Matching QSAR Sets

Sphere Algorithms, Fingerprints And Random Forests

Rajarshi Guha

Penn State University

Chemistry Development Kit

- A Java framework for
  - computational chemistry
  - cheminformatics
- Features include
  - Numerous file formats
  - Structure generation and viewing
  - Ring detection and graph theoretical algorithms
  - IUPAC name generation and SMILE parsing
  - Online database access
- The useful feature for the current study was fingerprint generation

Molecular Fingerprints

- Abstraction of *structural keys*
- Fingerprints are represented as bitstrings
- The process of fingerprinting involves
  - Looking for various structural features (patterns) exhaustively
  - Each pattern is hashed and the bits of the hash are OR'ed with the fingerprint
  - The CDK implements a form of Daylight fingerprints (without folding)

Properties of Fringerprints

- Fingperprints are characteristic to a given pattern
- If B is a substructure of A then bits set in the fingerprint of B will be set in that of A
- However, fingerprints can be certain about the absence of a feature, but **not** about the presence of one.
- Due to the non specifity of the bits wrt structural features fingerprints can contain much more information

Tversky Similarity

- Similarity is some function of the common features shared between A & B and the features unique to A and to B
- The similarity function is given by

$$S_{\mathbf{tversky}} = \frac{c}{\alpha a + \beta b + c}$$

- This can be used for two questions
  - How similar are A & B to each other?
  - How similar is B to A?

Tversky Similarity

- Our interest is in the first question and so $\alpha = \beta$
- If $\alpha = \beta = 0.5$ it reduces to the Dice index and if $\alpha = \beta = 1.0$ it reduces to the Tanimoto index
- If $\alpha, \beta > 1$ then more stress is given to distinguishing features than common features
- $\alpha = \beta = 0.5$ was used (as it allows for variable length descriptor vectors)

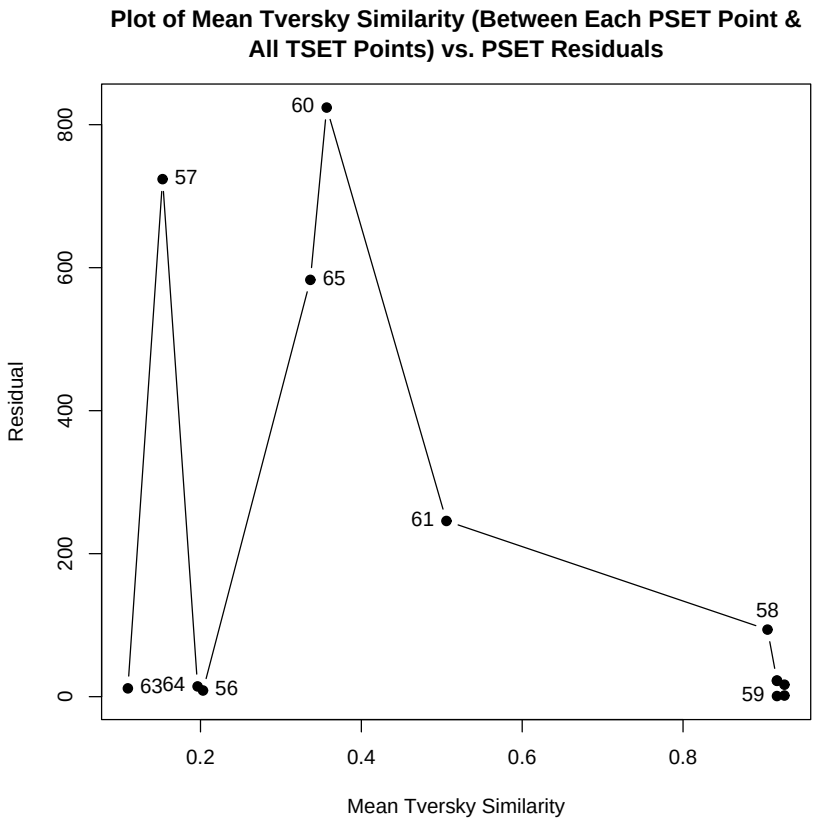
Using Fingerprints & Tversky Similarity

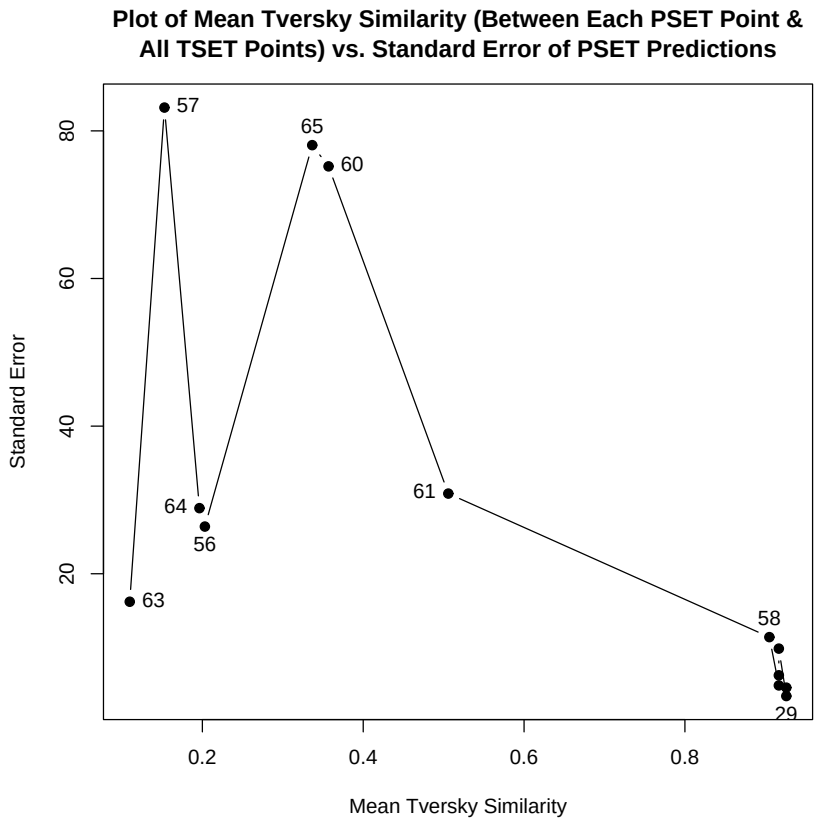
- The initial methodology involved:
  - Evaluate fingerprints of all the molecules in the dataset
  - Evaluate a similarity matrix for all the molecules
  - Look at the average similarity for each PSET molecule to all the TSET molecules
  - Use the best linear model for the dataset to make plots of average similarity versus residuals and standard errors of prediction

Toy Dataset - Review

- 65 molecules, 52 in TSET, 13 in PSET
- 8 PSET molecules were *very* dissimilar
- Dependent variable was BP
- 4 descriptor linear model: EMIN, EMAX, ECCN, SHDW
- R^2 = .91, F = 119.9 on (4,47) DF and $p < 2.2 \times 10^{-16}$

Toy Dataset - Direct Averaging of Similarities





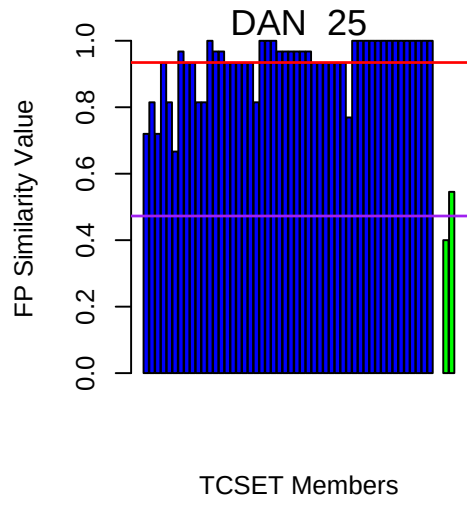
- The average is calculated between each PSET point and ***all*** TSET points in the dataset

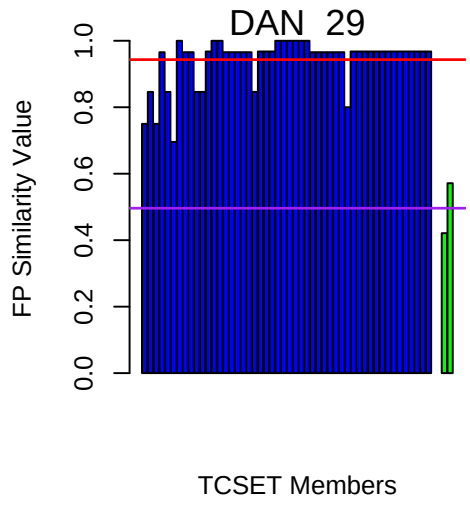
Toy Dataset - Comment

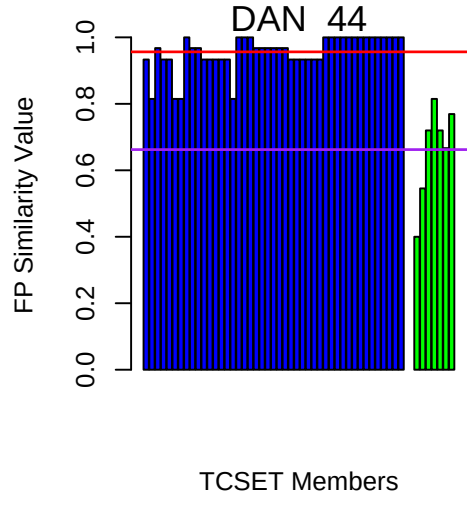
- Taking an average should give us a broad view of similarity of PSET points to the TSET
- The plots indicate a *general* trend of higher similarity correlated to lower residuals/SE
- However, the trend is spoiled by a number of *outliers*
- Maybe ignore the TSET molecules that might be skewing the average?
- Alternatives:
  - Exlcude PSET points with 0 density
  - kNN - replace Euclidean distance with FP similarity

