Biomarker discovery from large pharmacogenomics datasets Zhaleh Safikhani[1,2], Benjamin Haibe-Kains[1,2,3], Petr Smrinov[1, 2], and Seyed Ali Madani Tonekaboni[1, 2]

[1]Princess Margaret Cancer Centre, University Health Network, Toronto Canada
[2]Department of Medical Biophysics, University of Toronto, Toronto Canada
[3]Department of Computer Science, University of Toronto, Toronto Canada

Contents

Introduction

Pharmacogenomics holds much potential to aid in discovering drug response biomarkers and developing novel targeted therapies, leading to development of precision medicine and working towards the goal of personalized therapy. Several large experiments have been conducted, both to molecularly characterize drug dose response across many cell lines, and to examine the molecular response to drug administration. However, the experiments lack a standardization of protocols and annotations, hindering meta-analysis across several experiments. *PharmacoGx* was developed to address these challenges, by providing a unified framework for downloading and analyzing large pharmacogenomic datasets which are extensively curated to ensure maximum overlap and consistency. *PharmacoGx* is based on a level of abstraction from the raw experimental data, and allows bioinformaticians and biologists to work with data at the level of genes, drugs and cell lines. This provides a more intuitive interface and, in combination with unified curation, simplifies analyses between multiple datasets.

Installation and Settings

PharmacoGx requires that several packages are installed. However, all dependencies are available from CRAN or Bioconductor.

```
source('http://bioconductor.org/biocLite.R')
biocLite('PharmacoGx')
```

Load *PharamacoGx* into your current workspace:

```
library(PharmacoGx)
```

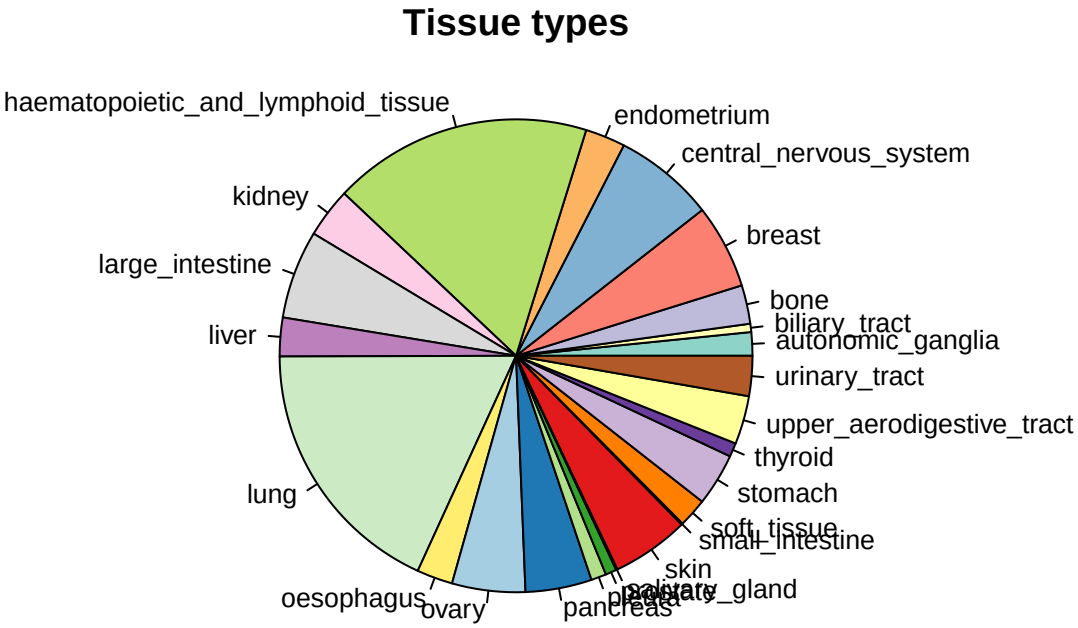


Figure 1: Tissue of origin of cell lines in CCLE study

Downloading PharmacoSet objects

We have made the PharmacoSet objects of the curated datasets available for download using functions provided in the package. A table of available PharmacoSet objects can be obtained by using the *availablePSets* function. Any of the PharmacoSets in the table can then be downloaded by calling *downloadPSet*, which saves the datasets into a directory of the users choice, and returns the data into the R session.

```
availablePSets()
GDSC <- downloadPSet("GDSC")
CCLE <- downloadPSet("CCLE")
```

Reproducibility

PharmacoGx can be used to process pharmacogenomic datasets. First we want to check the heterogenity of cell lines in one of the available psets, CCLE.

```
mycol <- c("#8dd3c7","#ffffb3","#bebada","#fb8072","#80b1d3","#fdb462",
           "#b3de69","#fccde5","#d9d9d9","#bc80bd","#ccebc5","#ffed6f",
           "#a6cee3","#1f78b4","#b2df8a","#33a02c","#fb9a99","#e31a1c",
           "#fdbf6f","#ff7f00","#cab2d6","#6a3d9a","#ffff99","#b15928")
pie(table(CCLE@cell[,"tissueid"]),
    col=mycol,
    main="Tissue types",
    radius=1,
    cex=0.8)
```

