\Sigma$.[3] Further statistical analyses will follow to solve and address the inconsistency sample covariance in high dimensions.[3] The above example assumes that $\Sigma$ is sparse and is this not always the correct approach.[3] For example, all genes from the same pathway may be co-regulated by a small number of regulatory factors, which makes gene expression highly correlated and so the researchers proposed the POET method to provide an integrated framework for combining latent factor analysis and sparse covariance matrix estimation.[3] The POET estimator is formed by directly running the singular value decomposition on the sample covariance matrix S.[3] This was just a sample of what needs to be done to analyze pharmacogenomic data. General methodological development for pharmacogenomic data has lagged behind the rapid development of new technologies and new datasets. Further statistical methods need to address the scalability, complexity, noise, and dependence of the datasets.[3]

# Chapter 4

# PharmacoGX

PharmacoGX is an R package for the meta-analysis of large pharmacogenomic data and is available on the Bioconductor website. It was created by researchers from the University of Toronto. The creators demonstrated their package on large drug sensitivity datasets such as the Genomics of Drug Sensitivity in Cancer (GDSC) and the Cancer Cell Line Encyclopedia (CCLE) to perform Connectivity Map analysis.[4] Drug sensitivity datasets are created by screening a large panel of cancer cell lines using multiple drug candidates. Studies by the GDSC and CCLE have revealed several known and novel drug sensitivities and biomarkers.[4] Other pharmacogenomic studies, such as the Connectivity Map project characterized the transcription changes induced by a large set of drugs; these data are referred to as drug perturbation datasets.[4]

PharmacoGX's goal was to investigate large pharmacogenomic datasets to parallelize functions to assess the reproducibility of pharmacological data and to identify the molecular features that are associated with drug effects.[4] A *PharmacoSet* class was developed to contain pharmacological and molecular data with corresponding experimental metadata. This class allows for the productive use of curated annotations for cell lines and drug compounds, which facilitates comparisons between different datasets stored as *PharmacoSet* objects.[4] Cell line names and drug identifiers lack standardization, and this signifies a major barrier for comparative analyses of large pharmacogenomics studies.[4] PharmacoGX addresses this issue by assigning a unique identifier to each cell line and drug.

The function *downloadPset* allows for users to download objects that have been curated by the research team through the *intersectPsets* function. The researchers also included functions to explore pharmacological measurements generated in the drug-sensitivity datasets.[4] Drug dose-response curves can be plotted with the *drugDoseResponseCurve* function. To link molecular features to drug sensitivity the *drugSensitivitySig* function is used to quantify the strength of each gene-drug association using a regression model controlled for treatment duration, tissue type and batch variables.[4] The function *drugPertubationSig* identifies differential gene expressions induced by drug treatments. To identify drugs with carcinogenic or therapeutic potential the *connectivityScore* function can be used.[4]

This package is the first to analyze pharmacogenomic datasets using structured objects. PharmacoGX provides a foundation to develop drug-related molecular signatures.[4] The GDSC and CCLE datasets promote further analysis for the creation of biomarkers of drug response. Further curation of more datasets in the future will provide a wealth of information for pharmacogenomic analyses. The generation of strong biomarkers for a unique drug response would constitute a major step toward the realization of precision medicine.[4]

### 4.0.1 Creating a PharmacoGx Object

```
PharmacoSet(name,
            molecularProfiles=list(),
            cell=data.frame(),
            drug=data.frame(),
            sensitivityInfo=data.frame(),
            sensitivityRaw=array(dim=c(0,0,0)),
            sensitivityProfiles=matrix(),
            curationDrug=data.frame(),
            curationCell=data.frame(),
            curationTissue=data.frame(),
            datasetType=c(”sensitivity”, ”perturbation”, ”both”),
            verify = TRUE)
```

This requires specific annotations for the data to be able to function correctly with the function provided in PharmacoGx. The annotations allow the user to interrogate the data on the basis of cells and drugs. The only data that is necessary when creating an PharmacoGx object is the name of the PharmacoSet which is passed to the constructor. The datasetType slot is a character string to determine if the PharmacoSet contains drug dose sensitivity data, genomic pertubation data or both. The cell slot contains a data.frame which holds information about the cell profiles across the data types in the PharmacoSet and includes pertubation and sensitivity experiments. The drug slot contains a data.frame as well, and contains information about all the drugs contained across all the data types in the PharmacoSet which includes pertubation and sensitivity experiments. Each entry in the data frame must be a unique compound identifier used across all cell data types for each compound.

The molecularProfiles slot of a PharmacoSet object contains the molecular expression data profiled in the dataset. Each data type is kept in a separate ExpressionSet object and they are all stored as a list. The ExpressionSet object must be labelled with the type of data that forms the object. The phenoData in each expression set object requires specific columns labelling each experiment. The cellid column contains a cell identifier that has to match the rownames of the cell slot of the PSet exactly. The drugid is the identifier of the drug used in each experiment. The drugid must match the rownames of the drug slot of the PharmacoSet object exactly.