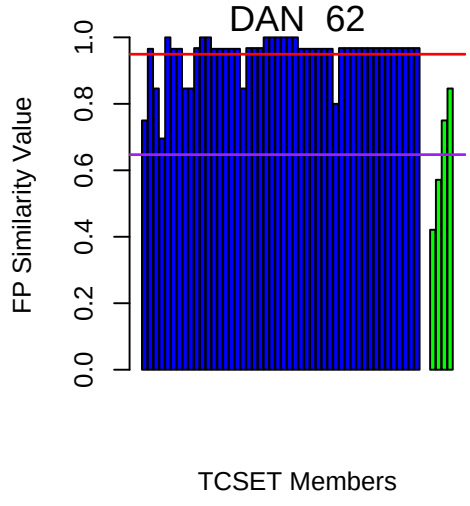
Toy Dataset/Sphere - FPS Distributions

Barplots of FP Similarity Values Between High Density PSET Points And TSET Members Inside And Outside Their Sphere



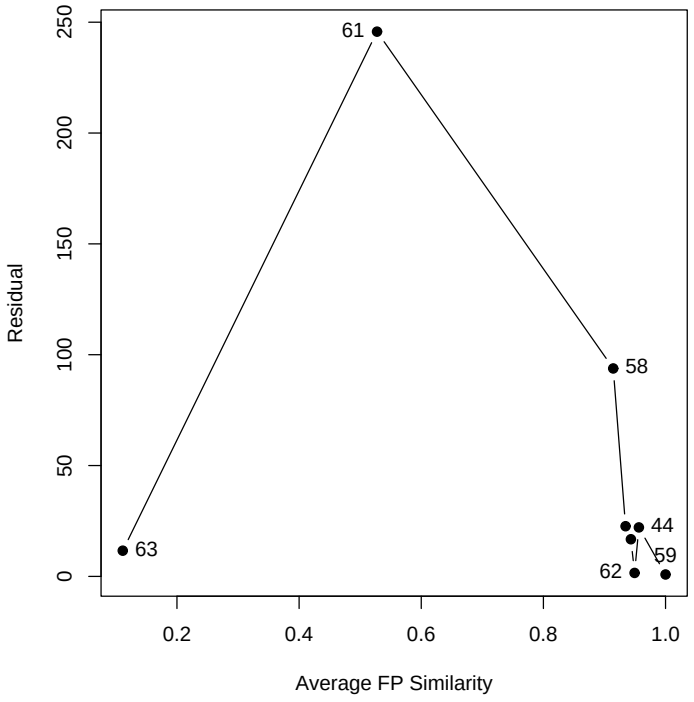




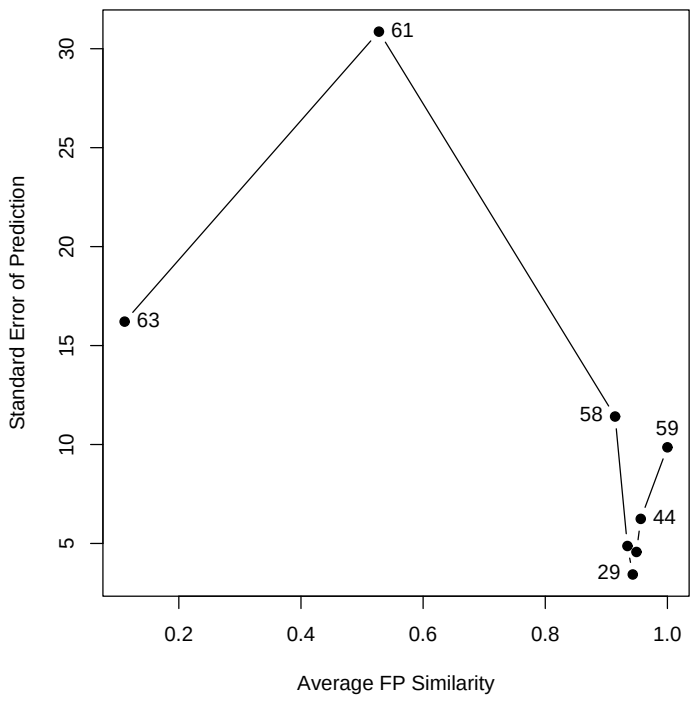


Toy Dataset/Sphere - Residuals & SEoP

Plot of Average FP Similarity for Each PSET Sphere
vs. Residuals of Prediction for PSET Points



Plot of Average FP Similarity for Each PSET Sphere
vs. Standard Error of Prediction for PSET Points

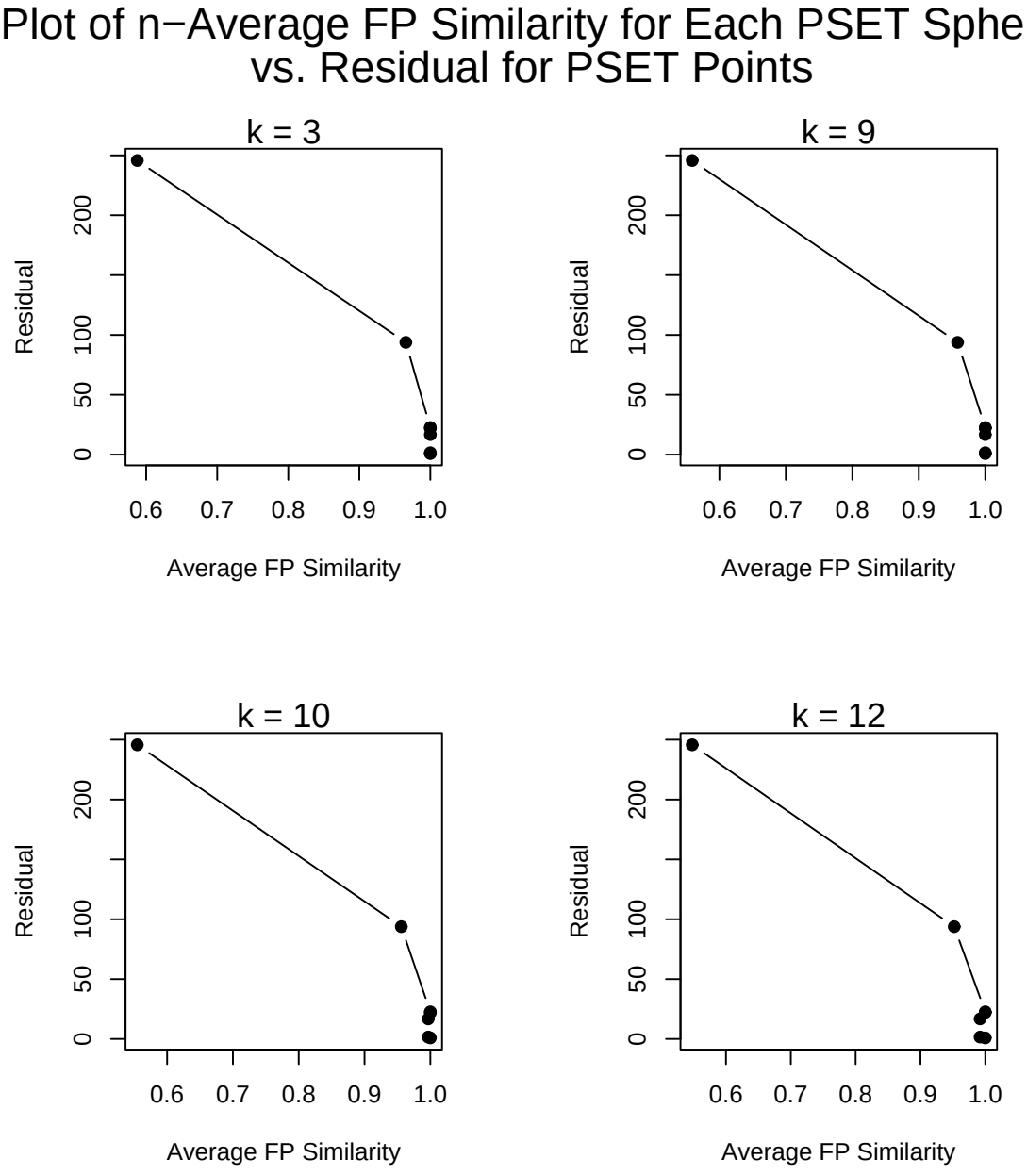


- PSET points with 0 density are excluded
- DAN 63 (pyrrole) is anomalous for the residual plot
- DAN 63 & 59 (decane) are anomalous for the SEoP plot
- Decane can be justifi ed somewhat but pyrrole is weird!