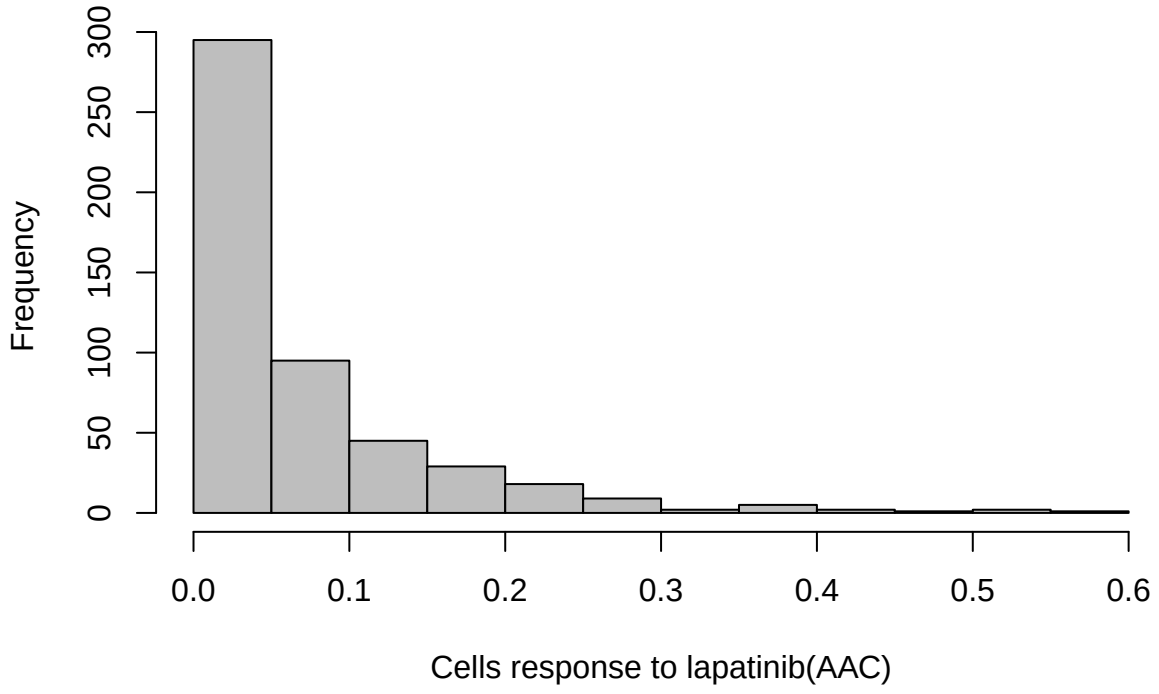


Figure 2: Cells response to lapatinib in CCLE

Pharmacological profiles

In pharmacogenomic studies Cells were also tested for their response to increasing concentrations of various compounds, and form this the IC50 and AUC were computed. These pharmacological profiles are available for all the psets in *PharmacoGx*.

```
CCLE.auc <- summarizeSensitivityProfiles(
                pSet=CCLE,
                sensitivity.measure="auc_published",
                summary.stat="median",
                verbose=FALSE)
hist(CCLE.auc["lapatinib",], xlab="Cells response to lapatinib(AAC)",
     col="gray", main="")
```

Plotting Drug-Dose Response Data

Drug-Dose response data included in the PharmacoSet objects can be conviniently plotted using the *drugDoseResponseCurve* function. Given a list of PharmacoSets, a drug name and a cell name, it will plot the drug dose response curves for the given cell-drug combination in each dataset, allowing direct comparisons of data between datasets.

```
lapatinib.aac <- CCLE.auc["lapatinib",]
cells <- names(lapatinib.aac)[
                c(which.min(lapatinib.aac),
                  which((lapatinib.aac > 0.2) & (lapatinib.aac < 0.4))[1],
                  which.max(lapatinib.aac))]
par(mfrow=c(2, 2))
drugDoseResponseCurve(drug="lapatinib", cellline=cells[1],
                      pSets=CCLE, plot.type="Fitted",
```

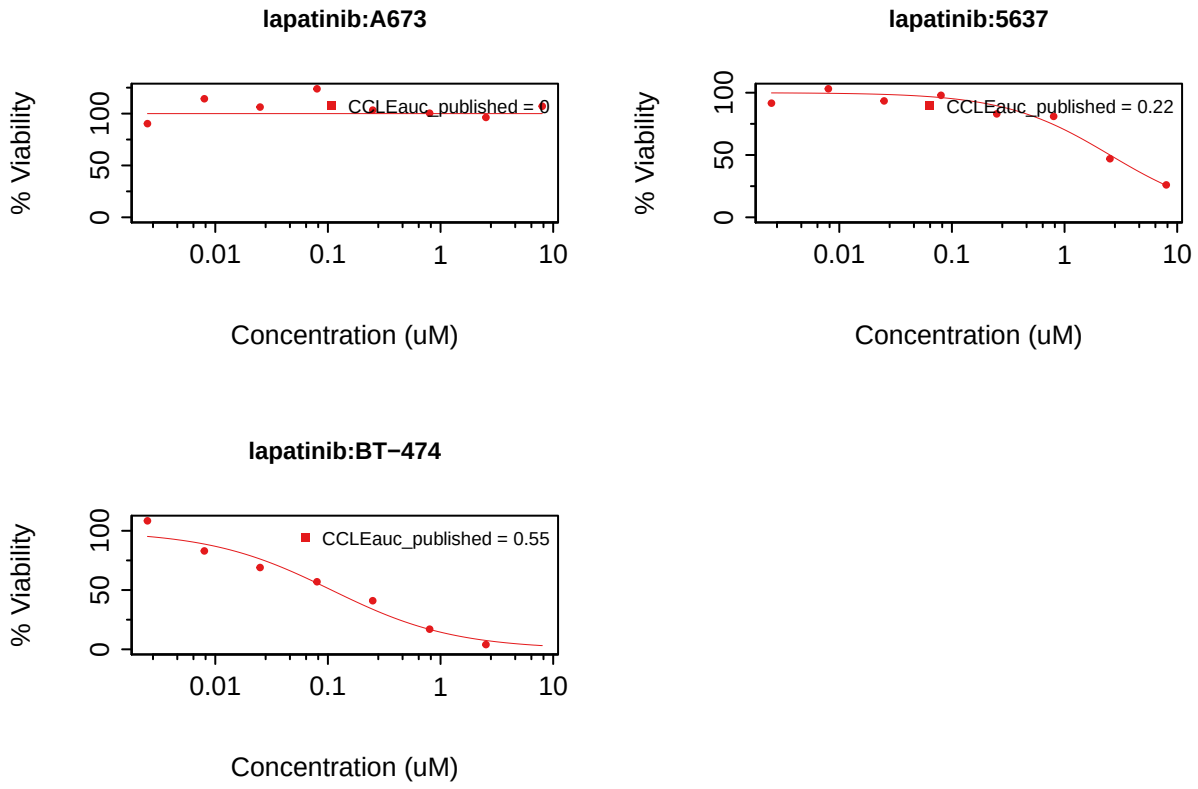


Figure 3: Cells response to lapatinib in CCLE

```
                      legends.label="auc_published")
drugDoseResponseCurve(drug="lapatinib", cellline=cells[2],
                      pSets=CCLE, plot.type="Fitted",
                      legends.label="auc_published")
drugDoseResponseCurve(drug="lapatinib", cellline=cells[3],
                      pSets=CCLE, plot.type="Fitted",
                      legends.label="auc_published")
```

Replication

In this section we will investigate the consistency between the GDSC and CCLE datasets. In both CCLE and GDSC, the transcriptome of cells was profiled using an Affymatrix microarray chip. Cells were also tested for their response to increasing concentrations of various compounds, and form this the IC50 and AUC were computed. However, the cell and drugs names used between the two datasets were not consistent. Furthermore, two different microarray platforms were used. However, *PharmacoGx* allows us to overcome these differences to do a comparative study between these two datasets.