The sensitivity list contains all the data related to drug does response experiments. The info data.frame contains the metadata for each pharmacological experiment. Each row of the info data.frame requires a unique experiment identifier. It must contain the following name slots in the list: info, cellid, raw, profiles and n. Raw is a 3-D array of raw drug dose response data. The first dimension is labeled as the experiments and the names of each row in this dimension matching the experiment ids used in the metadata data.frame. The second dimension is the doses. The third dimension is always fixed at 2 containing dose, and viability. Dose refers to the actual dose administered, and viability contains the viability measurement at that dose. Profiles is a data.frame containing matching rownames as labeled in the info data.frame and each column being a summary of the drug dose sensitivity. N is a matrix that contains cellids referring to the cell slot of the Pset object with drugids matching the rownames of the drug slot.

The pertubation slot is filled by the constructor. It contains n and info as seen in the sensitivity slot. This array summarizes how many pertubation experiments are in each molecular data type for each pair, and this allows for quick referencing when the pairs are used at the same time. The curation slot is made up of three data.frames: drugs, tissues and cells.

Once the data is prepared, it can be passed to a constructor function. Any data that is missing from the constructor will be created by the constructor.

### 4.0.2 Exploring drug sensitivity data from cancer pharmacogenomic studies

In an article by Pozdeyev *et.al* a new drug sensitivity test was developed to calculate the area under the dose response curve adjusted for the range of various drug concentrations. This allowed for the integration of heterogeneous drug sensitivity data from the CCLE, GDSC and the Cancer Therapeutics Response Portal (CTRP). From their research, they showed that there is a positive correlation of drug sensitivity data from many targeted therapies. The study determined the area under the dose-response curve (AUC) adjusted for the range of tested concentrations to unbiased comparison of drug screening data from the CCLE, GDSC, and CTRP databases.