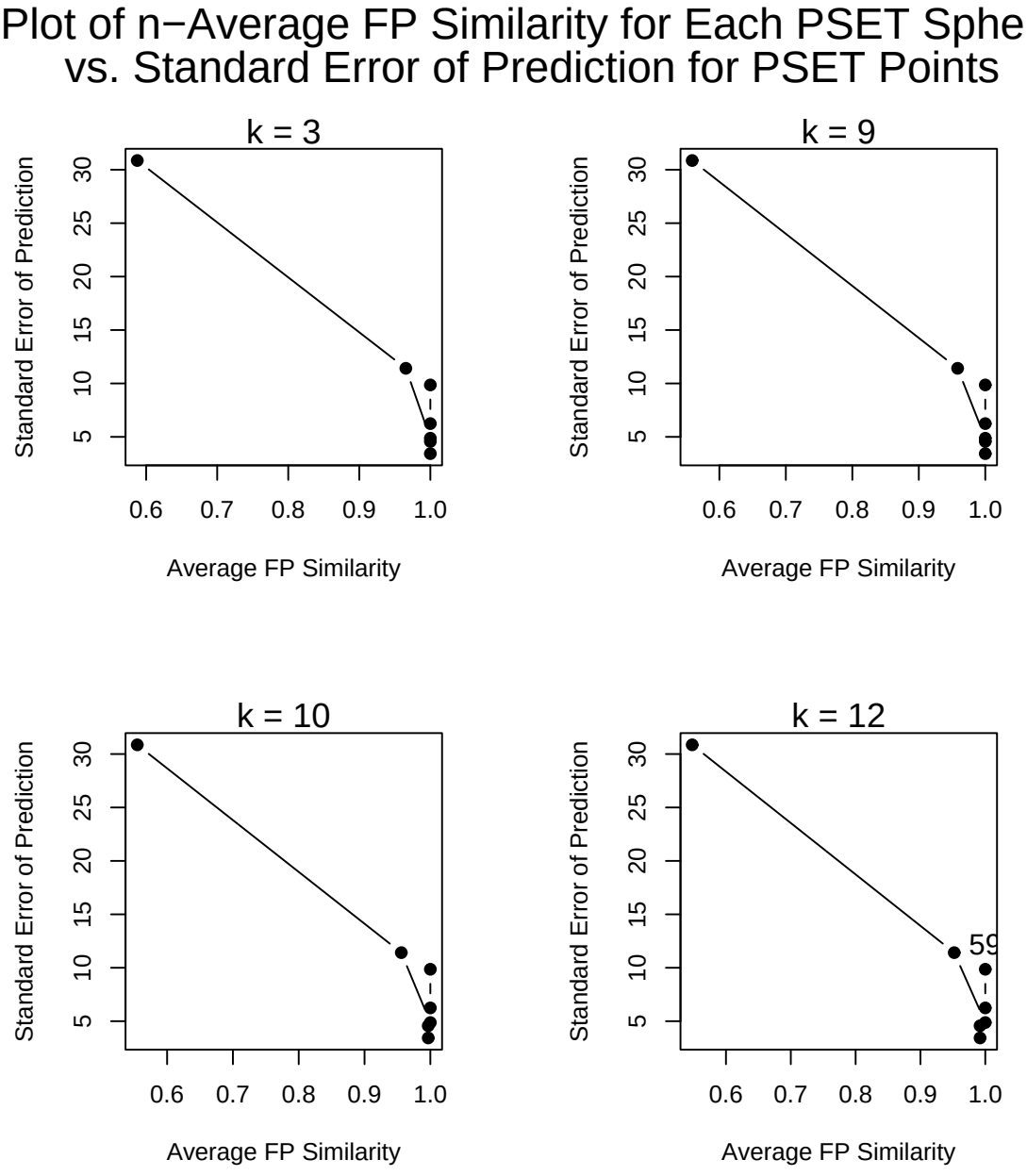
Toy Data/Sphere - kNN Modification

- Rather than consider all points in the PSET sphere take top k points and use the average similarity
- Here top would imply most similar, i.e., those points with highest FPS value
- In most cases the top 3 such points will have a FPS = 1.0, which does not allow for much discrimination
- So we increase k and see what happens
- (In all the plots pyrrole is excluded)

Toy Data/Sphere - kNN Modification



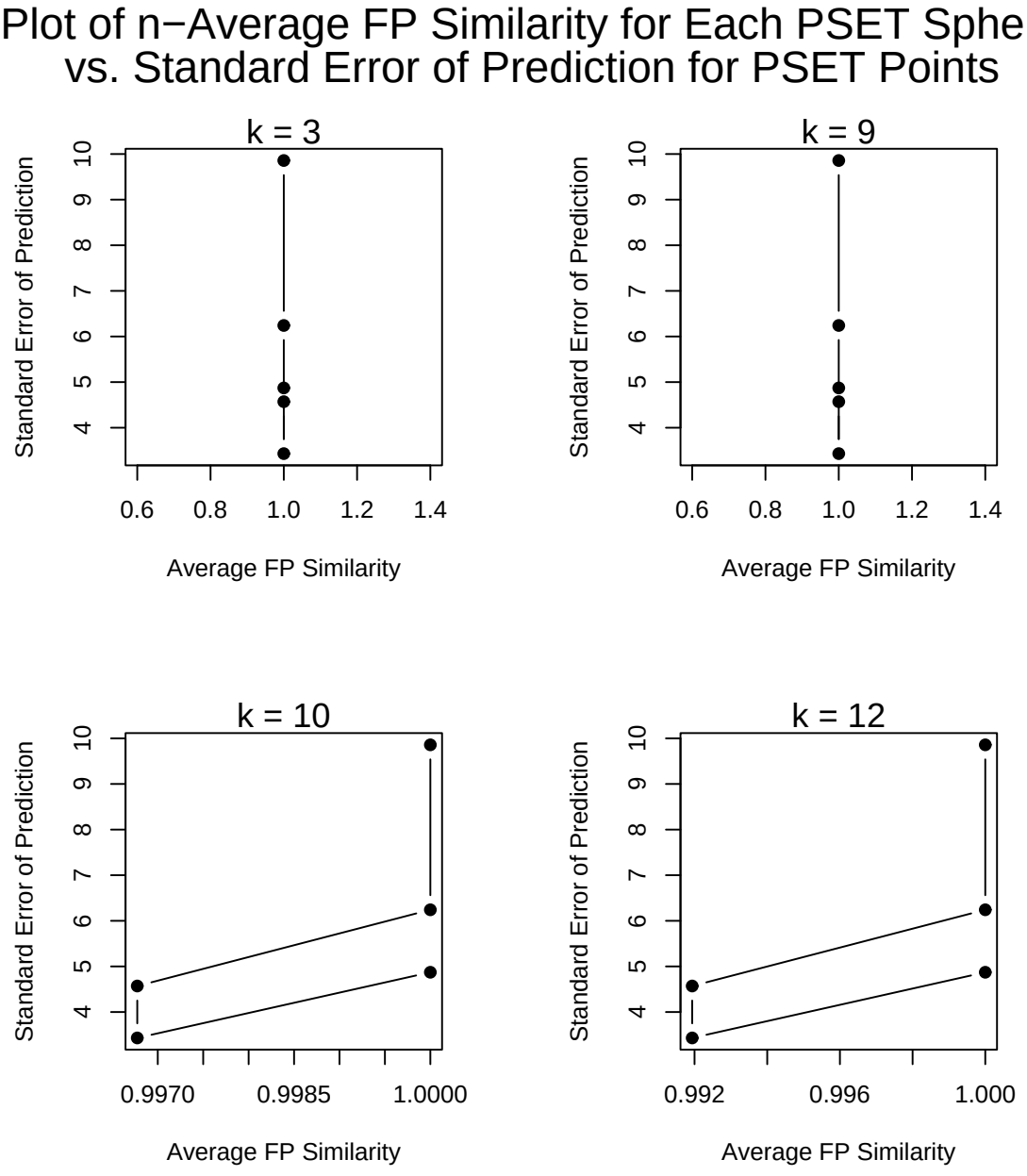
Toy Data/Sphere - kNN Modification



Toy Dataset/Sphere - kNN Modification

- Apparently this idea doesn't improve matters
- Not much difference with different k values
- There does appear to be a small region which is affected by values of k.
- However, focussing on that region doesn't improve matters much