GDSC was profiled using the hgu133a platform, while CCLE was profiled with the expanded hgu133plus2 platform. While in this case the hgu133a is almost a strict subset of hgu133plus2 platform, the expression information in *PharmacoSet* objects is summarized by Ensemble Gene Ids, allowing datasets with different platforms to be directly compared. The probe to gene mapping is done using the BrainArray customCDF for each platform [**?**].

To begin, you would load the datasets from disk or download them using the *downloadPSet* function above.

We want to investigate the consistency of the data between the two datasets. The common intersection between the datasets can then be found using *intersectPSet*. We create a summary of the gene expression and drug sensitivity measures for both datasets, so we are left with one gene expression profile and one sensitivity profile per cell line within each dataset. We can then compare the gene expression and sensitivity measures between the datasets using a standard correlation coefficient.

```
library(Biobase)
```

```
## Loading required package: BiocGenerics
```

```
## Loading required package: parallel
```

```
##
## Attaching package: 'BiocGenerics'
```

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

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, cbind, colMeans,
##     colnames, colSums, do.call, duplicated, eval, evalq, Filter,
##     Find, get, grep, grepl, intersect, is.unsorted, lapply,
##     lengths, Map, mapply, match, mget, order, paste, pmax,
##     pmax.int, pmin, pmin.int, Position, rank, rbind, Reduce,
##     rowMeans, rownames, rowSums, sapply, setdiff, sort, table,
##     tapply, union, unique, unsplit, which, which.max, which.min
```

```
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
```

```
common <- intersectPSet(pSets = list("CCLE"=CCLE, "GDSC"=GDSC),
                        intersectOn = c("cell.lines", "drugs"),
                        strictIntersect = TRUE)
```

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

```
drugs <- drugNames(common$CCLE)

##Example of concordant and discordant drug curves
cases <- rbind(
  c("CAL-85-1", "17-AAG"),
  c("HT-29", "PLX4720"),
  c("COLO-320-HSR", "AZD6244"),
  c("HT-1080", "PD-0332991"))

par(mfrow=c(2, 2))
for (i in 1:nrow(cases)) {
  drugDoseResponseCurve(pSets=common,
```

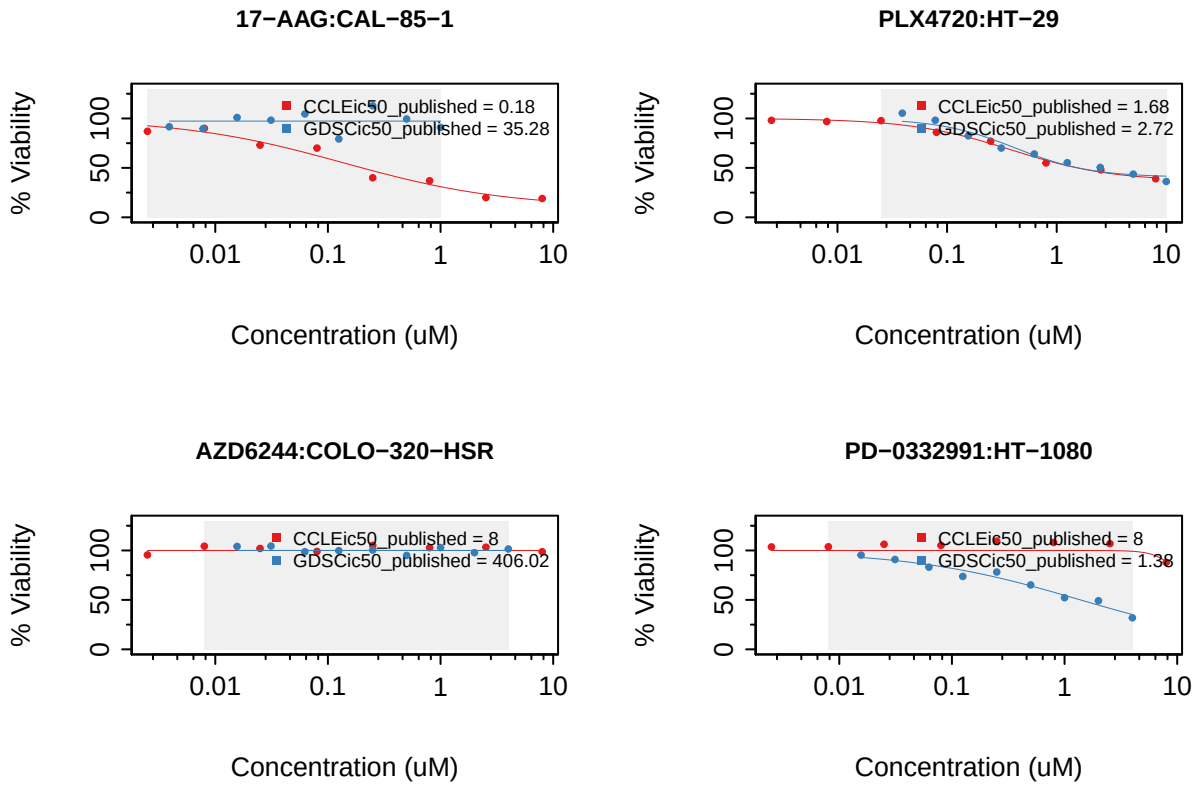


Figure 4: Consistency of drug response curves across studies