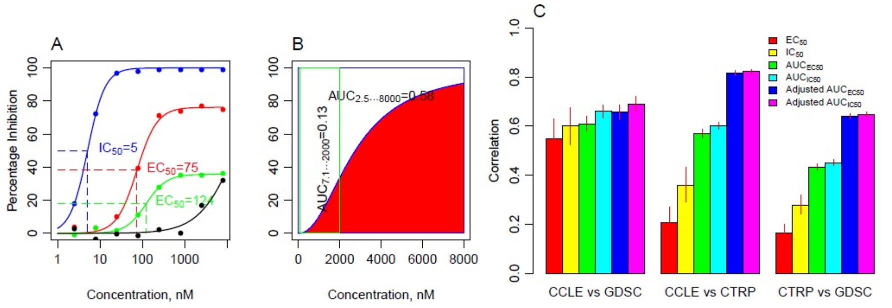

Figure 4.1: A. IC$_{50}$ defined as a drug concentration producing absolute 50 percent inhibition of growth in the proliferation assay denoted by the blue line. B. Comparison of AUC calculated from the same dose-response data for the range of concentrations used by GDSC and CCLE. C. Pearson correlation coefficients (r) for the comparison of drug sensitivity data in 3 databases using six drug sensitivity metrics.

```
# scipt generates Figure 4.1 that demonstrates pluses and
    minuses of various drug sensitvity metrics and
   compares databases.

library("drc")

rm(list=ls())
#Set working directory containing R scripts here
setwd("F:/Drug_Sensitvity_Metrics/R")

#Defining AUC function
```

```
AUC_calc <- function(IC50, slope, lower, upper, minC,
    maxC) {

  sequence <- seq(from=floor(minC), to=ceiling(maxC), by
      = 1)

  y_curve <- upper + (lower-upper)/(1+(sequence/IC50)^
      slope)
  y_total <- rep((max(100,upper)-lower), length(sequence)
      )

  return(sum(y_curve-lower)/sum(y_total))
}

#database
db = "CCLE"
#loading raw data in standardized format
load(file=file.path(paste0("../data/", db, "_raw_data_no_
    duplicates_consensus.RData")))



fpout <- paste0("../fig/Figure_1.pdf")

pdf(fpout, width=8, height=3)
par(pty = "s")
layout(matrix(c(1,1,1,2,2,2,3,3,3,3,3), nrow = 1, ncol =
    11))
```

```
par(mar=c(2, 3, 1, 1), oma = c(2,2,2,2))
par(mgp = c(3,0.7,0))


plot(NA, NA,
     ylim = c(0, 100),
     xlim = c(1, 8000),
     main = "",
     xlab = "Concentration,␣nM",
     log = "x")
title(ylab = "Percentage␣Inhibition", line = 2)
mtext("A", side = 3, adj = 0, cex = 0.9, line = 0.5)


#IC50 curve
drug_data <- ccle[(ccle[ , "Drug_name"]=="Paclitaxel") &
    (ccle[ , "Cell_line"] == "REH"), ]
response <- as.numeric(unlist(strsplit(as.character(drug_
    data[1, "Responses"]), ",", fixed = T)))
DrugC <- as.numeric(unlist(strsplit(as.character(drug_
    data[1, "Concentrations_nM"]), ",", fixed = T)))
curve <- data.frame(conc = DrugC[!is.na(response)], resp
    = response[!is.na(response)])

try(fit <- drm(formula = resp ~ conc, data = curve,
               fct = LL2.4(names=c("Slope","Lower␣Limit",
                  "Upper␣Limit", "IC50")),
                 lowerl = c(-Inf, 0, 99.99999, 0),
                 upperl = c(-0.1, 0.000001, 100, Inf)))

lower <-  coef(fit)[2]
upper <- coef(fit)[3]
slope <-  coef(fit)[1]*(-1)
IC50 <-  exp(coef(fit)[4])

curve(upper + (lower-upper)/(1+(x/IC50)^slope), 1, 8000,
    n = 1000, add = TRUE, col = ”blue”, lwd = 1)
points(curve$conc, curve$resp, col = ”blue”, pch = 20)
segments(1, 50, IC50, 50, col = ”blue”, lty = ”longdash”,
     lwd = 1)
segments(IC50, 50, IC50,0, col = ”blue”, lty = ”longdash”
    , lwd = 1)
text(IC50+1, 50, bquote(paste(IC[50], ”=”, .(round(IC50
    ,0)))), pos = 4, col = ”blue”)



#EC50 curve
drug_data <- ccle[(ccle[ , ”Drug_name”]==”Selumetinib”) &
     (ccle[ , ”Cell_line”] == ”A375”), ]
response <- as.numeric(unlist(strsplit(as.character(drug_
    data[1, ”Responses”]), ”,”, fixed = T)))
DrugC <- as.numeric(unlist(strsplit(as.character(drug_
    data[1, ”Concentrations_nM”]), ”,”, fixed = T)))
```

```
curve <- data.frame(conc = DrugC[!is.na(response)], resp
    = response[!is.na(response)])

try(fit <- drm(formula = resp ~ conc, data = curve,
                  fct = LL2.4(names=c(”Slope”,”Lower␣Limit”,
                      ”Upper␣Limit”, ”EC50”)),
                  lowerl = c(-Inf, min (0, min(curve$resp)),
                       min (0, min(curve$resp)), 0),
                  upperl = c(-0.1, 0.0001, max(100,max(curve
                      $resp)), Inf)))

lower <-  coef(fit)[2]
upper <- coef(fit)[3]
slope <-  coef(fit)[1]*(-1)
IC50 <-  exp(coef(fit)[4])
mid <- (upper - lower)/2

curve(upper + (lower-upper)/(1+(x/IC50)^slope), 1, 8000,
    n = 1000, add = TRUE, col = ”red”, lwd = 1)
points(curve$conc, curve$resp, col = ”red”, pch = 20)
segments(1, mid, IC50, mid, col = ”red”, lty = ”longdash”
    , lwd = 1)
segments(IC50, mid, IC50,0, col = ”red”, lty = ”longdash”
    , lwd = 1)
text(IC50+10, mid, bquote(paste(EC[50], ”=”, .(round(IC50
    ,0)))), pos = 4, col = ”red”)
```

```
#low amplitude EC50 curve
drug_data <- ccle[(ccle[ , "Drug_name"]=="Selumetinib") &
      (ccle[ , "Cell_line"] == "639V"), ]
response <- as.numeric(unlist(strsplit(as.character(drug_
    data[1, "Responses"]), ",", fixed = T)))
DrugC <- as.numeric(unlist(strsplit(as.character(drug_
    data[1, "Concentrations_nM"]), ",", fixed = T)))
curve <- data.frame(conc = DrugC[!is.na(response)], resp
    = response[!is.na(response)])

try(fit <- drm(formula = resp ~ conc, data = curve,
                  fct = LL2.4(names=c("Slope","Lower Limit",
                      "Upper Limit", "EC50")),
                  lowerl = c(-Inf, min (0, min(curve$resp)),
                       min (0, min(curve$resp)), 0),
                  upperl = c(-0.1, 0.0001, max(100,max(curve
                      $resp)), Inf)))

lower <-  coef(fit)[2]
upper <- coef(fit)[3]
slope <-  coef(fit)[1]*(-1)
IC50 <-  exp(coef(fit)[4])
mid <- (upper - lower)/2

curve(upper + (lower-upper)/(1+(x/IC50)^slope), 1, 8000,
    n = 1000, add = TRUE, col = "green", lwd = 1)
```

```
points(curve$conc, curve$resp, col = "green", pch = 20)
segments(1, mid, IC50, mid, col = "green", lty = "
    longdash", lwd = 1)
segments(IC50, mid, IC50,0, col = "green", lty = "
    longdash", lwd = 1)
text(IC50+10, mid, bquote(paste(EC[50], "=", .(round(IC50
    ,0)))), pos = 4, col = "green")



#incomplete curve
#IC50 curve
drug_data <- ccle[(ccle[ , "Drug_name"]=="Selumetinib") &
     (ccle[ , "Cell_line"] == "EFO27"), ]
response <- as.numeric(unlist(strsplit(as.character(drug_