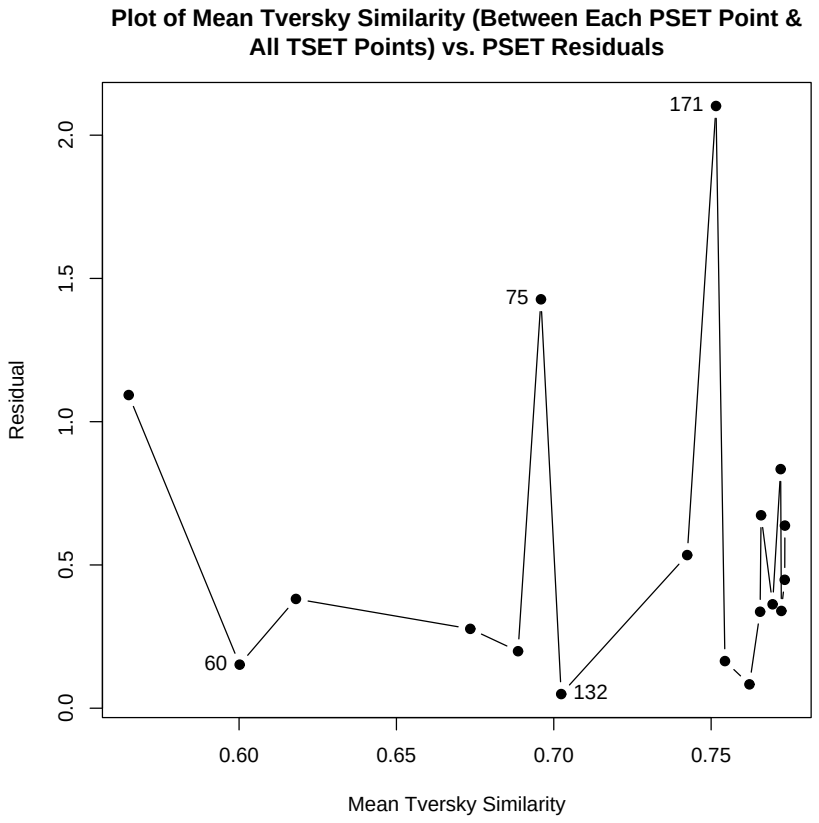
Toy Data/Sphere - kNN Modification

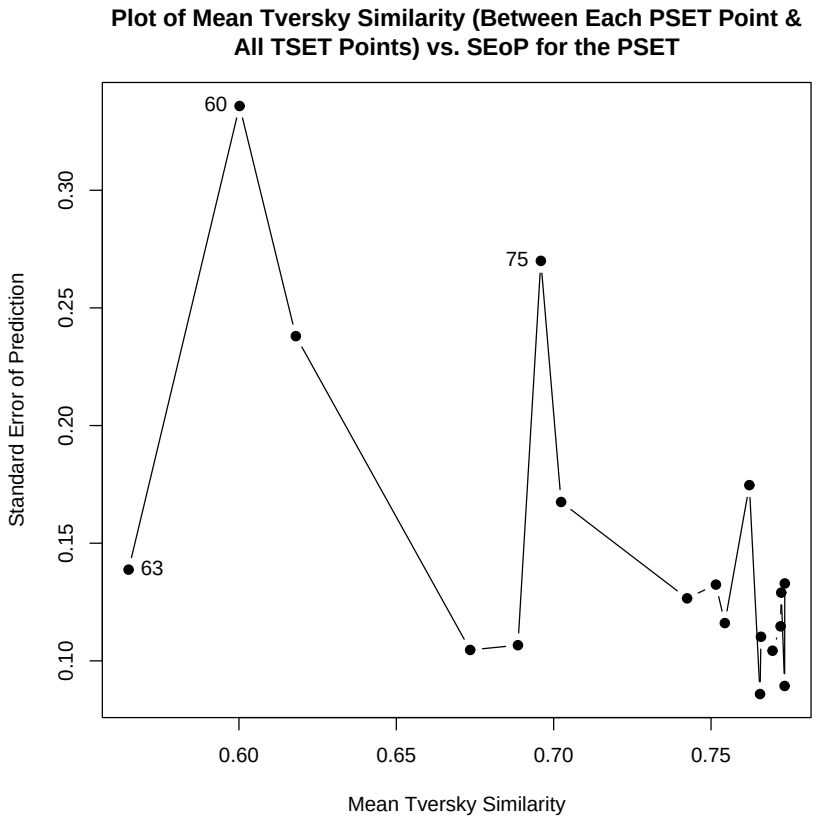


Artemisinin

- 179 molecules, 161 in TSET, 18 in PSET
- Dependent variable is log RA
- 4 descriptor linear model: N7CH-20, NSB, WTPT-2, MDE-14
- $R^2 = 0.71$, $F = 95.28$ on (4,156) DF and $p < 2.2 \times 10^{-16}$

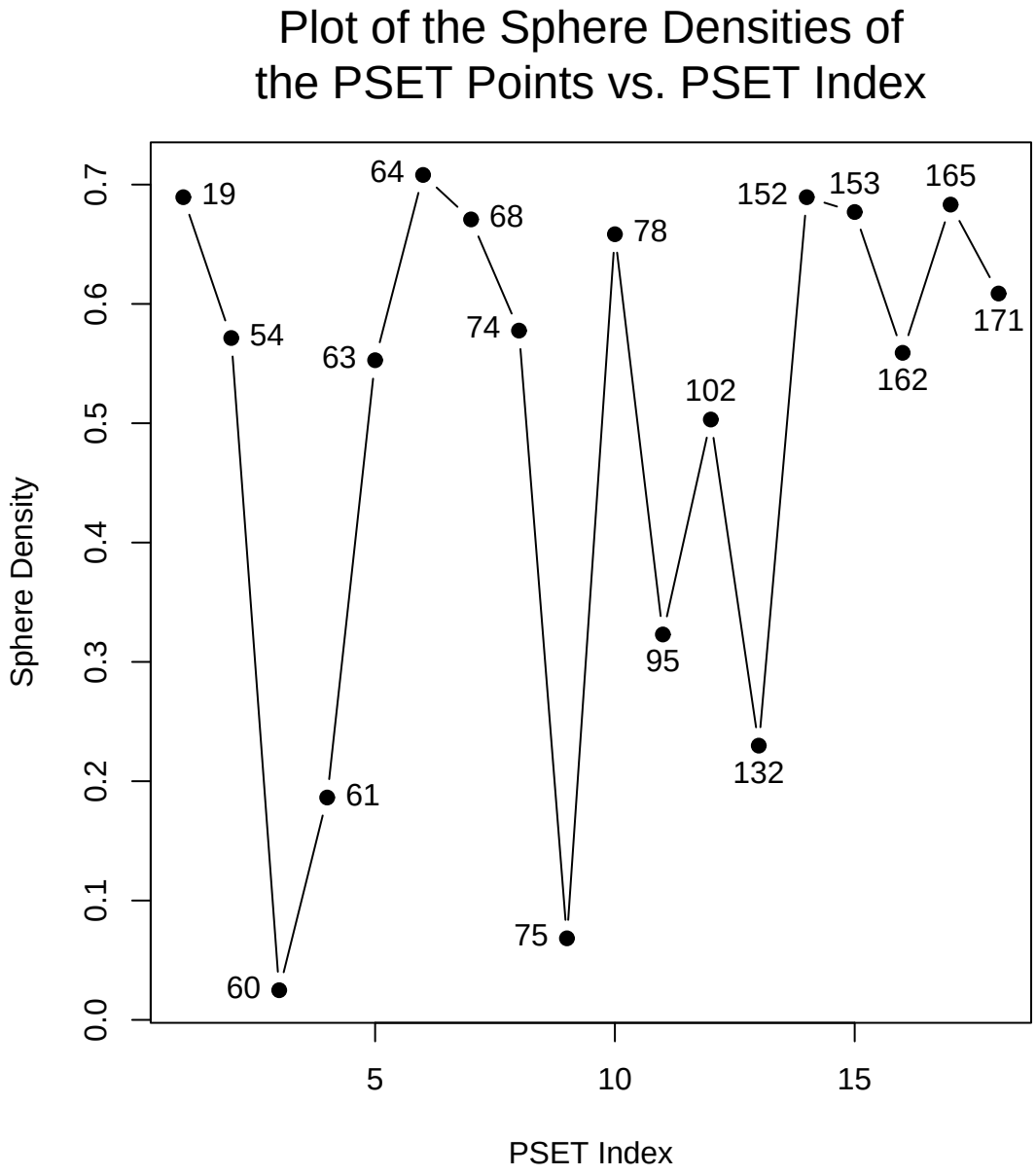
Artemisinin - Direct Averaging of Similarities





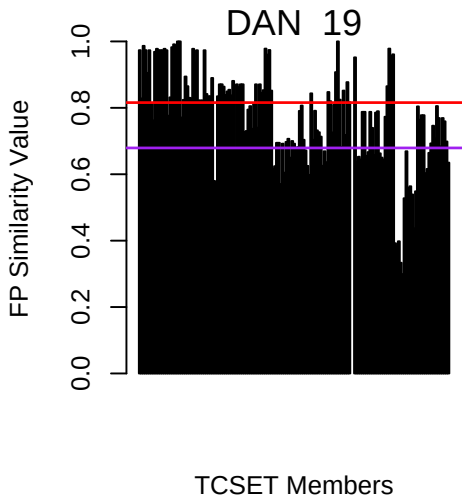
- The average is calculated between each PSET point and ***all*** TSET points in the dataset