```
                    drug=cases[i,2],
                    cellline=cases[i,1],
                    legends.label="ic50_published",
                    plot.type="Fitted",
                    ylim=c(0,130))
}
```

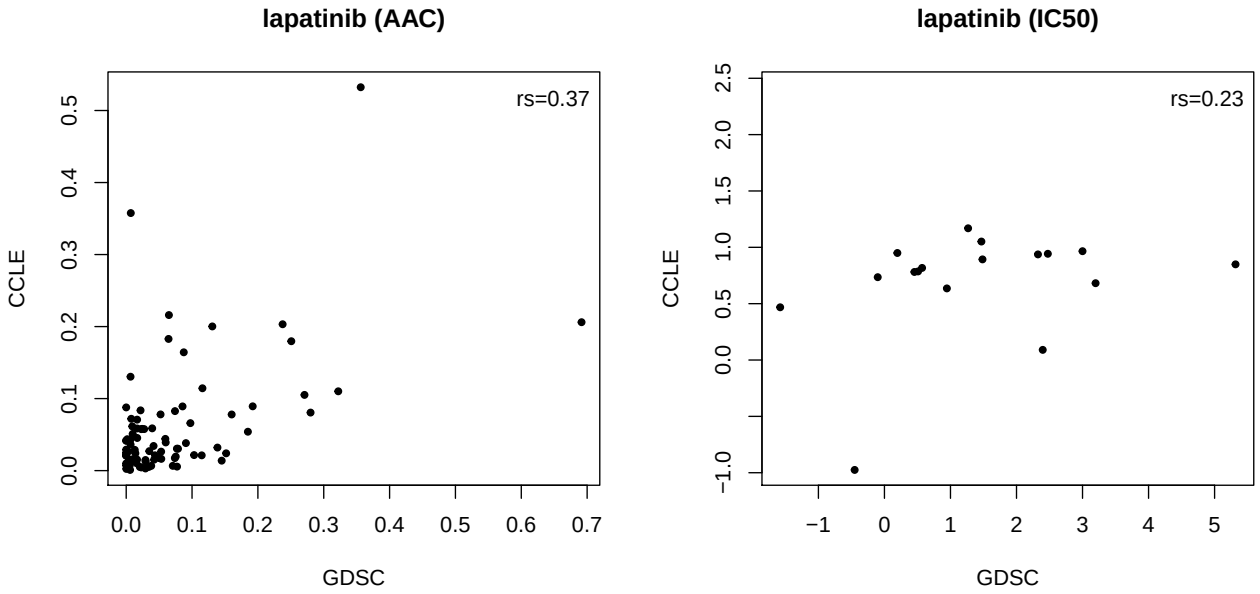
Consistency of pharmacological profiles

```
  ##AAC scatter plot
GDSC.aac <- summarizeSensitivityProfiles(
    pSet=common$GDSC,
    sensitivity.measure='auc_recomputed',
    summary.stat="median",
    verbose=FALSE)
CCLE.aac <- summarizeSensitivityProfiles(
  pSet=common$CCLE,
  sensitivity.measure='auc_recomputed',
  summary.stat="median",
  verbose=FALSE)

GDSC.ic50 <- log10(summarizeSensitivityProfiles(
    pSet=common$GDSC,
    sensitivity.measure='ic50_recomputed',
```

```
    summary.stat="median",
    verbose=FALSE))
CCLE.ic50 <- log10(summarizeSensitivityProfiles(
  pSet=common$CCLE,
  sensitivity.measure='ic50_recomputed',
  summary.stat="median",
  verbose=FALSE))

  drug <- "lapatinib"
par(mfrow=c(1, 2))
plot(GDSC.aac[drug,], CCLE.aac[drug,],
       xlab="GDSC", ylab="CCLE",
       main="lapatinib (AAC)", pch=20)
legend("topright",
         legend=sprintf("rs=%s",
                round(cor(GDSC.aac[drug,],
                          CCLE.aac[drug,],
                          method="spearman",
                          use="pairwise.complete.obs"),
                      digits=2)),
         bty="n")
plot(GDSC.ic50[drug,], CCLE.ic50[drug,],
       xlab="GDSC", ylab="CCLE",
       main="lapatinib (IC50)", pch=20)
legend("topright",
         legend=sprintf("rs=%s",
                round(cor(GDSC.ic50[drug,],
                          CCLE.ic50[drug,],
                          method="spearman",
                          use="pairwise.complete.obs"),
                      digits=2)),
         bty="n")
```



consistency assessment improved by Modified Concordance Index

To better assess the concordance of multiple pharmacogenomic studies we introduced the modified concordance index (mCI). Recognizing that the noise in the drug screening assays is high and may yield to inaccurate sensitive-based ranking of cell lines with close AAC values, the mCI only considers cell line pairs with drug sensitivity (AAC) difference greater than δ .