    data[1, "Responses"]), ",", fixed = T)))
DrugC <- as.numeric(unlist(strsplit(as.character(drug_
    data[1, "Concentrations_nM"]), ",", fixed = T)))
curve <- data.frame(conc = DrugC[!is.na(response)], resp
    = response[!is.na(response)])

try(fit <- drm(formula = resp ~ conc, data = curve,
                  fct = LL2.4(names=c("Slope","Lower_Limit",
                      "Upper_Limit", "IC50")),
                  lowerl = c(-Inf, 0, 99.99999, 0),
                  upperl = c(-0.1, 0.000001, 100, Inf)))

lower <-  coef(fit)[2]
```

```
upper <- coef(fit)[3]
slope <-  coef(fit)[1]*(-1)
IC50 <-  exp(coef(fit)[4])

curve(upper + (lower-upper)/(1+(x/IC50)^slope), 1, 8000,
    n = 1000, add = TRUE, col = ”black”, lwd = 1)
points(curve$conc, curve$resp, col = ”black”, pch = 20)



#AUC demonstration
plot(NA, NA,
      ylim = c(0, 100),
      xlim = c(1, 8000),
      main = ””,
      ylab = ””, xlab = ”Concentration,␣nM”)

mtext(”B”, side = 3, adj = 0, cex = 0.9, line = 0.5)

drug_data <- ccle[(ccle[ , ”Drug_name”]==”Crizotinib”) &
    (ccle[ , ”Cell_line”] == ”KMS26”), ]
response <- as.numeric(unlist(strsplit(as.character(drug_
    data[1, ”Responses”]), ”,”, fixed = T)))
DrugC <- as.numeric(unlist(strsplit(as.character(drug_
    data[1, ”Concentrations_nM”]), ”,”, fixed = T)))
curve <- data.frame(conc = DrugC[!is.na(response)], resp
    = response[!is.na(response)])
```

```
try(fit <- drm(formula = resp ~ conc, data = curve,
               fct = LL2.4(names=c(”Slope”,”Lower_Limit”,
                 ”Upper_Limit”, ”EC50”)),
               lowerl = c(-Inf, min (0, min(curve$resp)),
                  min (0, min(curve$resp)), 0),
               upperl = c(-0.1, 0.0001, max(100,max(curve
                 $resp)), Inf)))

lower <-  coef(fit)[2]
upper <- coef(fit)[3]
slope <-  coef(fit)[1]*(-1)
IC50 <-  exp(coef(fit)[4])
mid <- (upper - lower)/2

s <- seq(1,8000, 0.1)
y <- upper + (lower-upper)/(1+(s/IC50)^slope)
segments(s, 0, s, y, col = ”red”, lty = ”solid”, lwd = 1)

curve(upper + (lower-upper)/(1+(x/IC50)^slope), 1, 8000,
  n = 1000, add = TRUE, col = ”blue”, lwd = 1)

segments(c(2.5,2.5,8000,8000), c(0, 100, 100 ,0), c
  (2.5,8000,8000,2.5), c(100,100, 0, 0), col = ”blue”,
  lwd = 1)
text(4000, 90, bquote(paste(’AUC’[2.5][...][8000], ”=”,
  .(round(AUC_calc(IC50, slope, lower, upper, 2.5, 8000)
  ,2)))), pos = 1)
```

```
segments(c(75,75,2000,2000), c(0, 100, 100 ,0), c
    (75,2000,2000,75), c(100,100, 0, 0), col = ”green”,
    lwd = 1)
text(1000, 50, bquote(paste('AUC'[7.1][...][2000], ”=”,
    .(round(AUC_calc(IC50, slope, lower, upper, 7.8125,
    2000),2)))), pos = 1, srt = 90)



# Comparing drug sensitivity metrics
db1 = ”CCLE”
db2 = ”GDSC”
db3 = ”CTRP”

#loading fitted data and lists of matched drugs and cell
    lines
load(file=file.path(paste0(”../data/”, db1, ”␣vs␣”, db2,
    ”␣correlations.RData”)))
correl <- correlations
load(file=file.path(paste0(”../data/”, db1, ”␣vs␣”, db3,
    ”␣correlations.RData”)))
correl <- rbind(correl, correlations)
load(file=file.path(paste0(”../data/”, db3, ”␣vs␣”, db2,
    ”␣correlations.RData”)))
correl <- rbind(correl, correlations)
```

```
# plotting 6 drug sensitivity metrics for 3 database
    comparisons

bar1 <- correl[(correl$database1 == db1) & (correl$
    database2 == db2) & (correl$drug == ”All”), ]
bar1 <- bar1[c(1,3,5,7,9,11), ]
bar2 <- correl[(correl$database1 == db1) & (correl$
    database2 == db3) & (correl$drug == ”All”), ]
bar2 <- bar2[c(1,3,5,7,9,11), ]
bar3 <- correl[(correl$database1 == db3) & (correl$
    database2 == db2) & (correl$drug == ”All”), ]
bar3 <- bar3[c(1,3,5,7,9,11), ]

bars <- cbind(as.numeric(as.character(bar1$correlation)),
     as.numeric(as.character(bar2$correlation)),
                as.numeric(as.character(bar3$correlation)))

ci1 <- as.character(correl[(correl$database1 == db1) & (
    correl$database2 == db2) & (correl$drug == ”All”), ”
    confidence.interval”])
ci1 <- ci1[c(1,3,5,7,9,11)]
ci1_down <- as.numeric(unlist(strsplit(as.character(ci1),
     ”,”, fixed = T)))[c(1,3,5,7,9,11)]
ci1_up <- as.numeric(unlist(strsplit(as.character(ci1), ”
    ,”, fixed = T)))[c(2,4,6,8,10,12)]
```

```
ci2 <- as.character(correl[(correl$database1 == db1) & (
    correl$database2 == db3) & (correl$drug == "All"), "
    confidence.interval"])
ci2 <- ci2[c(1,3,5,7,9,11)]
ci2_down <- as.numeric(unlist(strsplit(as.character(ci2),
     ",", fixed = T)))[c(1,3,5,7,9,11)]
ci2_up <- as.numeric(unlist(strsplit(as.character(ci2), "
    ,", fixed = T)))[c(2,4,6,8,10,12)]

ci3 <- as.character(correl[(correl$database1 == db3) & (
    correl$database2 == db2) & (correl$drug == "All"), "
    confidence.interval"])
ci3 <- ci3[c(1,3,5,7,9,11)]
ci3_down <- as.numeric(unlist(strsplit(as.character(ci3),
     ",", fixed = T)))[c(1,3,5,7,9,11)]
ci3_up <- as.numeric(unlist(strsplit(as.character(ci3), "
    ,", fixed = T)))[c(2,4,6,8,10,12)]

ci_down <- cbind(ci1_down, ci2_down, ci3_down)
ci_up <- cbind(ci1_up, ci2_up, ci3_up)

par(pty = "m")
par(mar=c(2, 3, 1, 0))
# par(pty = "s", mfrow = c(2,3), cex.main = 1)
# par(mar=c(1.5, 4, 4.5, 1), oma = c(2,2,2,2))
# plot.window(c(0,3), c(0,2))
# par(cex = 0.7)
```

```
colnames(bars) <- c("CCLE␣vs␣GDSC", "CCLE␣vs␣CTRP", "CTRP␣vs␣GDSC")
# cols <- c("blue", "red", "cyan", "orange", "deepskyblue", "deeppink")
bp <- barplot(as.matrix(bars), beside = T, col = rainbow(6), ylim = c(0,1))
title(ylab = "Correlation", line = 2)
legend_texts <- expression(EC[50], IC[50], AUC[EC50], AUC[IC50], Adjusted~AUC[EC50], Adjusted~AUC[IC50])
legend(x=14, y=1, legend_texts, bty = "n", fill = rainbow(6), cex= 0.8)
mtext("C", side = 3, adj = 0, cex = 0.9, line = 0.5)

arrows(bp, ci_down, bp, ci_up, length=0, col = "red", lwd = 1)



dev.off()
```