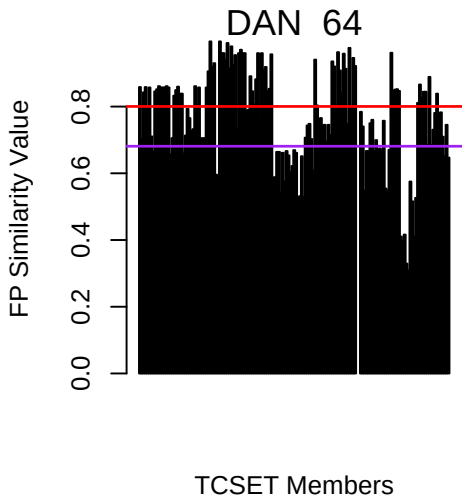
Artemisinin / Sphere - Density Plot

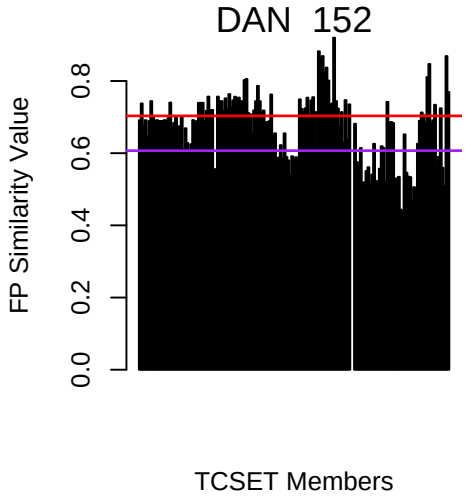


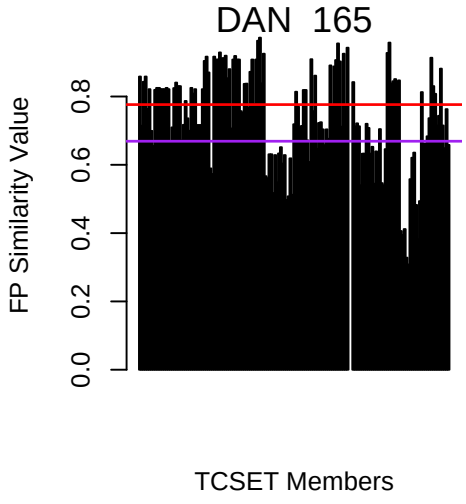
Artemisinin / Sphere - FPS Distributions

Barplots of FP Similarity Values Between High Density PSET Points And TSET Members Inside And Outside Their Sphere

