R Toolkit for Biomarker Discovery

Hendrik Weisser, Carmen Livi, Georgia Tsagkogeorga

STORM Therapeutics Ltd, Cambridge, UK
hendrik.weisser@stormtherapeutics.com

Introduction

In the age of precision medicine, discovery of biomarkers for drug efficacy is a critical part of the drug discovery process. While the end goal is selection of patients for therapy, important opportunities for biomarker discovery arise at several stages:

- Based on genetic data (e.g. dependency maps in cancer cell lines) during initial target validation;
- Based on *in vitro* or *in vivo* response data generated with compounds;
- Based on patient data from clinical trials.

From a data analysis perspective, the general approach is the same in all cases: Evaluate available "features" (e.g. omics data) for associations with the "response" (e.g. drug sensitivity); either one by one (univariate analysis) or in combinations (multivariate analysis). However, different methodologies are required for different data modalities (e.g. numerical/categorical).

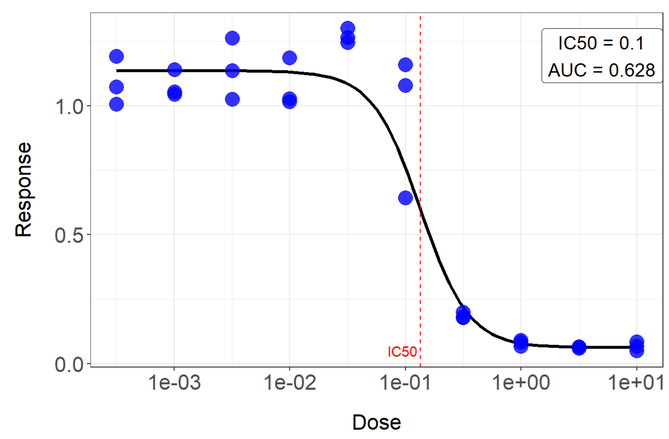
We are developing a toolkit for biomarker discovery using the R language and environment. Building on several existing R packages, we are creating a framework that is convenient to use, can handle various data types, and offers compelling functionality.

Our code is available at:
https://github.com/storm-therapeutics/BiomarkerDiscoveryToolkit

Dose Response Curves

Dose response curves (e.g. for drugs tested in cell lines) can be fit robustly using the R package "dr4pl". Results can be visualized and IC50 or AUC values calculated – suitable for subsequent association-finding analyses (see below).

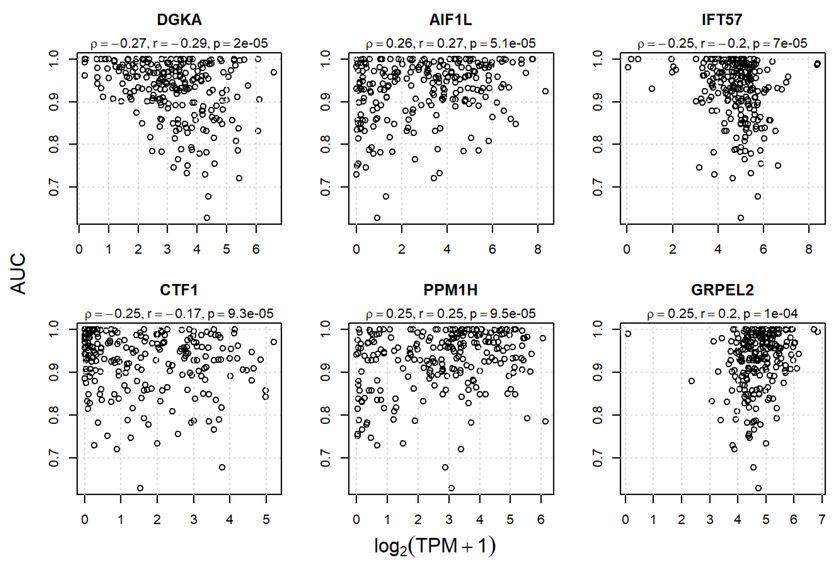
```
fits <- fit.data.long(df, "cellLineName", "conc", "poc",
                      percent=TRUE)
aucs <- sapply(fits, compute.AUC)
plot.curve.fit(fits[[1]])
```



Correlation-Based Analyses

To find univariate associations between a numerical response (e.g. *in vitro* drug sensitivity measured as AUC) and numerical features (e.g. gene expression data), we run a correlation-based analysis. By default, Spearman and Pearson correlation coefficients as well as p-values (based on shuffled null distributions) are calculated.
Top hits can be visualized as scatter plots.

