Sad Clown Factory Report

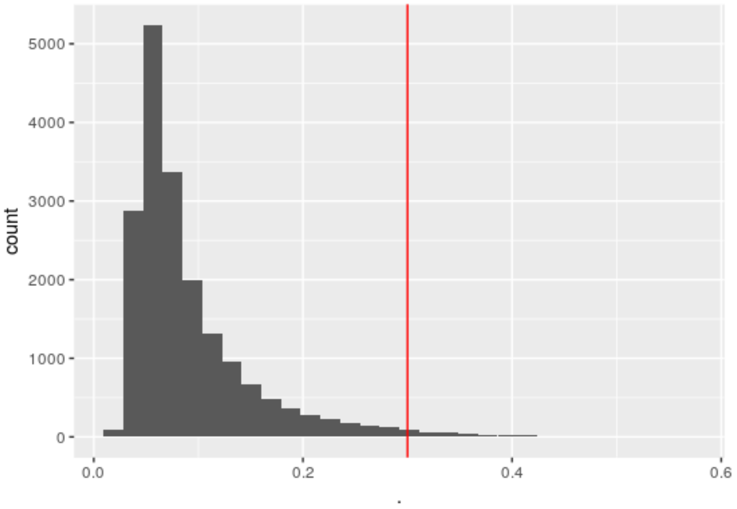
Lawrence Hsu and Philip Robinson

Contributions

Lawrence primarily focused on support vector machines (SVM). Philips primarily focused on random forest. Both authors discussed on strategies how features would be preprocessed before feature selections. Lawrence provided sample codes of using caret which were then converted into functions later on. Lastly, Philips cleaned up the code to be readable and reproducible.

Feature Selection

Before feature selection was performed, the features were preprocessed to make the project more manageable. The coefficient of variance was evaluated for each feature and the cutoff was originally 0.3. Coefficient of variance was evaluated because some genes are invariant and are not useful in analyzing the effectiveness of the drug. The cutoff was selected based on the histogram of the coefficient variances. Originally, we also incorporated a cutoff of highly correlated features. When discussed with Dr. Laderas and Shannon, they implied this cutoff could limit explainability (we would run into issues as certain genes may only work in tandem with another and if someone was to test it separately, they may receive no signals). However we are primary concerned with classifications. We opted to remove this correlation filter.



As we progressed through the project, we eventually added a cut-off for correlation towards success of each individual drug. We chose to correlate to each drug because when we modeled the data, we noticed certain features were more correlated to particular drugs than others. Features that did under the threshold of 0.3 were removed from training sets. The reason we set the threshold so low was due to the fact we wanted to use random forest as a feature selection and we wanted to leave in features that were highly correlated.

Random forest was ultimately what we ended up using as a feature selection for the following reasons. 1) The nature of the data. As we discussed in class, genes are hard to predict when they’re turned on and off and genes can be linked to one another (shown as collinearity). Furthermore, when we tried to run a recursive feature elimination using a linear approach, caret produces an error stating there might be colinearity. 2) Non-parametric. The data we were given was too limited, it was difficult to make an educated decision about the data distributions.

We also decided to allow for adding specific features to the automatic features set, as cell subtype was often excused from the set regardless of it’s high correlation with several target drugs.

SVM Approach

In our initial attempts, we used a recursive feature selection using a radial kernel to select features for our support vector machine. However, this approach was wrong because the function would only return a few features and when we tested these selected features we saw many support vectors, and many false positive and false negatives. We tried including resampling within the training sets to offset class imbalance; however, that did not help due to the limited number of observations. Downsampling would further reduce the size of the fold training set leading to underfitting/overfitting. Upsampling would not help the minority class because it might continue to resample the same observation from that class due to limited observations.

Our next attempt, we performed a feature selection using a random forest recursive feature selection with cross validation of 3 folds with 5 repeats and then inputted the features into the SVM with linear kernel to be trained. In addition, feature selection was performed for each particular drug. The biggest problem with this feature selection approach is it is a greedy method as such it does not care about the implications of the combinations it generates and it can overfit. In several attempts of training the models, we noticed that certain drugs performed mediocrely (60%~ accuracy) and others performed very well (80-90% accuracy). The second thing we checked was the confusion matrix for each drug to make sure that there was not a high number of false positive and false negatives. Lastly, we checked the number of support vectors because if the model lists that it has a high accuracy but a high number of support vectors that would mean the model is overfitted. The support vectors for each model never exceeded 12. The score of the submitted model was 63%.