```
library(devtools)
devtools::install_github("bhklab/mci", ref="light")
```

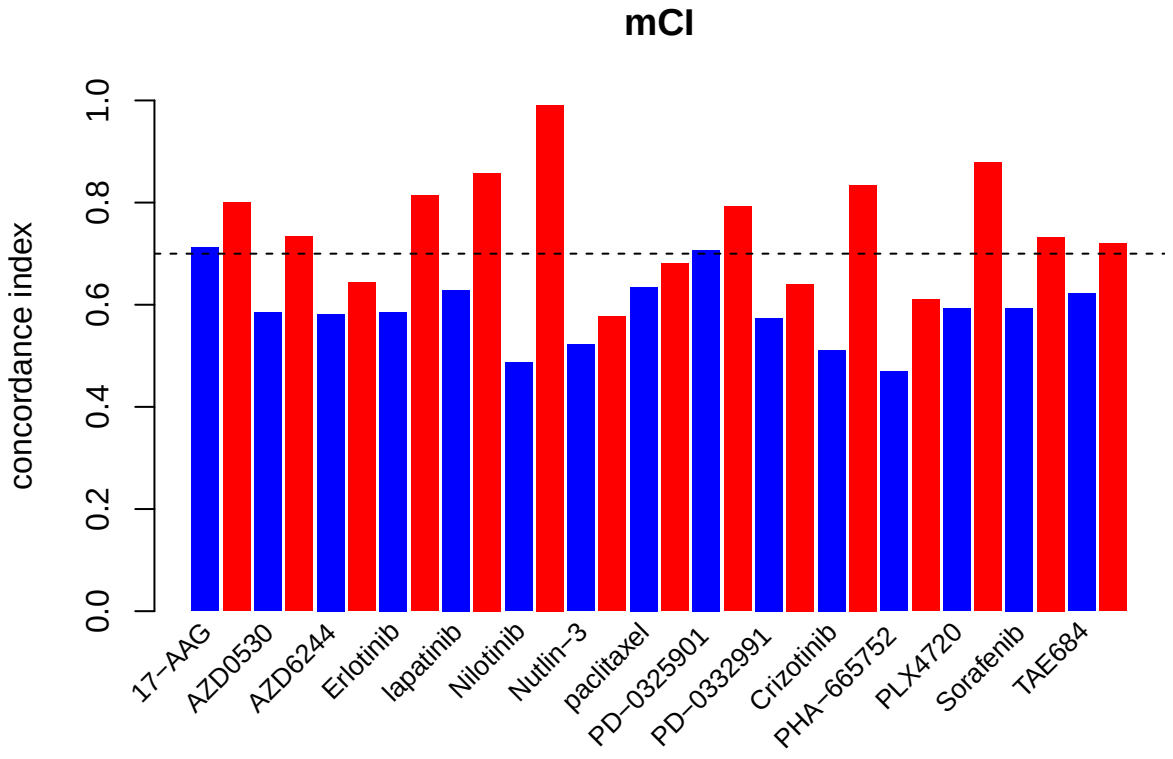
```
## Downloading GitHub repo bhklab/mci@light
## from URL https://api.github.com/repos/bhklab/mci/zipball/light
```

```
## Installing mCI
```

```
## '/Library/Frameworks/R.framework/Resources/bin/R' --no-site-file  \
##   --no-environ --no-save --no-restore --quiet CMD INSTALL  \
##   '/private/var/folders/h5/35bht58d3r79crq08cgj1x4m0000gn/T/RtmpHzRj1V/devtools63fc17af0259/bhklab-m
##   --library='/Library/Frameworks/R.framework/Versions/3.4/Resources/library'  \
##   --install-tests
```

```
##
```

```
library(mCI)
c_index <-  mc_index <- NULL
for(drug in drugs){
  tt <- mCI::paired.concordance.index(GDSC.aac[drug,], CCLE.aac[drug,], delta.pred=0, delta.obs=0, alte
  c_index <- c(c_index, tt$cindex)
  tt <- mCI::paired.concordance.index(GDSC.aac[drug,], CCLE.aac[drug,], delta.pred=0.2, delta.obs=0.2, a
  mc_index <- c(mc_index, tt$cindex)
}
mp <- barplot(as.vector(rbind(c_index, mc_index)), beside=TRUE, col=c("blue", "red"), ylim=c(0, 1), yla
text(mp, par("usr")[3], labels=as.vector(rbind(drugs, rep("", 15))), srt=45, adj=c(1.1,1.1), xpd=TRUE, 
abline(h=.7, lty=2)
```



Known Biomarkers

The association between molecular features and response to a given drug is modelled using a linear regression model adjusted for tissue source:

$$Y = \beta_0 + \beta_i G_i + \beta_t T + \beta_b B$$

where Y denotes the drug sensitivity variable, G_i, T and B denote the expression of gene i, the tissue source and the experimental batch respectively, and βs are the regression coefficients. The strength of gene-drug association is quantified by β_i, above and beyond the relationship between drug sensitivity and tissue source. The variables Y and G are scaled (standard deviation equals to 1) to estimate standardized coefficients from the linear model. Significance of the gene-drug association is estimated by the statistical significance of β_i (two-sided t test). P-values are then corrected for multiple testing using the false discovery rate (FDR) approach.

As an example of the reproducibility of biomarker discovery across pharmacogenomic studies, we can model the significance of the association between two drugs and their known biomarkers in CCLE and GDSC. We examine the association between drug *17-AAG* and gene *NQO1*, as well as drug *PD-0325901* and gene *BRAF*:

```
features <- PharmacoGx::fNames(CCLE, "rna")[
  which(featureInfo(CCLE,
                    "rna")$Symbol == "NQO1")]
  ccle.sig.rna <- drugSensitivitySig(pSet=CCLE,
                                     mDataType="rna",
                                     drugs=c("17-AAG"),
                                     features=features,
                                     sensitivity.measure="auc_published",
                                     molecular.summary.stat="median",
                                     sensitivity.summary.stat="median",
                                     verbose=FALSE)
```

```
    gdsc.sig.rna <- drugSensitivitySig(pSet=GDSC,
                                       mDataType="rna",
                                       drugs=c("17-AAG"),
                                       features=features,
                                       sensitivity.measure="auc_published",
                                       molecular.summary.stat="median",
                                       sensitivity.summary.stat="median",
                                       verbose=FALSE)
    ccle.sig.mut <- drugSensitivitySig(pSet=CCLE,
                                       mDataType="mutation",
                                       drugs=c("PD-0325901"),
                                       features="BRAF",
                                       sensitivity.measure="auc_published",
                                       molecular.summary.stat="and",
                                       sensitivity.summary.stat="median",
                                       verbose=FALSE)
    gdsc.sig.mut <- drugSensitivitySig(pSet=GDSC,
                                       mDataType="mutation",
                                       drugs=c("PD-0325901"),
                                       features="BRAF",
                                       sensitivity.measure="auc_published",
                                       molecular.summary.stat="and",
                                       sensitivity.summary.stat="median",
                                       verbose=FALSE)
    ccle.sig <- rbind(ccle.sig.rna, ccle.sig.mut)
    gdsc.sig <- rbind(gdsc.sig.rna, gdsc.sig.mut)
    known.biomarkers <- cbind("GDSC effect size"=gdsc.sig[,1],
                              "GDSC pvalue"=gdsc.sig[,6],
                              "CCLE effect size"=ccle.sig[,1],
                              "CCLE pvalue"=ccle.sig[,6])
    rownames(known.biomarkers) <- c("17-AAG + NQO1","PD-0325901 + BRAF")
    library(xtable)
    xtable(known.biomarkers, digits=c(0, 2, -1, 2, -1), caption='Concordance of biomarkers across stuudi
```

% latex table generated in R 3.4.2 by xtable 1.8-2 package % Sat Mar 3 16:08:07 2018

	GDSC effect size	GDSC pvalue	CCLE effect size	CCLE pvalue
17-AAG + NQO1	0.56	2.7E-34	0.60	9.6E-26
PD-0325901 + BRAF	0.82	1.2E-08	0.84	1.1E-09

Table 1: Concordance of biomarkers across stuudies