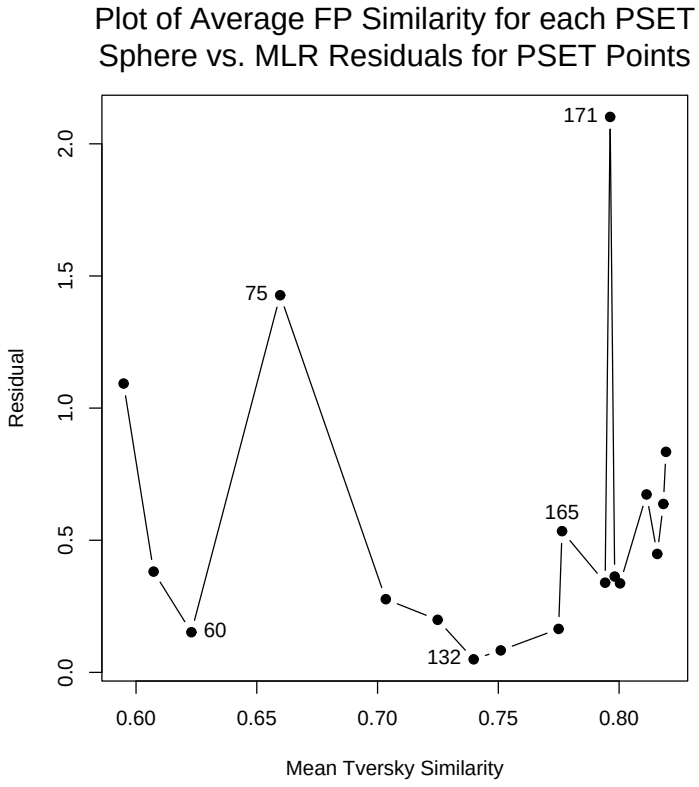


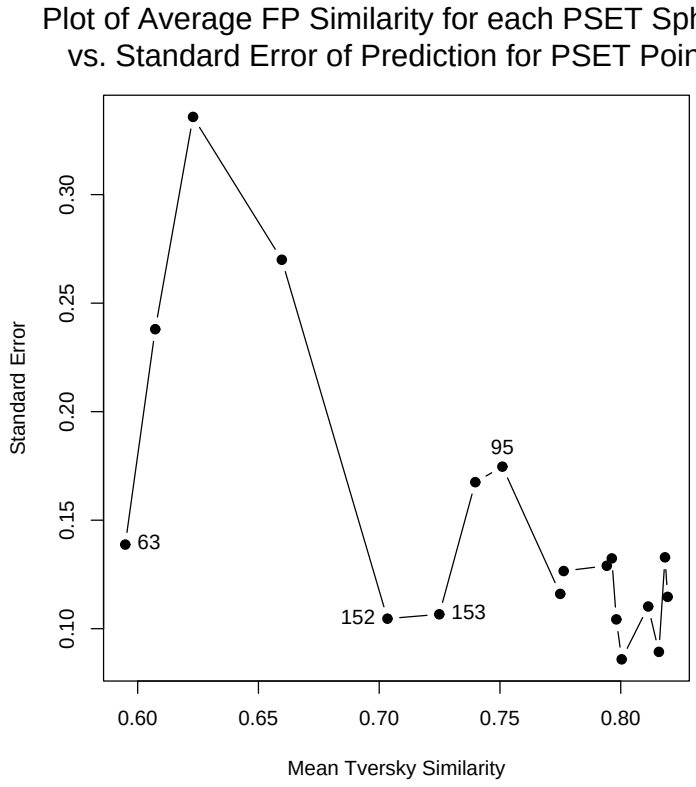




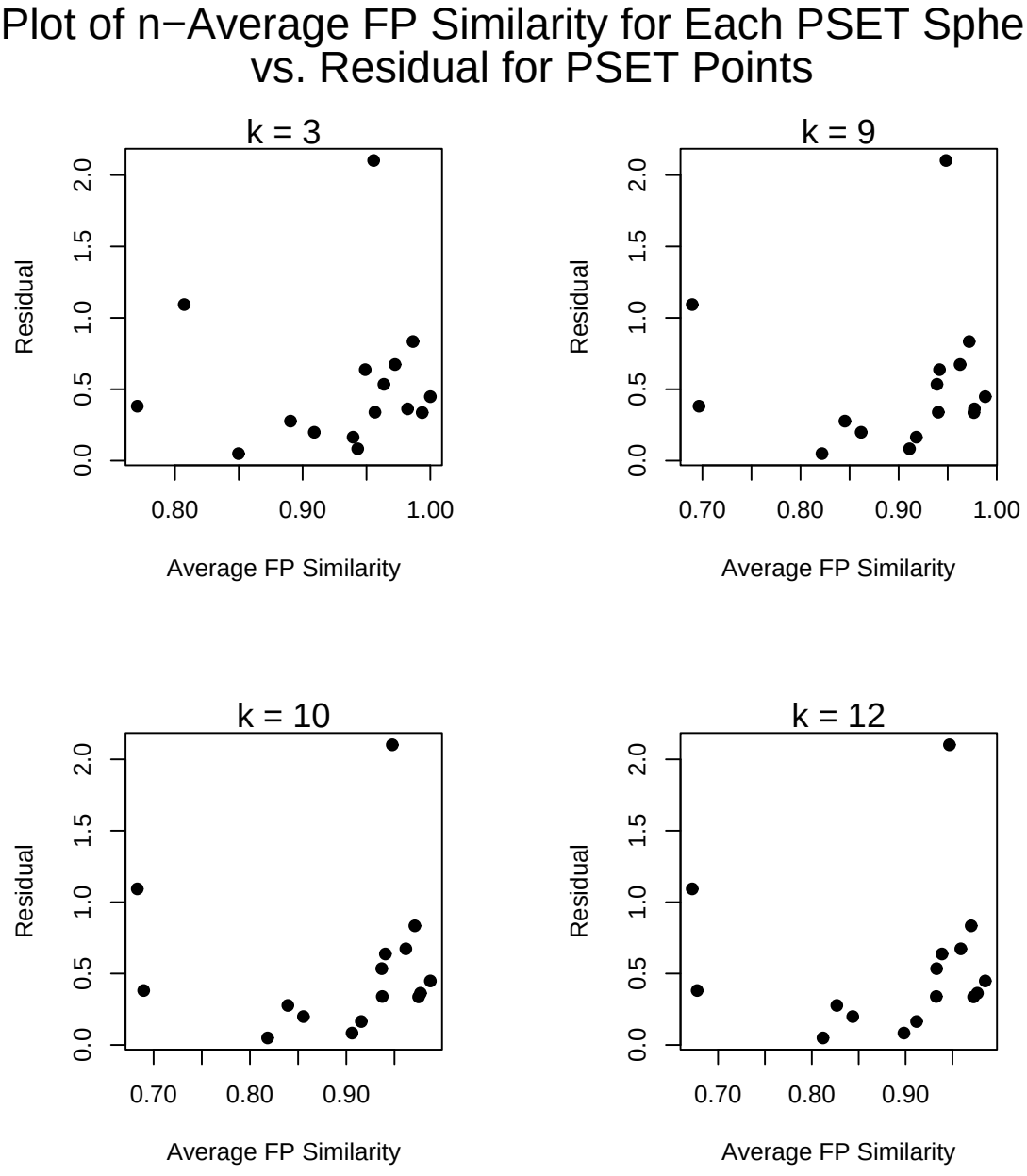


Artemisinin / Sphere - Residuals & SEoP

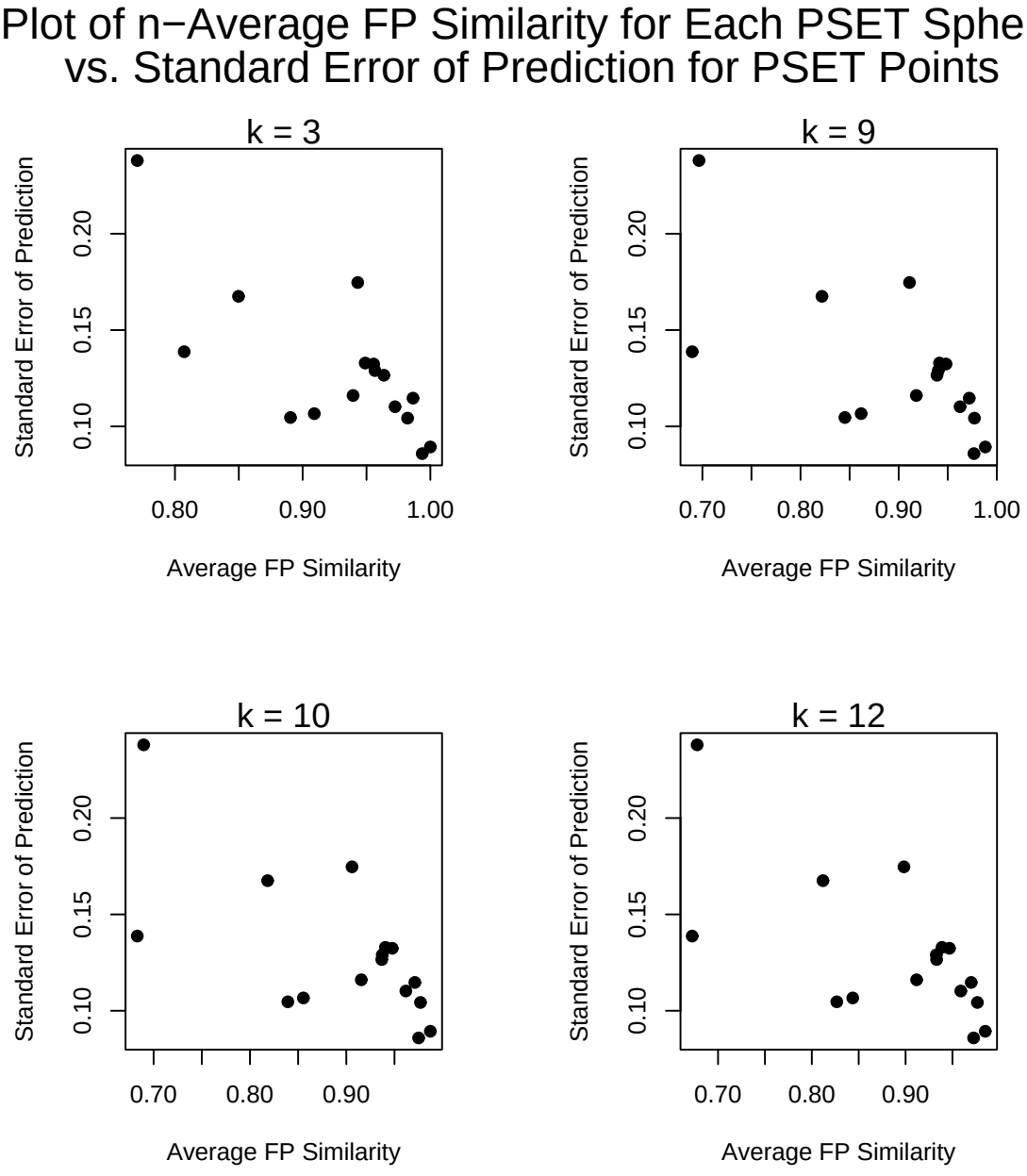




Artemisinin/Sphere - kNN Modification



Artemisinin/Sphere - kNN Modification



Artemisinin / Sphere - Comments

- The results using fingerprints are quite similar to those using atom pairs
- Once again, standard error of predictions seem to correlate better than residuals
- The kNN results indicate *some* trend. But its not very distinct. As with the toy dataset varying k doesn't seem to affect the result much

Random Forests

Classifying Predictions

- The aim is to predict whether an unknown molecule will have a *good* or *bad* prediction
- Strategy
  - Assign the TSET residuals to two classes: *good* or *bad*
  - Build a classification model with these assignments
  - Use the model to predict the class of the unknown molecules' residuals
  - See if/how well they match!

Classifying Predictions

- How do we decide *good* and *bad*?
  - regression diagnoistics
  - studentized residuals
  - standardized residuals (SR)
- How do we classify?
  - LDA
  - logistic regression
  - SVM
  - Random Forests (RF)

Classifying Predictions

- Regression diagnostics and studentized residuals generally require $\hat{X}$ or some model feature
- This prevents us from calculating them for PSET molecules
- SVM's, LDA and LR require us to do some form of variable selection before they can be used
- Standardized residuals & RF are nice -
  - SR's can be calculated for TSET and PSET
  - RF's do not require variable selection and are resistant to redundant descriptors

Assigning Classes

- Arbitrary assignments
- Two classes are created.
- A split value, s, is specified by the user.

$$\begin{aligned} \mathbf{SR} &\geqslant s \rightarrow \textit{bad} \\ \mathbf{SR} &< s \rightarrow \textit{good} \end{aligned}$$

Random Forest

- The default parameters for the RF were used
- Number of trees = 500
- Number of descriptors used at each split = $\sqrt{N_{desc}}$
- The number of descriptors submitted was varied:
  - entire descriptor pool
  - reduced pool
  - model descriptors

Classifier: Datasets

- Tutorial
  - 277 molecules. 30 descriptors
  - TSET: 235 PSET: 42
- Toy
  - 65 molecules
  - TSET: PSET:
- Artemisinin
  - 179 molecules, 65 descriptors
  - TSET: 161 PSET: 18

Classification Results - Tutorial

- A split value of 1.0 was chosen.
- This gave 51 bad molecules and 184 good molecules in the TSET
- OOB estimate of error rate = 13.62%