We also tried to using a radial SVM with random forest feature selection to see if we would get a better score. We performed the same checks as we did with linear SVM and found that there were more support vectors used. Meaning, a radial kernel isn’t a good model to be fitting the data because it might be overfitting. In the end, we did not submit a radial model and continue to tune linear SVM.

Random Forest Approach

For our meta-algorithm selected use of random forests, without significant feature filtering. The expectation is that we could retrieve the prioritized features and compare them to learned feature subsets in future models. Our primary mechanisms for scoring each Random Forest was Accuracy measured by K-Fold Cross Validation and eventually Out of Bag Error.

In our first attempt, we batch filtered features based on co-correlation rules. This generated our first strawman scoring 56%. We originally found our Accuracy scores to be a poor indicator of performance, and attributed this to our small dataset. We later decided that decreasing the number of folds and increasing our repeats would yield preferable metrics. In order to best understand whether accuracy was giving us useful data, we would review the confusion matrices and generated predictions on our train data to compare against.[1]

With the advice we received (mention in earlier section) we removed co-correlation. When we, instead, batch filtered on success-correlation to the the model improved significantly with little other tuning, scoring at 63%. Once we had this high performing forest we were able to extract the 'most influential' features for that tree, and compare them to the smaller feature set selected by for the SVM. Ideally, we prefer the SVM with small set of features as it may provide more targeted testing options. Since Random Forests scored best, yet was least informative, we are using information derived from our Random Forests to improve the SVMs.

[1] Several submissions ended up scoring near 30%, which inspired re-checking our predictions vs expected results in tabular form. We found that at some point in our development we flipped our reported class bits. The code generated to check correct reporting is now baked into our processing pipeline.

Comparisons

The next step was to look into a boxplots of the features that were selected via random forest for each drug and the goal was to see if each features was a good predictor for each drug. We plotted boxplots of the outcomes for each genes and manually tuned the parameters. Features that had overlapping boxes were removed from the function and retested.

The features selected by `recursive feature elimination` for the SVMs are often a subset of the 10 most influential features of the Random Forest models. It would be interesting to expand features, per drug, to include the most differentiable remaining features of this set.

Future

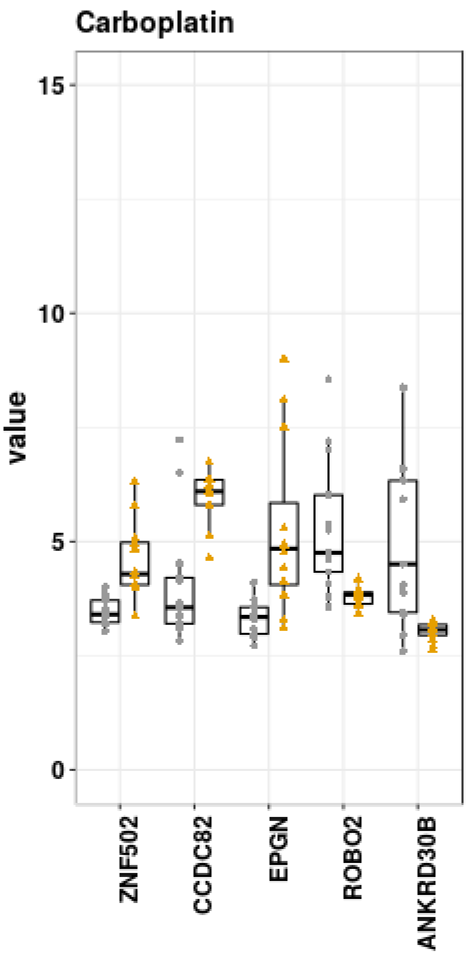
We think it would be interesting to modify our processing pipeline to use the prescribed below.