Figure 4.1a. displays a graph that shows the inhibition of growth in the proliferation assay and it relies on the assumption that at a high concentration of the drug 100 percent of the effect is achieved.[7] Pharmacogenomics focuses on targeted therapies and drugs such as MEK inhibitors which are cytotastic. Dose response curves produced for cytostatic drugs plateau at a percent inhibition (maximal drug effect, $A_{max}$) of less than 100 percent (Figure 4.1a, red). 4-point logistic regression can estimate lower and upper asymptotes of the sigmoid dose-response curve that differ from 0 and 100 percent of an $IC_{50}$ curve and calculate half maximal effective concentration ($EC_{50}$, also called relative $IC_{50}$), which is defined as a concentration of a drug causing an effect equal to the 50 percent of the Amax.[7] $EC_{50}$ (Fig 4.1a. green) represents a true low amplitude drug response to a cytotastic drug. Incomplete dose response curves (Fig 4.1a. black) show a challenge when interpreting drug sensitivity data.

The AUC (Fig 4.1b) is a very good drug sensitivity metric because it can be calculated for any dose response curve. The AUC combines the information about the potency ($EC_{50}$ and $IC_{50}$) and efficacy ($A_{max}$) of the drug into a single measure. This has been proven to be a strong metric for comparing a single drug across cell lines, and a better measure of cell line selectivity when compared to IC50.[7] The only issue is that the AUC requires that the range of drug concentrations varies between studies. To solve this problem the researchers adjusted the AUC where this new metric takes into account the differences in the range of tested drug concentrations. The adjusted AUC uses sigmoid curve parameters estimated with a standard logistic regression ($IC_{50}$ or $EC_{50}$ models for adjusted $AUC_{IC50}$ and adjusted $AUC_{EC50}$, respectively). However, this was only calculated only for the range of concentrations that were shared by the dose-response curves being compared (Fig 4.1b. green).[7] To evaluate the performance of the adjusted AUCs and compare it to the traditional drug sensitivity metrics ($IC_{50}$ and $EC_{50}$, and unadjusted AUCs), they correlated drug sensitivity data from the CCLE, GDSC, and CTRP databases (Fig 4.1c.).