```
    par(mfrow=c(2, 2))
    CCLE_expr <- t(exprs(summarizeMolecularProfiles(CCLE, mDataType="rna", fill.missing=FALSE)))
    CCLE_cells <- intersect(rownames(CCLE_expr), colnames(CCLE.aac))
    plot(CCLE.aac["17-AAG", CCLE_cells], CCLE_expr[CCLE_cells, features],
     main="CCLE + 17-AAG + NQO1",
     cex.main=1, ylab="Predictions", xlab="drug sensitivity", pch=20, col="gray40")

    GDSC_expr <- t(exprs(summarizeMolecularProfiles(GDSC, mDataType="rna", fill.missing=FALSE)))
```

```
## Summarizing rna molecular data for:  GDSC
```

|
| | 0% |

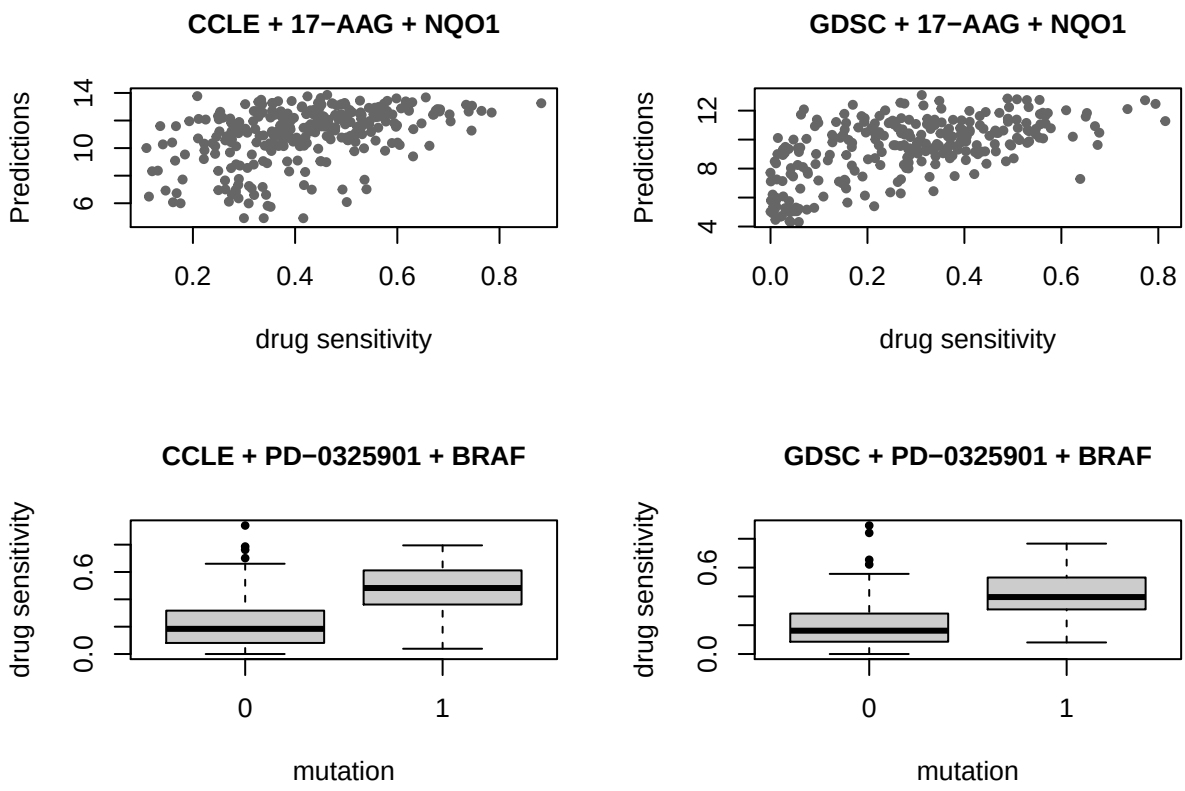
=	2%
==	3%
===	5%
====	7%
=====	8%
======	10%
=======	11%
=========	13%
==========	15%
===========	16%
============	18%
=============	20%
===============	21%
================	23%
=================	25%
==================	26%
====================	28%
=====================	30%
======================	31%
========================	33%
=========================	34%
==========================	36%
============================	38%
=============================	39%
==============================	41%
================================	43%
=================================	44%
==================================	46%
====================================	48%
=====================================	49%
=======================================	51%
==	52%
==	54%
===	56%
===	57%
==	59%
==	61%
===	62%
===	64%
==	66%
==	67%
===	69%
===	70%
==	72%
==	74%
===	75%
===	77%
==	79%
==	80%
==	82%
==	84%
==	85%
==	87%
==

```
89% |
|=============================================================== |
90% |
|================================================================
| 92% |
|================================================================
| 93% |
|================================================================
| 95% |
|================================================================
| 97% |
|================================================================
| 98% |
|================================================================|
100%
```

```
    GDSC_cells <- intersect(rownames(GDSC_expr), colnames(GDSC.aac))
    plot(GDSC.aac["17-AAG", GDSC_cells], GDSC_expr[GDSC_cells, features],
     main="GDSC + 17-AAG + NQO1",
     cex.main=1, ylab="Predictions", xlab="drug sensitivity", pch=20, col="gray40")

    CCLE_mut <- t(exprs(summarizeMolecularProfiles(CCLE, mDataType="mutation", fill.missing=FALSE, summa
CCLE_cells <- intersect(rownames(CCLE_mut), colnames(CCLE.aac))
boxplot(CCLE.aac["PD-0325901", CCLE_cells]~ CCLE_mut[CCLE_cells, "BRAF"], col="gray80", pch=20, main="C
        cex.main=1, xlab="mutation", ylab="drug sensitivity")