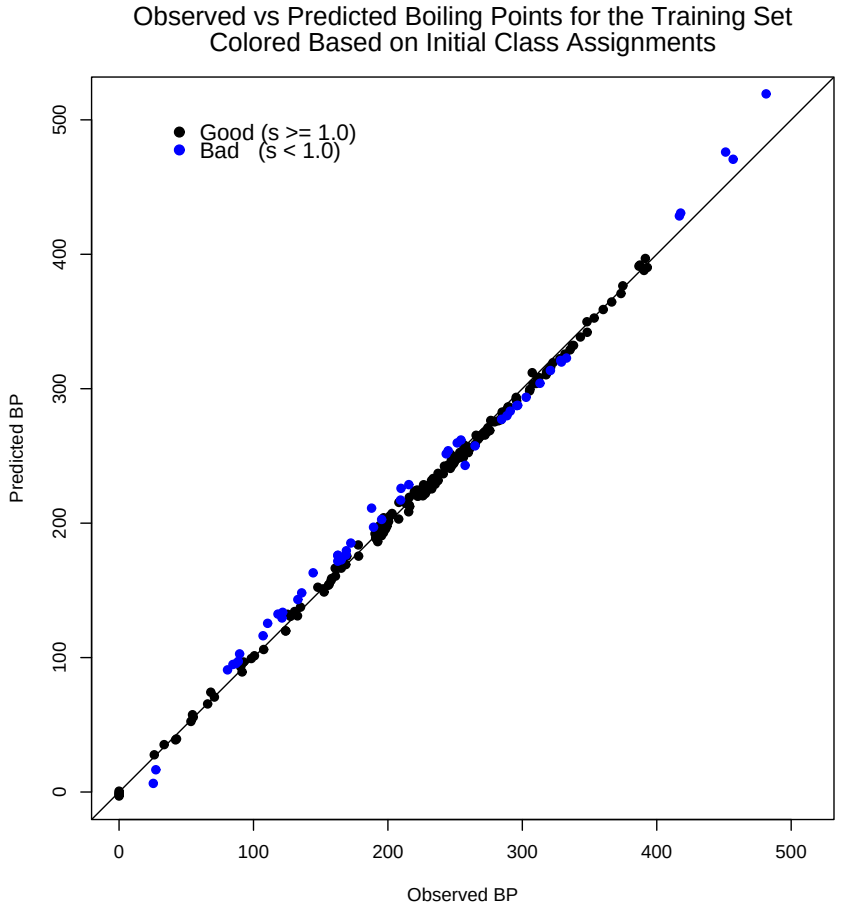
TSET Confusion Matrix

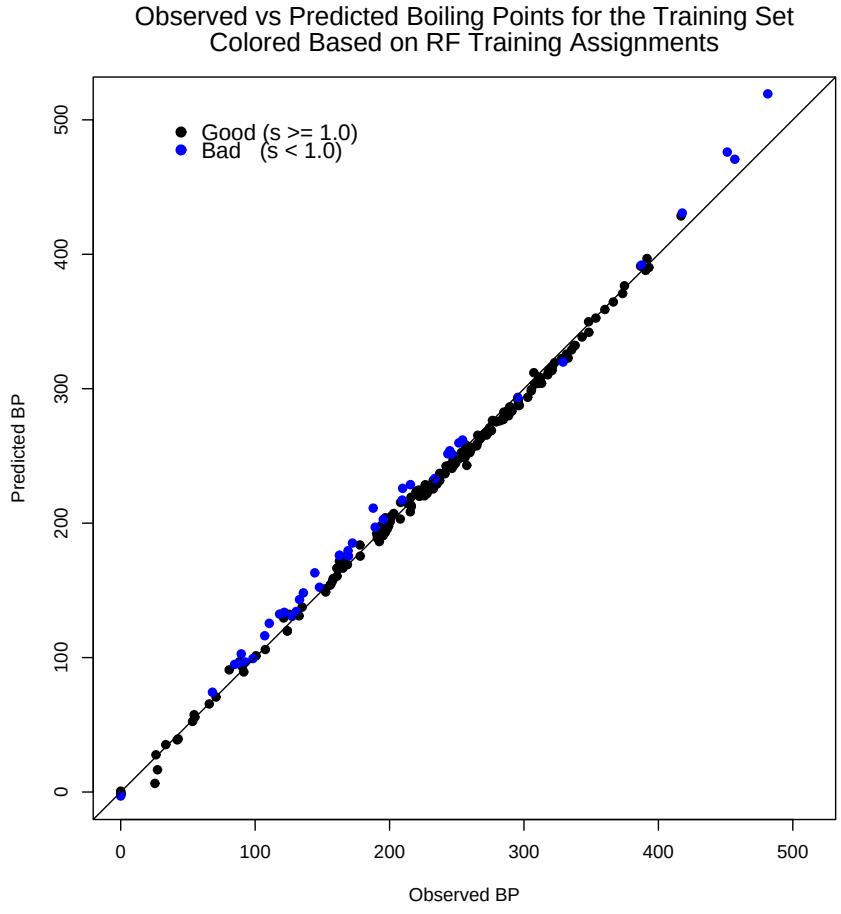
	b	g	✓
b	32	19	62%
g	13	171	92%

PSET Confusion Matrix

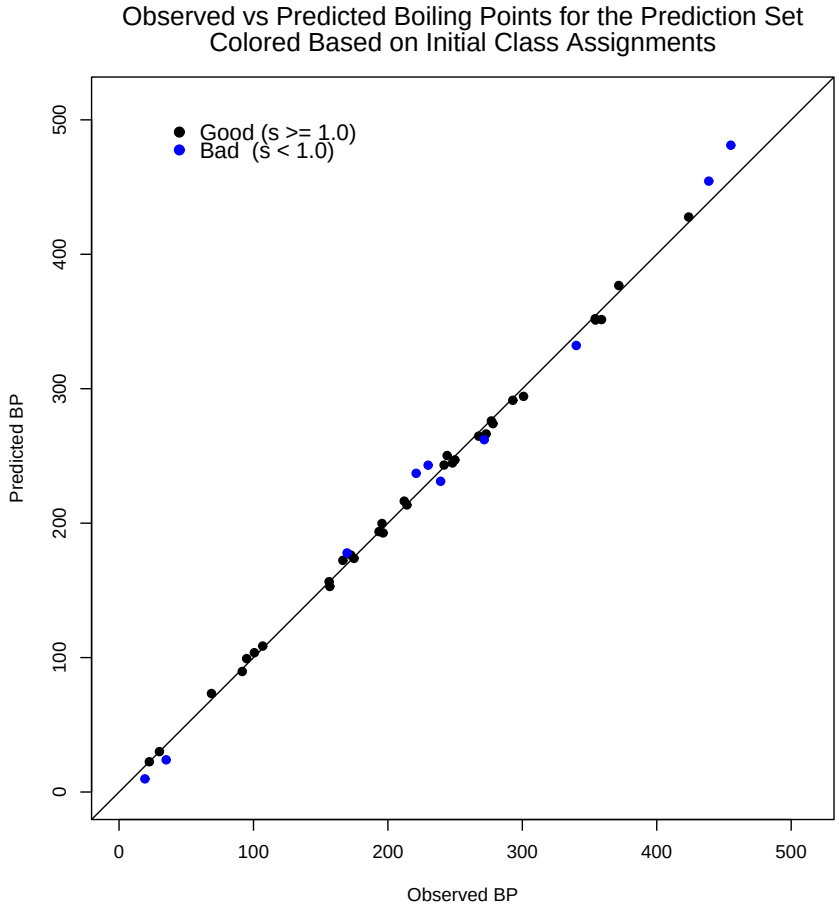
	b	g	✓
b	5	5	50%
g	3	29	90%

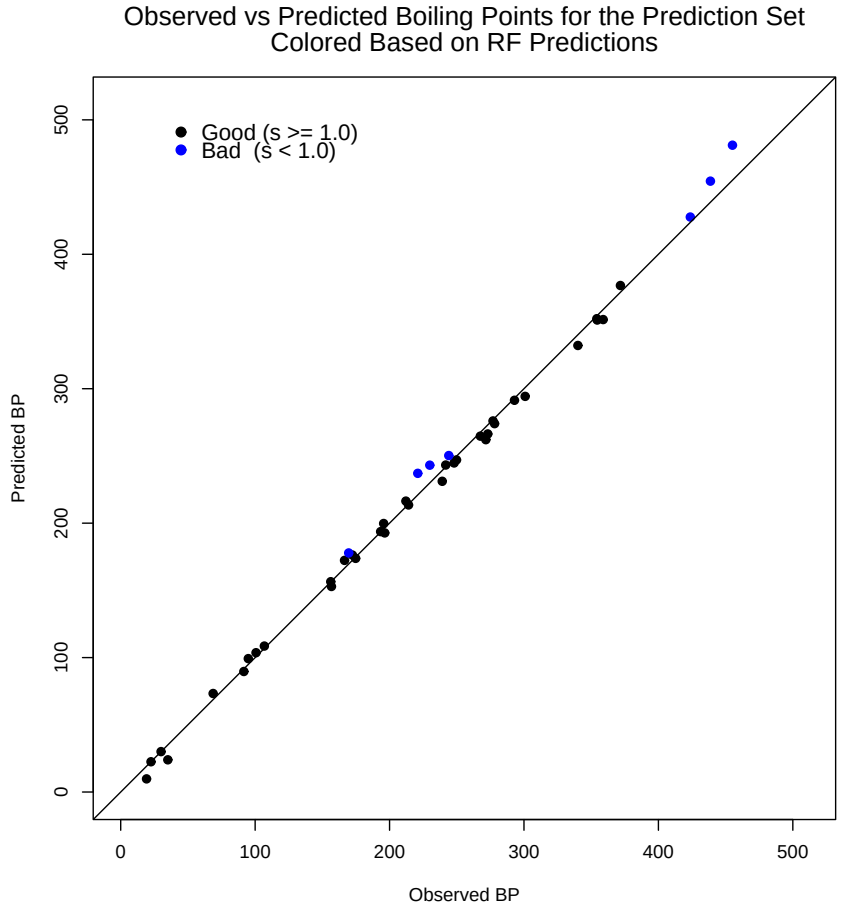
Classification Results - Tutorial (TSET)





Classification Results - Tutorial (PSET)

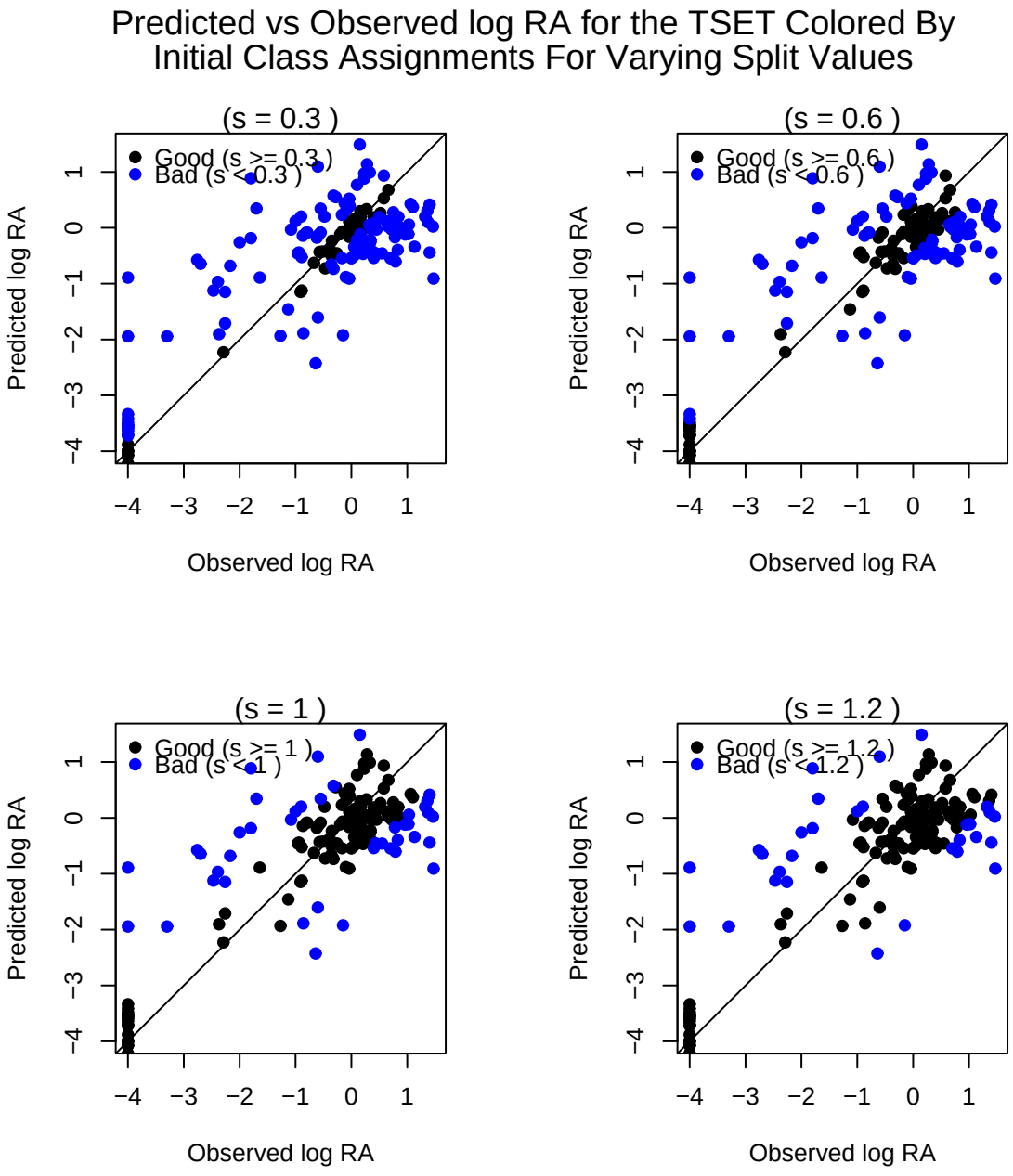




Classification Results - Tutorial

- On using model descriptors only RF training improves slightly, predictions are a little poorer
- In general the *bad* class has higher error, justified due to the small size of the class