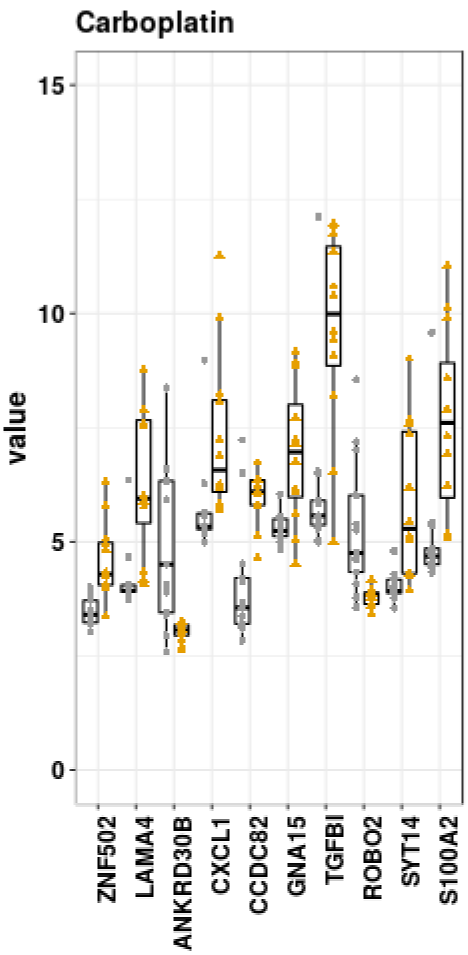
1. Select loose with our correlation and coefficient of variation bounds
2. Reduce features with random-forest `recursive features select`
3. Add celltype features (and perhaps key additional genes features)
4. * Project our data set using LDA or PCA to increase differentiability
5. * Learn our SVMs on the projected space

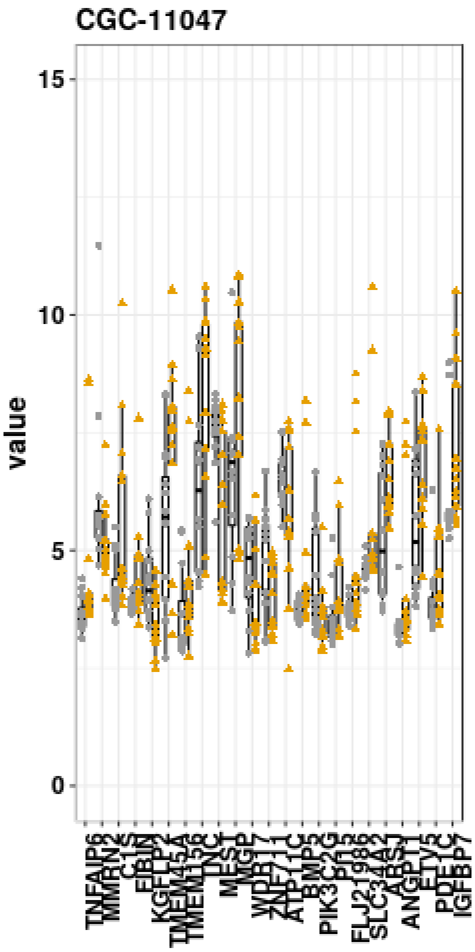
This would have the advantage of employing obscuring learning techniques that abstract away the biological data, while preserving communicability of the model, as we would be projecting from a fairly small feature set.