GDSC_mut <- t(exprs(summarizeMolecularProfiles(GDSC, mDataType="mutation", fill.missing=FALSE, summary.
GDSC_cells <- intersect(rownames(GDSC_mut), colnames(GDSC.aac))
boxplot(GDSC.aac["PD-0325901", GDSC_cells]~ GDSC_mut[GDSC_cells, "BRAF"], col="gray80", pch=20, main="G
        cex.main=1, xlab="mutation", ylab="drug sensitivity")
```



Sactter Plot for know biomarkers (4 plots)

Machne Learning and Biomarker Discovery

Some of the widely used multivariate machine learning methods such as elastic net, Random Forest (RF) and Support Vector Machine (SVM) have been already implemented in the MLWorkshop. It optimizes hyperparameters of these methods in the training phase. To assess the performance of the predictive models, it implements m number of sampling with *n-fold* cross validations (CV). The performance will then be assessed by multiple metrics including pearson correlation coefficient, concordance index and modified concordance index.

```
library(mRMRe)
```

```
## Loading required package: survival
```

```
## Loading required package: igraph
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:BiocGenerics':
##
##     normalize, union
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

```
##
## Attaching package: 'mRMRe'
```

```
## The following objects are masked from 'package:Biobase':
##
##     featureData, featureNames, sampleNames
```

```
library(Biobase)
train_expr <- t(exprs(summarizeMolecularProfiles(GDSC, mDataType="rna", fill.missing=FALSE)))
```

```
## Summarizing rna molecular data for:  GDSC
```

```
|
|                                                                      |   0%
|=                                                                     |   2%
|==                                                                    |   3%
|===                                                                   |   5%
|====                                                                  |   7%
|=====                                                                 |   8%
|======                                                                |  10%
|=======                                                               |  11%
|=========                                                             |  13%
|==========                                                            |  15%
|===========                                                           |  16%
|============                                                          |  18%
|==============                                                        |  20%
|===============                                                       |  21%
|================                                                      |  23%
|=================                                                     |  25%
|==================                                                    |  26%
|====================                                                  |  28%
|=====================                                                 |  30%
|======================                                                |  31%
|=======================                                               |  33%
|========================                                              |  34%
|=========================                                             |  36%
|===========================                                           |  38%
|============================                                          |  39%
|=============================                                         |  41%
|==============================                                        |  43%
|================================                                      |  44%
|=================================                                     |  46%
|==================================                                    |  48%
|===================================                                   |  49%
|====================================                                  |  51%
|======================================                                |  52%
|=======================================                               |  54%
|========================================                              |  56%
|=========================================                             |  57%
|==========================================                            |  59%
|============================================                          |  61%
|=============================================                         |  62%
|==============================================                        |  64%
|===============================================                       |  66%
|================================================                      |  67%
|=================================================                     |  69%
```

```
|=================================================== | 70% |
|==================================================== | 72% |
|===================================================== | 74% |
|====================================================== | 75% |
|======================================================= | 77% |
|======================================================== | 79% |
|========================================================= | 80% |
|========================================================== | 82% |
|=========================================================== | 84% |
|============================================================ | 85% |
|============================================================= | 87% |
|============================================================== |
89% |
|=============================================================== |
90% |
|================================================================
| 92% |
|================================================================
| 93% |
|================================================================
| 95% |
|================================================================
| 97% |
|================================================================
| 98% |
|================================================================|
100%
```

```
aac <- summarizeSensitivityProfiles(GDSC, sensitivity.measure="auc_recomputed", drug="lapatinib", fill.m
cells <- intersect(rownames(train_expr), names(aac))

df <- as.matrix(cbind(train_expr[cells,], "lapatinib"=aac[cells]))

library(devtools)
install_github("bhklab/PharmacoGx-ML")
```

```
## Downloading GitHub repo bhklab/PharmacoGx-ML@master
## from URL https://api.github.com/repos/bhklab/PharmacoGx-ML/zipball/master
```

```
## Installing PharmacoGxML
```

```
## '/Library/Frameworks/R.framework/Resources/bin/R' --no-site-file  \
##   --no-environ --no-save --no-restore --quiet CMD INSTALL  \
##   '/private/var/folders/h5/35bht58d3r79crq08cgj1x4m0000gn/T/RtmpHzRj1V/devtools63fc31d19478/bhklab-Ph
##   --library='/Library/Frameworks/R.framework/Versions/3.4/Resources/library'  \
##   --install-tests
```

```
##
```

```
library(PharmacoGxML)
```

```
##
## Attaching package: 'PharmacoGxML'
```

```
## The following object is masked from 'package:survival':
##
##     ridge
```

```
library(Hmisc)
```

```
## Loading required package: lattice
```

```
## Loading required package: Formula
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:xtable':
##
##     label, label<-
```

```
## The following objects are masked from 'package:Biobase':
##
##     combine, contents
```

```
## The following object is masked from 'package:BiocGenerics':
##
##     combine
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, round.POSIXt, trunc.POSIXt, units
```