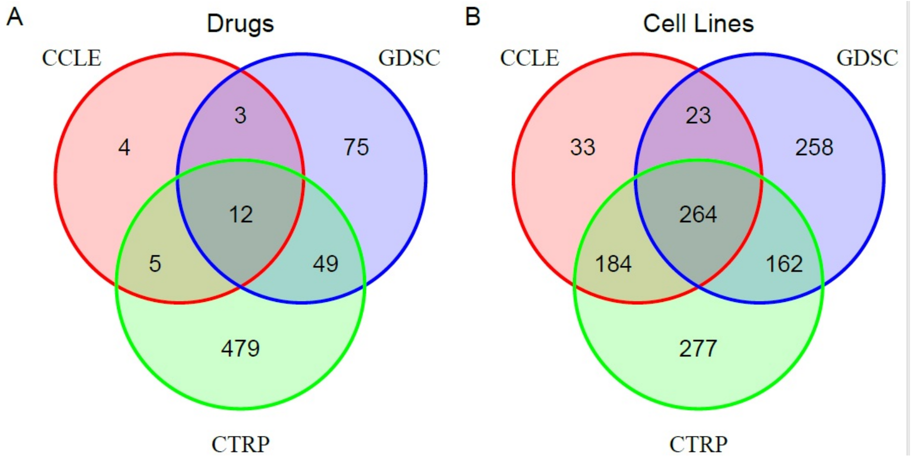

Figure 4.2: There is significant overlap in the drugs and cell lines analyzed by the databases.

The researchers compared data for all the compounds with the goal of identifying the sensitivity metric that shows the best relationship between databases and in theory can be the most reproducible quantitative assessment of the drug sensitivity for the combined pharmacogenomic analysis.[7] (Fig 4.1c) $IC_{50}$, $EC_{50}$ and unadjusted area under the curve drug sensitivity metrics produces low correlation between the databases.