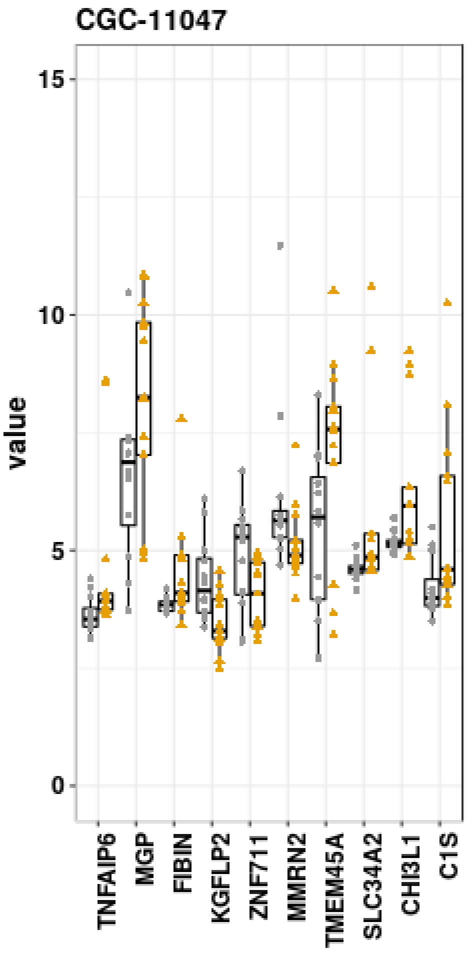
Appendix

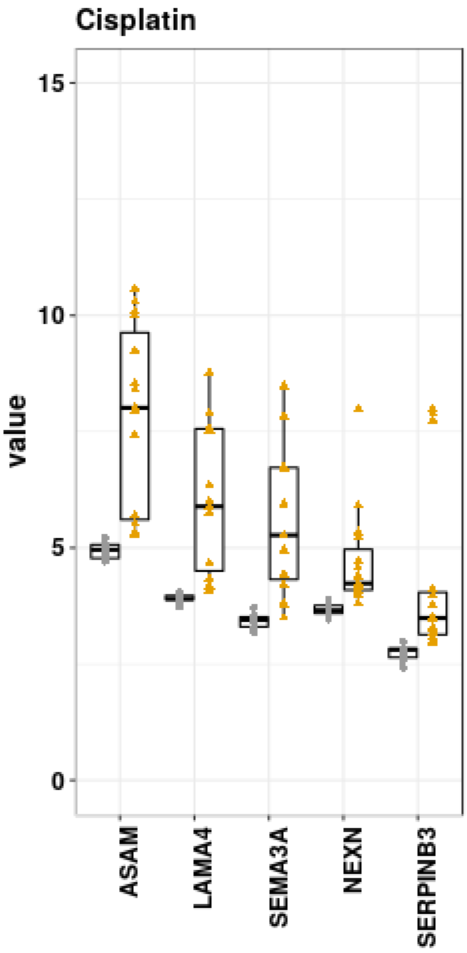
The barplots below show gene expressions ranges separated by drug success. Orange denotes a successful treatment, grey denotes a failed treatment. The Left plot shows the critical features identified by recursive feature elimination, and the right are the top 10 most influential features as identified by our random forest models (per drug).