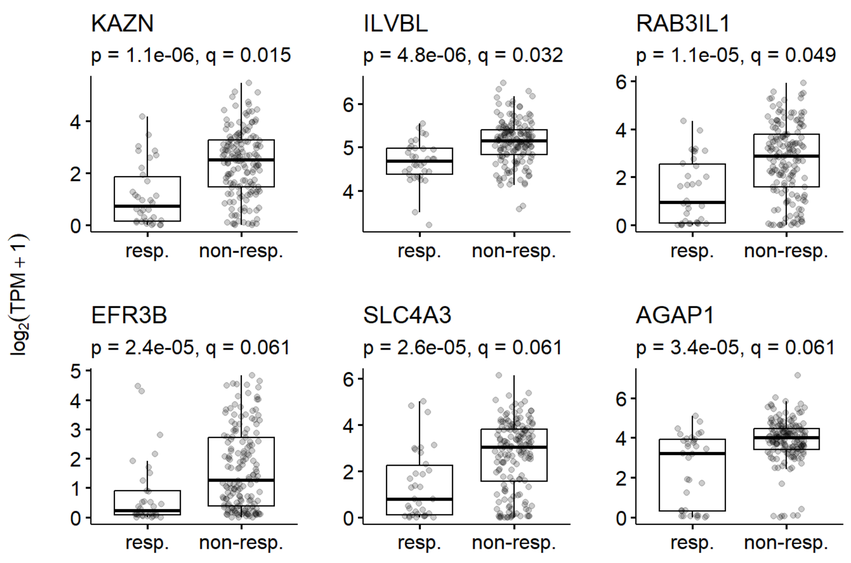
```
cors <- correlation.analysis(aucs, exp.mat)
plot.correlations(cors, aucs, exp.mat, 1:6, title="",
                  xlab=expression(log[2](TPM+1)), ylab="AUC")
```



Group-Based Analyses

A group-based analysis uses statistical tests (e.g. Wilcoxon/Mann-Whitney) to find univariate associations between a categorical response (e.g. responders/non-responders of a treatment) and numerical features (e.g. gene expression data).
Top hits are visualized as box plots.

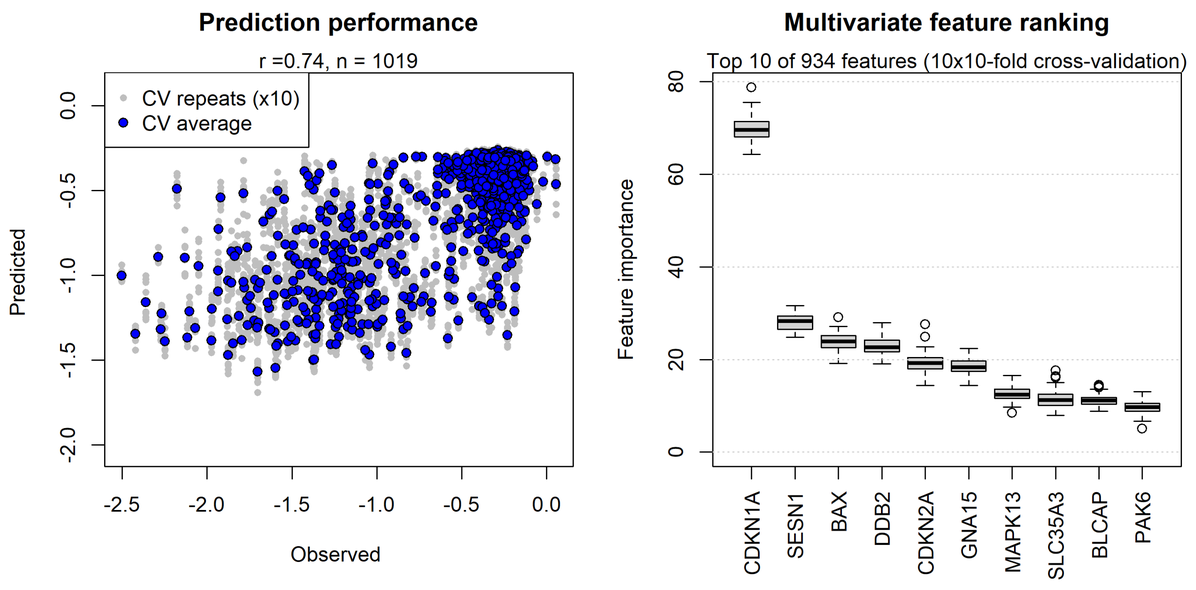
```
stats <- group.analysis(aucs, exp.mat)
plot.groups(stats, groups, exp.mat, 1:6,
            ylab=expression(log[2](TPM+1)))
```



Multivariate Analyses

Multivariate analysis selects a limited set of features to generate a model with optimal prediction performance. Here, we train random forest models using cross-validation and backwards feature elimination, powered by the R package "caret".

```
model <- multivariate.analysis(effect, exp.mat)
plot.predictions(model, asp=1)
plot.feature.ranking(model)
```



Gene Set Enrichment Analyses

Results from univariate analyses on gene data can be aggregated to the level of pathways or functional annotations using gene set enrichment analysis (GSEA), based on "clusterProfiler" and related R packages. We provide a single function for GSEA using Gene Ontology annotations ("Biological Process" and "Molecular Function"), Reactome pathways and MSigDB Hallmark gene sets.
Dot plots are used for visualization, with "+" and "-" annotations indicating enrichment at the top or bottom of the gene ranking.

```
gse <- gsea.all(get.gsea.input(cors))
dotplot.direction(gse$Hallmark, 10, label_format=50)
```