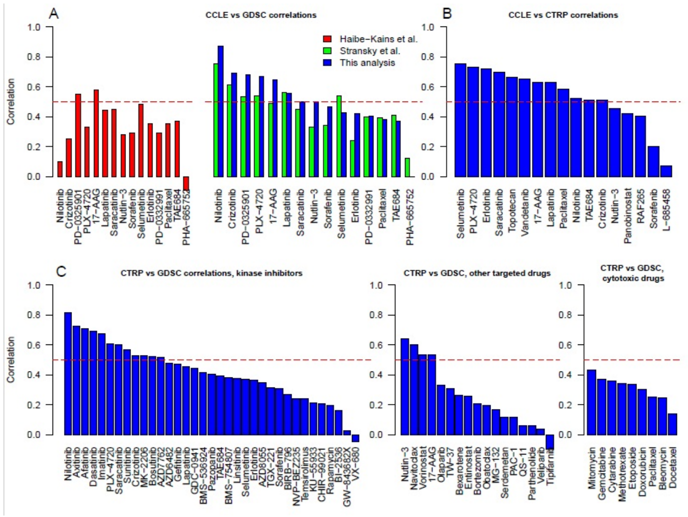

Figure 4.3: Correlations were found when data is stratified by an individual compound

```
#Fig 4.3
rm(list=ls())
#Set working directory containing R scripts here
setwd(”F:/Drug_Sensitvity_Metrics/R”)

# load(file=file.path(paste0(”../data/drugs.RData”)))

# Fig 2A
db1 = ”CCLE”
db2 = ”GDSC”

#loading fitted data and lists of matched drugs and cell
    lines
load(file=file.path(paste0(”../data/”, db1, ”␣vs␣”, db2,
    "␣correlations.RData")))
correl <- correlations

drugs <- as.character(unique(correl$drug))
drugs <- drugs[drugs != "All"]

this_analysis <- correl[(correl$drug %in% drugs) & (
    correl$correlation_type == "pearson") & (correl$drug_
    sensitivity_metric == "AUC_IC50_adj"), ]
# this_analysis <- this_analysis[order(as.character(this_
    analysis$drug)), ]

#data from Haibe-Kains et al. and Stransky et al.
hk <- c(0.58, 0.1, 0.28, 0.55, 0.29, 0.33, 0.48, 0.25,
    0.35, 0.44, 0.35, -0.09, 0.45, 0.29, 0.37)
stransky <- c(0.49, 0.75, 0.33, 0.53, 0.4, 0.54, 0.54,
    0.61, 0.24, 0.56, 0.39, 0.12, 0.45, 0.34, 0.41)

plotdata <- rbind(hk,  stransky,  this = as.numeric(as.
    character(this_analysis$correlation)))
colnames(plotdata) <- drugs
plotdata <- plotdata[, order(plotdata["this",],
    decreasing = T)]

pd1 <- plotdata[1, ,drop = F]
pd2 = plotdata[-1, ]
```

```
#testing for significant difference between databases
cors <- rbind(cbind(hk, rep(1, length(hk))),
                cbind(stransky, rep(2, length(stransky))),
                cbind(as.numeric(as.character(this_analysis
                    $correlation)), rep(3, length(this_
                    analysis$correlation))))
colnames(cors) <- c("correlations", "analysis")
kruskal.test(correlations ~ analysis, data = cors)
wilcox.test(stransky, as.numeric(as.character(this_
    analysis$correlation)), paired = T)

fpout <- "../fig/Figure␣2.pdf"

pdf(fpout, width=8, height=6)
layout(rbind(c
    (1,1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,2,3,3,3,3,3,3,3,3,3,3,3,3,
    ,
               c
                   (4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5,5
                   ))
par(cex = 0.7, las = 2, cex.main = 0.8)
par(mar=c(7, 4, 2, 0))
barplot(pd1, beside = T, col = "red", ylim = c(0,1), ylab
     = "Correlation",
         main = "")
abline(h = 0.5, col = "red", lty = "longdash")
```

```
mtext("A", side = 3, at = −1, adj = 0, cex= 1.1, las = 1,
    line = 0.5)

par(cex = 0.7, las = 2, cex.main = 0.8)
par(mar=c(7, 1, 2, 1))
barplot(pd2, beside = T, col = c("green", "blue"), ylim =
    c(0,1), ylab = "Correlation",
        main = "CCLE␣vs␣GDSC␣correlations", adj = 0, axes
            = F)
abline(h = 0.5, col = c("red"), lty = "longdash")
legend_texts <− c("Haibe−Kains␣et␣al.", "Stransky␣et␣al."
    , "This␣analysis")
legend("topright", legend_texts, bty = "n", fill =
    rainbow(3))
# mtext("CCLE vs GDSC correlations", side = 3, at = −1,
    adj = 0, cex= 0.7, las = 1, line = 0.5)

#Fig2B
#correlations between CCLE and CTRP
rm(list=ls())
setwd("E:/Pharmacogenomics/R")

db1 = "CCLE"
db2 = "CTRP"

#loading fitted data and lists of matched drugs and cell
    lines
```

```
load(file=file.path(paste0("../data/", db1, "␣vs␣", db2,
   "␣correlations.RData")))
correl <- correlations
# drugnames <- read.csv("../raw_data/Drug_names_matched.
   csv", header = T, stringsAsFactors = F)

drugs <- as.character(unique(correl$drug))
drugs <- drugs[drugs != "All"]

this_analysis <- as.numeric(as.character(correl[(correl$
   drug %in% drugs) & (correl$correlation_type == "
   pearson") &
                        (correl$drug_sensitivity_metric
                           == "AUC_IC50_adj"), "
                          correlation"]))
names(this_analysis) <- drugs
plotdata <- sort(this_analysis, decreasing = T)


barplot(plotdata, beside = T, col = "blue", ylim = c(0,1)
   , ylab = "Correlation",
        main = "CCLE␣vs␣CTRP␣correlations")
abline(h = 0.5, col = "red", lty = "longdash")
mtext("B", side = 3, at = -1, adj = 0, cex= 1.1, las = 1,
    line = 0.5)

#Fig2C1-3
```

```
#correlations between CTRP and GDSC

rm(list=ls())
setwd("F:/Pharmacogenomics/R")

#dividing intersect drugs into groups: kinase inhibitors,
      cytotoxic chemotherapies, other targeted therapies

ki <- c("Afatinib", "Axitinib", "AZD6482", "AZD7762", "
    AZD8055", "BI-2536", "BIRB-796", "BMS-536924", "BMS
    -754807", "Bosutinib",
          "CHIR-99021", "Crizotinib", "Dasatinib", "
              Erlotinib", "GDC-0941", "Gefitinib", "GW
              -843682X", "Imatinib", "KU-55933",
          "Lapatinib", "Linsitinib", "MK-2206", "Nilotinib"
              , "NVP-BEZ235", "TAE684", "Pazopanib", "PLX
              -4720", "Saracatinib",
          "Selumetinib", "Rapamycin", "Sorafenib", "
              Sunitinib", "Temsirolimus", "TGX-221", "VX-680
              ")

cytotoxic <- c("Bleomycin", "Cytarabine", "Docetaxel", "
    Doxorubicin", "Etoposide", "Gemcitabine", "
    Methotrexate",
                "Mitomycin", "Paclitaxel")

other <- c("Bexarotene", "Bortezomib", "Entinostat", "MG
    -132", "Navitoclax", "Nutlin-3", "Obatoclax", "
    Olaparib", "PAC-1",
             "Parthenolide", "QS-11", "Serdemetan", "17-AAG
                ", "Tipifarnib", "TW-37", "Veliparib", "
                Vorinostat")

db1 = "CTRP"
db2 = "GDSC"

#loading fitted data and lists of matched drugs and cell
    lines
load(file=file.path(paste0("../data/", db1, "␣vs␣", db2,
    "␣correlations.RData")))
correl <- correlations

par(mar=c(7, 4, 2, 1))

this_analysis <- correl[(correl$drug %in% ki) & (correl$
    correlation_type == "pearson") &
                              (correl$drug_sensitivity_metric
                                  == "AUC_IC50_adj"), ]
drug_names <- as.character(this_analysis[, "drug"])
plotdata <- as.numeric(as.character(this_analysis[, "
    correlation"]))
names(plotdata) <- drug_names
plotdata <- sort(plotdata, decreasing = T)
```

```
barplot(plotdata, beside = T, col = "blue", ylim = c(0,1)
    ,
        main = "CTRP vs GDSC correlations, kinase 
            inhibitors", ylab = "Correlation")
abline(h = 0.5, col = "red", lty = "longdash")
mtext("C", side = 3, at = -1, adj = 0, cex= 1.1, las = 1,
     line = 0.5)

par(mar=c(7, 1, 2, 1))
#Other targeted therapies
this_analysis <- correl[(correl$drug %in% other) & (
    correl$correlation_type == "pearson") &
                            (correl$drug_sensitivity_metric
                                == "AUC_IC50_adj"), ]

drug_names <- as.character(this_analysis[, "drug"])
plotdata <- as.numeric(as.character(this_analysis[, "
    correlation"]))
names(plotdata) <- drug_names
plotdata <- sort(plotdata, decreasing = T)

barplot(plotdata, beside = T, col = "blue", ylim = c(0,1)
    ,
        main = "CTRP vs GDSC, other targeted drugs")
abline(h = 0.5, col = "red", lty = "longdash")

#Cytotoxic therapies
```

```
this_analysis <- correl[(correl$drug %in% cytotoxic) & (
    correl$correlation_type == "pearson") &
                            (correl$drug_sensitivity_metric
                               == "AUC_IC50_adj"), ]
drug_names <- as.character(this_analysis[, "drug"])
plotdata <- as.numeric(as.character(this_analysis[, "
    correlation"]))
names(plotdata) <- drug_names
plotdata <- sort(plotdata, decreasing = T)

barplot(plotdata, beside = T, col = "blue", ylim = c(0,1)
    ,
         main = "CTRP vs GDSC,\n cytotoxic drugs")
abline(h = 0.5, col = "red", lty = "longdash")

dev.off()
```