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-13
```

```
library(caret)
```

```
##
## Attaching package: 'caret'
```

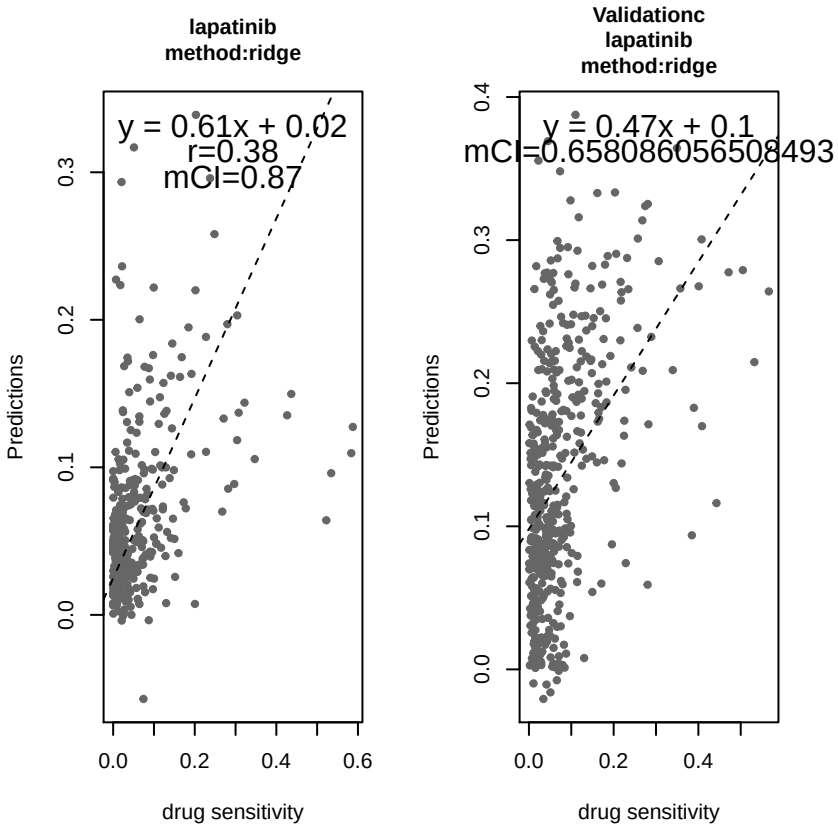
```
## The following object is masked from 'package:survival':
##
##     cluster
```

```
par(mfrow=c(1, 3))
method <- "ridge"
res <- optimization(train=df[, -ncol(df), drop=F],
                    labels=t(df[, ncol(df), drop=F]),
                    method="ridge",
                    folds.no=2,
                    sampling.no=1,
                    features.no=100,
                    feature.selection="mRMR",
                    assessment=c("corr", "mCI"))
```

[1] "1, lapatinib, method: ridge, fold#: 1 sampling#: 1" [1] "1, lapatinib, method: ridge, fold#: 2 sampling#: 1"

```
model <- res$model
##Prediction
validation_expr <- summarizeMolecularProfiles(CCLE, mDataType="rna", fill.missing=FALSE)
```

```
actual_labels <- summarizeSensitivityProfiles(CCLE, sensitivity.measure="auc_recomputed", drug="lapatini
validation_labels <- prediction(models=model$lapatinib$sampling_1,
                                validation.set=t(exprs(validation_expr)),
                                validation.labels=actual_labels,
                                method=method,
                                assessment="mCI")
```



Limitations/ Future direction

Session Info

This document was generated with the following R version and packages loaded:

- R version 3.4.2 (2017-09-28), `x86_64-apple-darwin15.6.0`
- Locale: `en_CA.UTF-8/en_CA.UTF-8/en_CA.UTF-8/C/en_CA.UTF-8/en_CA.UTF-8`
- Running under: `OS X El Capitan 10.11.6`
- Matrix products: default
- BLAS: `/Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRblas.0.dylib`
- LAPACK: `/Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRlapack.dylib`
- Base packages: base, datasets, graphics, grDevices, methods, parallel, stats, utils

- Other packages: Biobase 2.38.0, BiocGenerics 0.24.0, caret 6.0-77, devtools 1.13.4, foreach 1.4.3, Formula 1.2-2, ggplot2 2.2.1, glmnet 2.0-13, Hmisc 4.0-3, igraph 1.1.2, lattice 0.20-35, Matrix 1.2-11, mCI 0.1.0, mRMRe 2.0.7, PharmacoGx 1.8.0, PharmacoGxML 0.1.0, survival 2.41-3, xtable 1.8-2
- Loaded via a namespace (and not attached): acepack 1.4.1, assertthat 0.2.0, backports 1.1.1, base64enc 0.1-3, bindr 0.1, bindrcpp 0.2, BiocParallel 1.12.0, bitops 1.0-6, broom 0.4.3, caTools 1.17.1, celestial 1.4.1, checkmate 1.8.5, class 7.3-14, cluster 2.0.6, codetools 0.2-15, colorspace 1.3-2, compiler 3.4.2, curl 3.0, CVST 0.2-1, data.table 1.10.4-3, ddalpha 1.3.1, DEoptimR 1.0-8, digest 0.6.12, dimRed 0.1.0, downloader 0.4, dplyr 0.7.4, DRR 0.0.2, evaluate 0.10.1, fastmatch 1.1-0, fgsea 1.4.0, foreign 0.8-69, gdata 2.18.0, git2r 0.19.0, glue 1.2.0, gower 0.1.2, gplots 3.0.1, grid 3.4.2, gridExtra 2.3, gtable 0.2.0, gtools 3.5.0, highr 0.6, htmlTable 1.9, htmltools 0.3.6, htmlwidgets 0.9, httr 1.3.1, ipred 0.9-6, iterators 1.0.8, kernlab 0.9-25, KernSmooth 2.23-15, knitr 1.20, latticeExtra 0.6-28, lava 1.5.1, lazyeval 0.2.1, limma 3.34.1, lsa 0.73.1, lubridate 1.7.1, magicaxis 2.0.3, magrittr 1.5, mapproj 1.2-5, maps 3.2.0, marray 1.56.0, MASS 7.3-47, memoise 1.1.0, mnormt 1.5-5, ModelMetrics 1.1.0, munsell 0.4.3, NISTunits 1.0.1, nlme 3.1-131, nnet 7.3-12, piano 1.18.0, pkgconfig 2.0.1, plotrix 3.6-6, plyr 1.8.4, pracma 2.1.1, prodlim 1.6.1, psych 1.7.8, purrr 0.2.4, R6 2.2.2, RANN 2.5.1, RColorBrewer 1.1-2, Rcpp 0.12.15, RcppRoll 0.2.2, recipes 0.1.1, relations 0.6-7, reshape2 1.4.2, rlang 0.1.4, rmarkdown 1.9, robustbase 0.92-8, rpart 4.1-11, rprojroot 1.3-2, scales 0.5.0, sets 1.0-17, sfsmisc 1.1-1, slam 0.1-40, sm 2.2-5.4, SnowballC 0.5.1, splines 3.4.2, stats4 3.4.2, stringi 1.1.6, stringr 1.2.0, tibble 1.3.4, tidyr 0.7.2, tidyselect 0.2.3, timeDate 3042.101, tools 3.4.2, withr 2.1.0, yaml 2.1.14