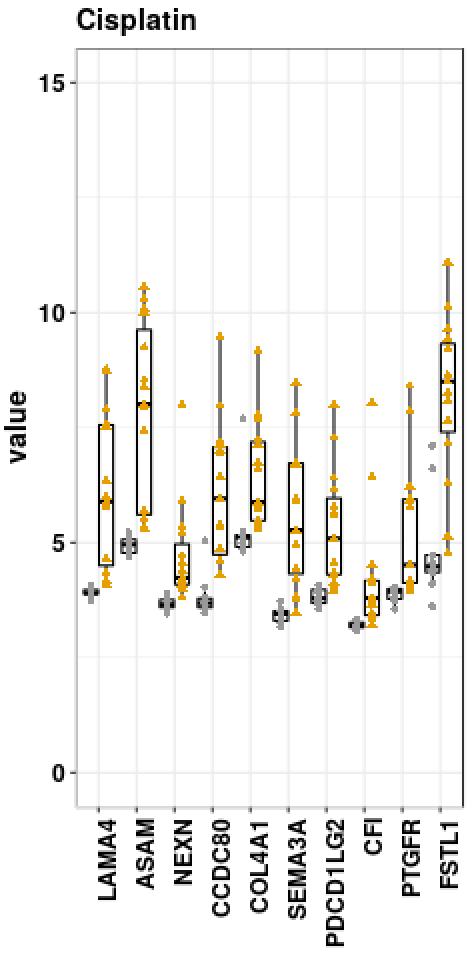


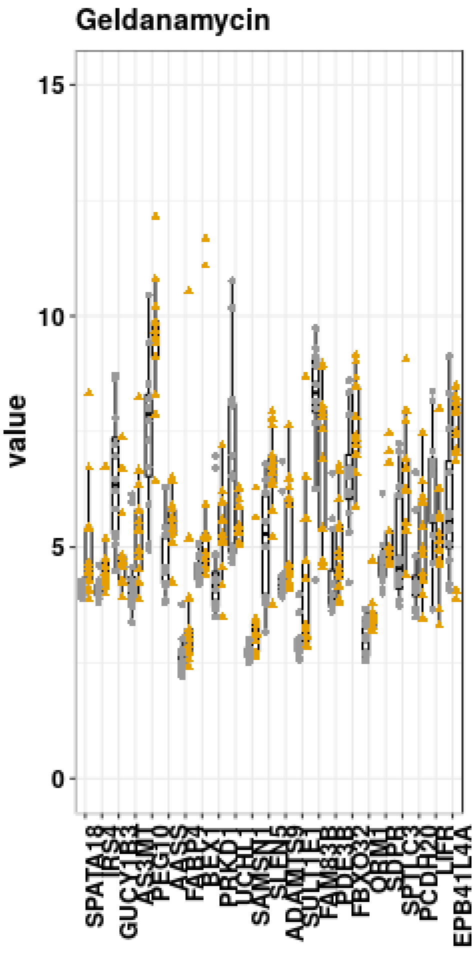


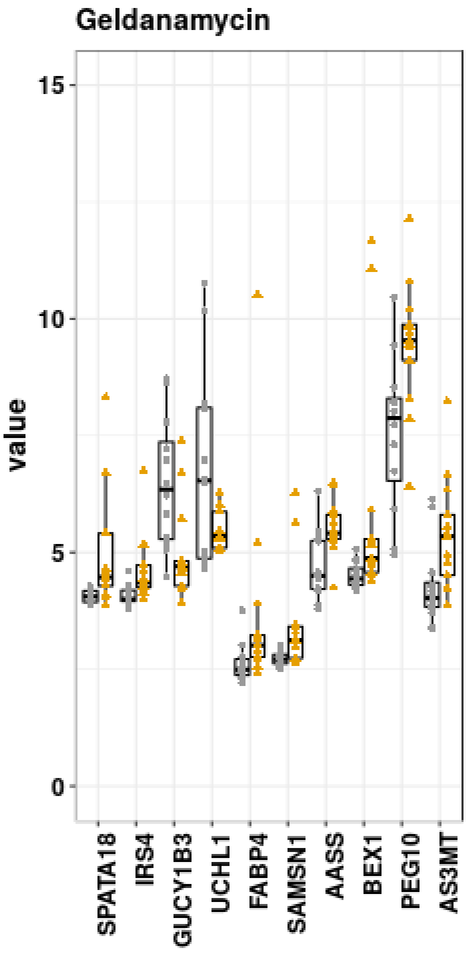

CGC-11047
value
15
10
5
0
TNFAIP6
MMRN2
C1S
FIBIN
KGFLP2
TMEM45A
TMEM158
TNC
MEST
MGP
WDR17
ZNF711
ATP11C
BMP5
PIK3C2G
PI15
FLJ21986
SLC34A2
ABCA8
ANGPTL1
ETV5
PDE1C
IGFBP7