Adjusted $AUC_{IC50}$ gave improved correlation of the drug sensitivity data in CCLE and GDSC when compared to the similar analyses that used unadjusted AUC values.[7] The correlation has improved by more than 0.1 for 6 out of 15 drugs: nilotinib, PD-0325901, PLX-4720, nutlin-3, sorafenib, and erlotinib, (Figure 4.3a).[7] CCLE and CTRP drug sensitivities saw a positive correlation (Figure 4.3b) with 12 out of 17 drugs displaying a moderate correlation. The researchers analyzed 61 drugs by both GDSC and CTRP and divided compounds based on the primary mechanism of action. The drug responses of kinase inhibitors showed the best correlation with 12 out of 35 drugs with a correlation of greater than 0.5 , and only 4 out of 17 other target therapies demonstrated moderate correlation.[7] The AUC can be calculated for the specified range of drug concentrations using functions from the the PharmacoGx package but analysis is limited to only the CCLE and GDSC databases.[7]

## References

1. Issa A. ETHICAL PERSPECTIVES ON PHARMACOGENOMIC PROFILING IN THE DRUG DEVELOPMENT PROCESS. Nature Reviews Drug Discovery. 2002;1(4):300-308. doi:10.1038/nrd771.

2. Relling M, Evans W. Pharmacogenomics in the clinic. Nature. 2015;526(7573):343-350. doi:10.1038/nature15817.

3. Fan J, Liu H. Statistical analysis of big data on pharmacogenomics. Adv Drug Deliv Rev. 2013;65(7):987-1000. doi:10.1016/j.addr.2013.04.008.

4. Smirnov P, Safikhani Z, El-Hachem N et al. PharmacoGx: an R package for analysis of large pharmacogenomic datasets. Bioinformatics. 2015;32(8):1244-1246. doi:10.1093/bioinformatics/btv723.

5. FAQ About Pharmacogenomics. National Human Genome Research Institute (NHGRI). 2018. Available at: https://www.genome.gov/27530645/faq-about-pharmacogenomics/. Accessed March 13, 2018.

6. Schiermeier Q, Tollefson J, Maxmen A et al. Nature Research: science journals, jobs, information and services. Naturecom. 2018. Available at: https://www.nature.com/ scitable/topicpage/pharmacogenomics-and-personalized-medicine-643. Accessed March 13, 2018.

7. Pozdeyev N, Yoo M, Mackie R, Schweppe R, Tan A, Haugen B. Integrating heterogeneous drug sensitivity data from cancer pharmacogenomic studies. Oncotarget. 2016;7(32). doi:10.18632/oncotarget.10010.