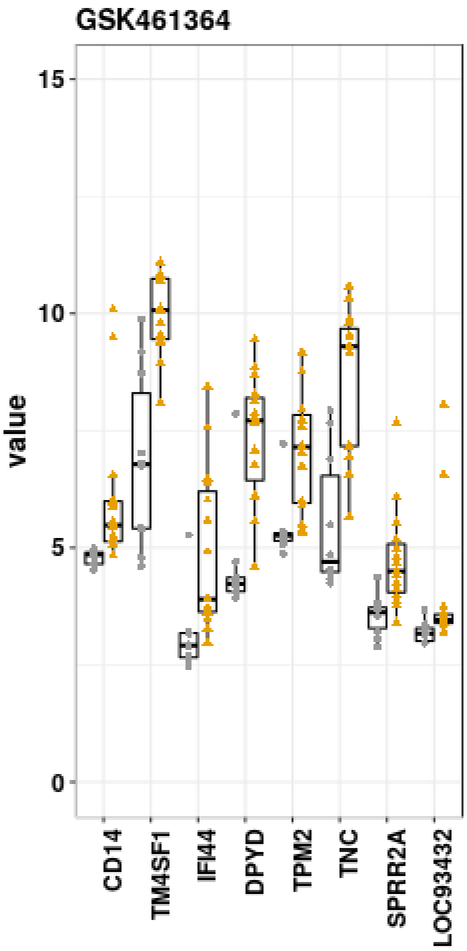

CGC-11047
value
15
10
5
0
TNFAIP6
MGP
FIBIN
KGFLP2
ZNF711
MMRN2
TMEM45A
SLC34A2
CHI3L1
C1S

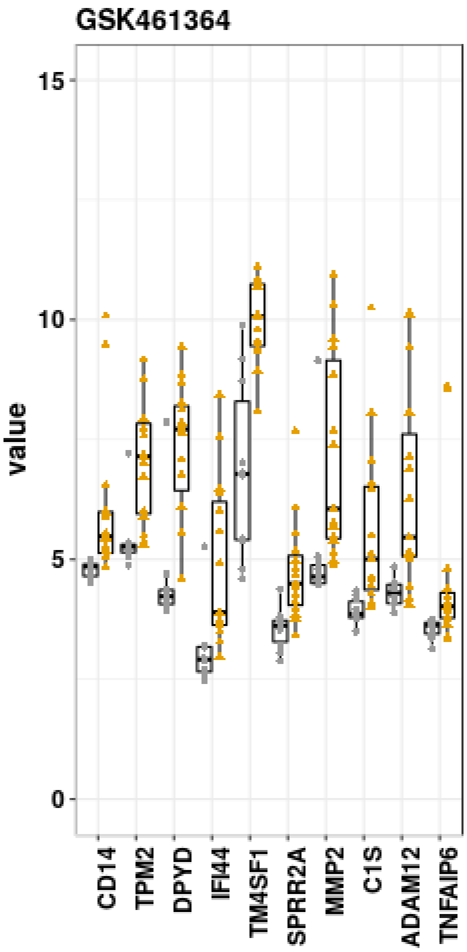

Cisplatin
value
15
10
5
0
ASAM
LAMA4
SEMA3A
NEXN
SERPINB3


Cisplatin
value
15
10
5
0
LAMA4
ASAM
NEXN
CCDC80
COL4A1
SEMA3A
PDCD1LG2
CFI
PTGFR
FSTL1

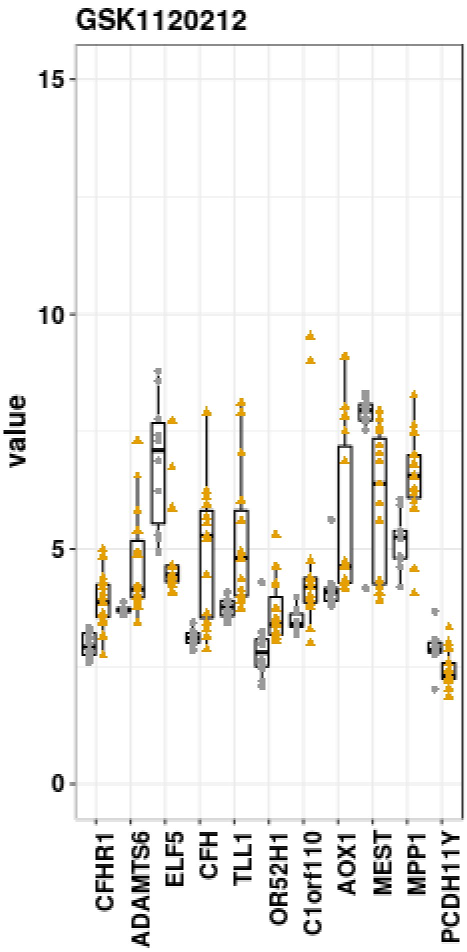

Geldanamycin
value
15
10
5
0
SPATA18
IRS4
GUCY1B3
AS3MT
PEG10
AASS
FABP4
BEX1
PRKD1
UCHL1
SAMSN1
SLFN5
ADAMTS9
SULT1E1
PDE3B
FBXO32
SDPR
SPTLC3
PCDH20
LIFR
EPB41L4A

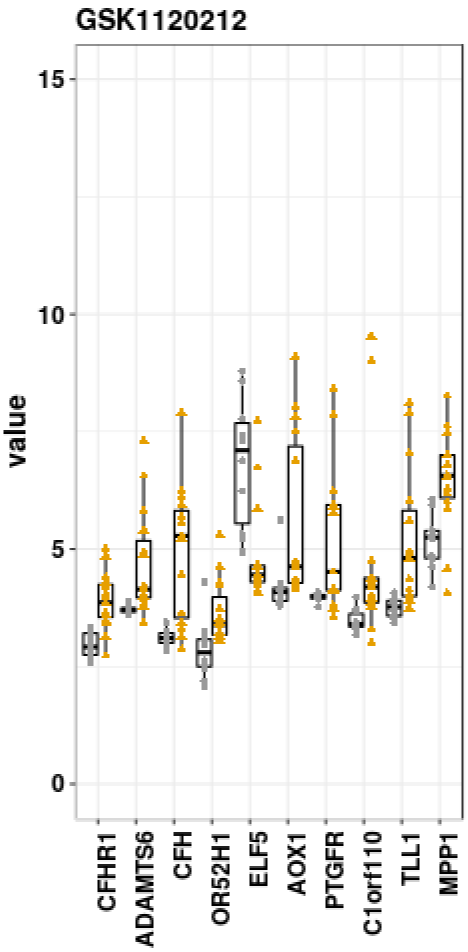

Geldanamycin
value
15
10
5
0
SPATA18
IRS4
GUCY1B3
UCHL1
FABP4
SAMSN1
AASS
BEX1
PEG10
AS3MT

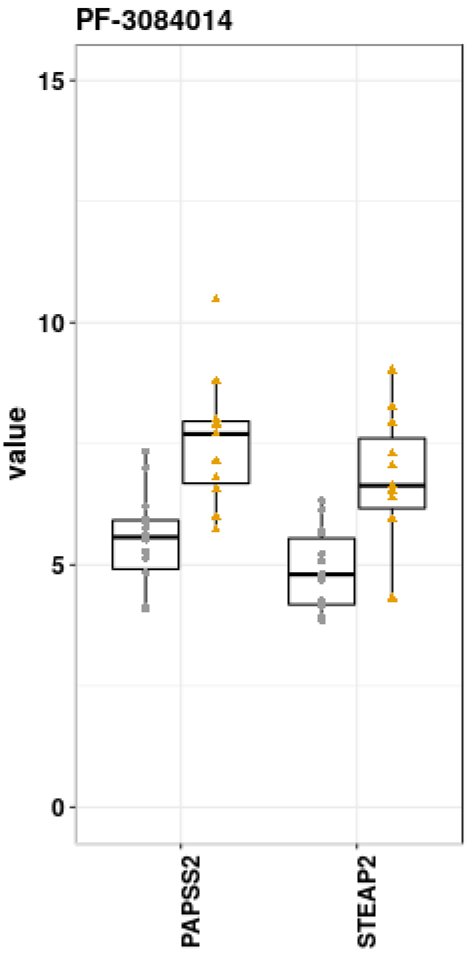

GSK461364
value
15
10
5
0
CD14
TM4SF1
IFI44
DPYD
TPM2
TNC
SPRR2A
LOC93432

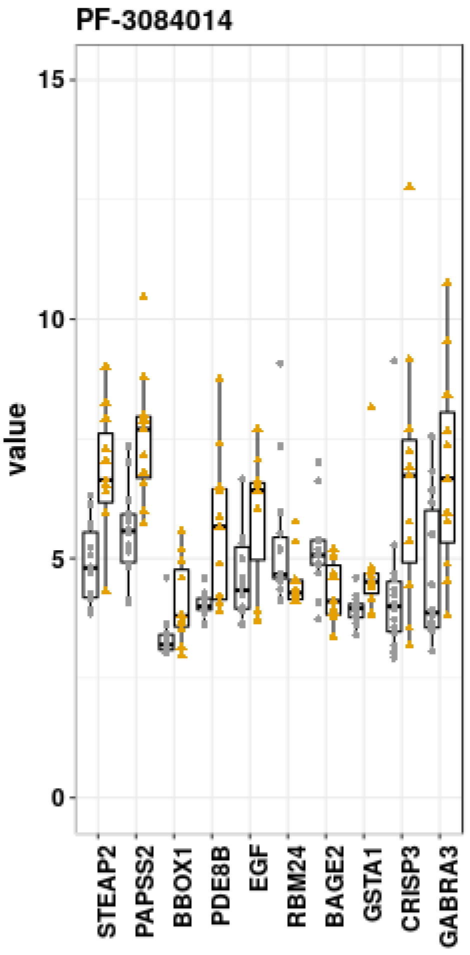

GSK461364
value
15
10
5
0
CD14
TPM2
DPYD
IFI44
TM4SF1
SPRR2A
MMP2
C1S
ADAM12
TNFAIP6

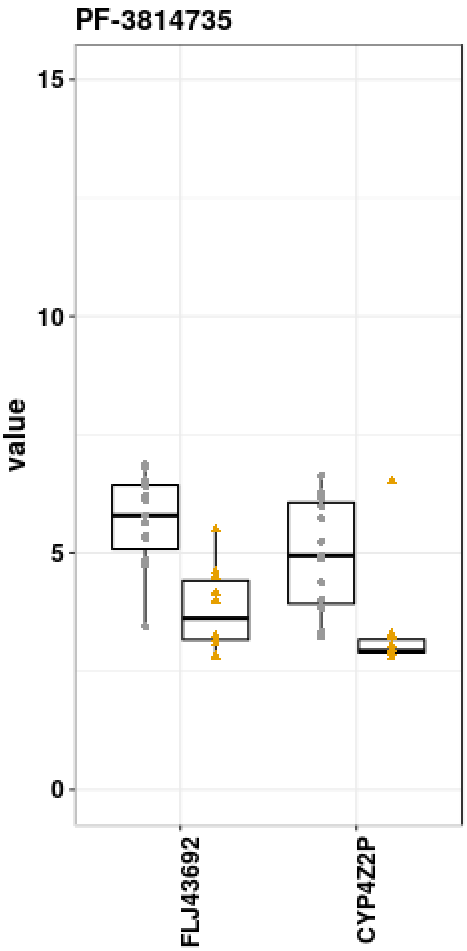
y

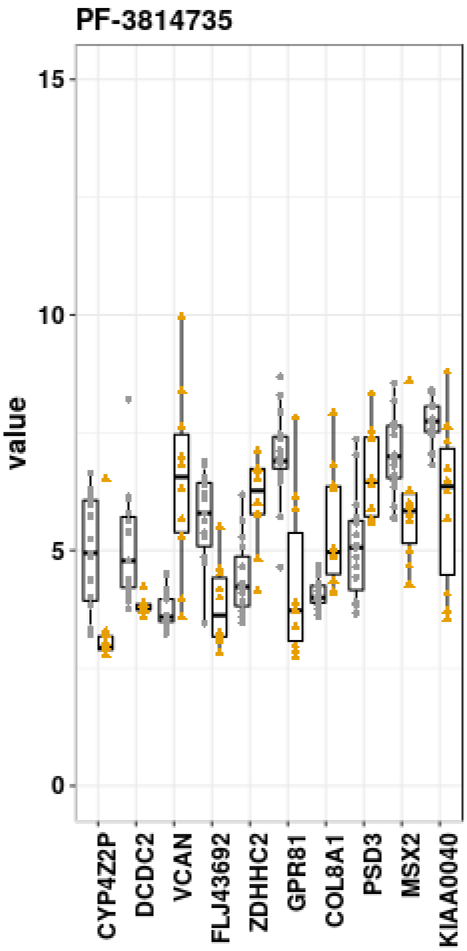

GSK1120212
value
15
10
5
0
CFHR1
ADAMTS6
ELF5
CFH
TLL1
OR52H1
C1orf110
AOX1
MEST
MPP1
PCDH11Y


GSK1120212
value
15
10
5
0
CFHR1
ADAMTS6
CFH
OR52H1
ELF5
AOX1
PTGFR
C1orf110
TLL1
MPP1


PF-3084014
value
15
10
5
0
PAPSS2
STEAP2


PF-3084014
value
15
10
5
0
STEAP2
PAPSS2
BBOX1
PDE8B
EGF
RBM24
BAGE2
GSTA1
CRISP3
GABRA3


PF-3814735
value
15
10
5
0
FLJ43692
CYP4Z2P


PF-3814735
value
15
10
5
0
CYP4Z2P
DCDC2
VCAN
FLJ43692
ZDHHC2
GPR81
COL8A1
PSD3
MSX2
KIAA0040