Classification Results - Artemisinin



Classification Results - Artemisinin

- A split value of 0.85 was chosen.
- This gave 50 bad molecules and 111 good molecules in the TSET
- OOB estimate of error rate = 34.16%

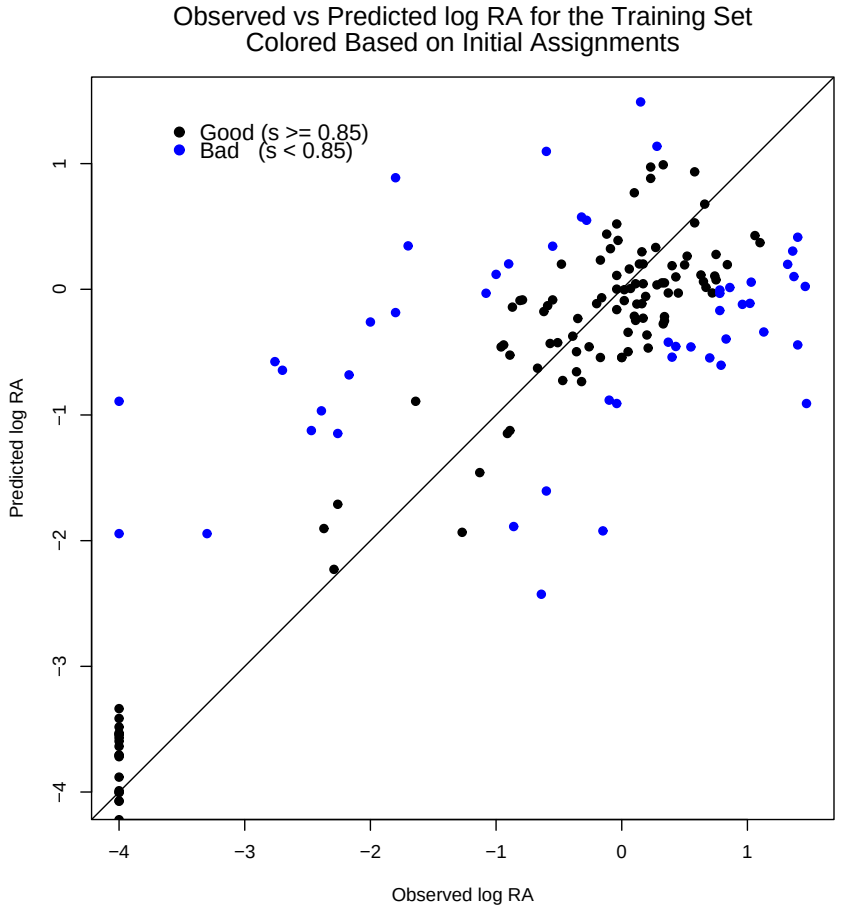
TSET Confusion Matrix

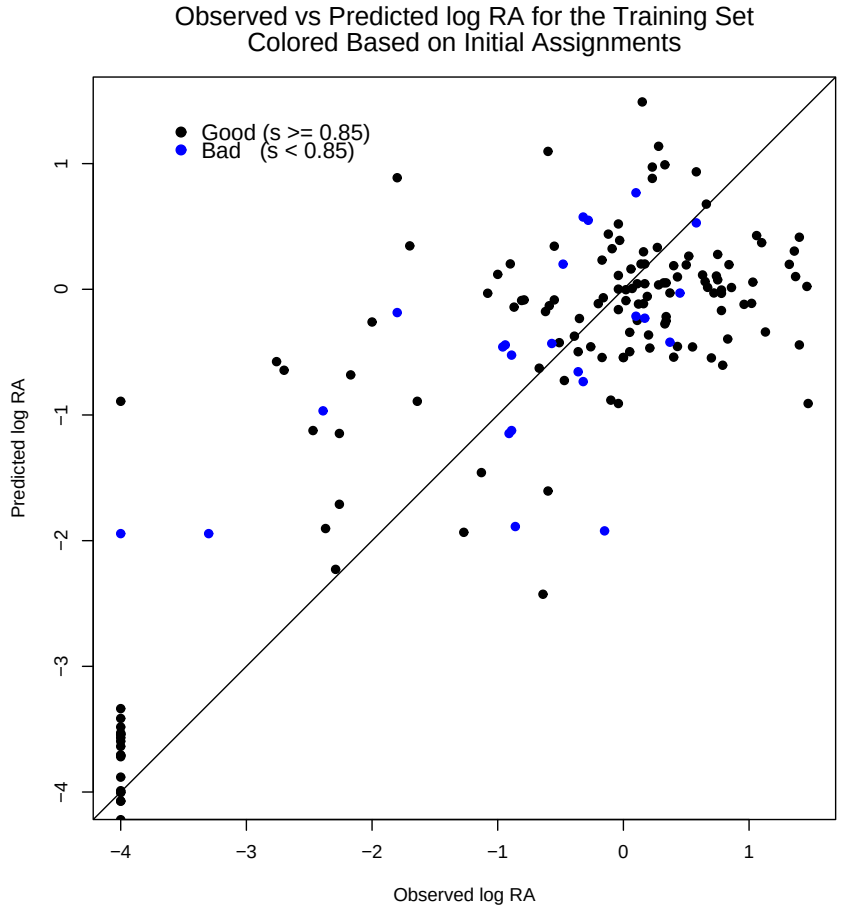
	b	g	✓
b	8	42	16%
g	18	93	83%

PSET Confusion Matrix

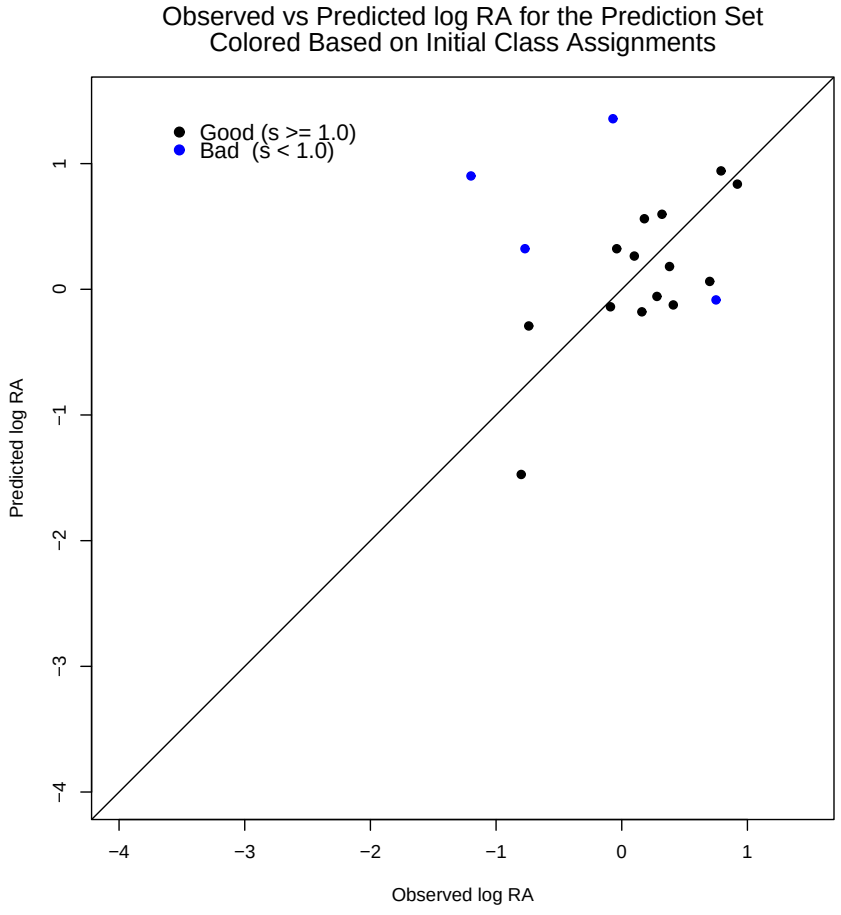
	b	g	✓
b	0	4	0%
g	0	14	100%

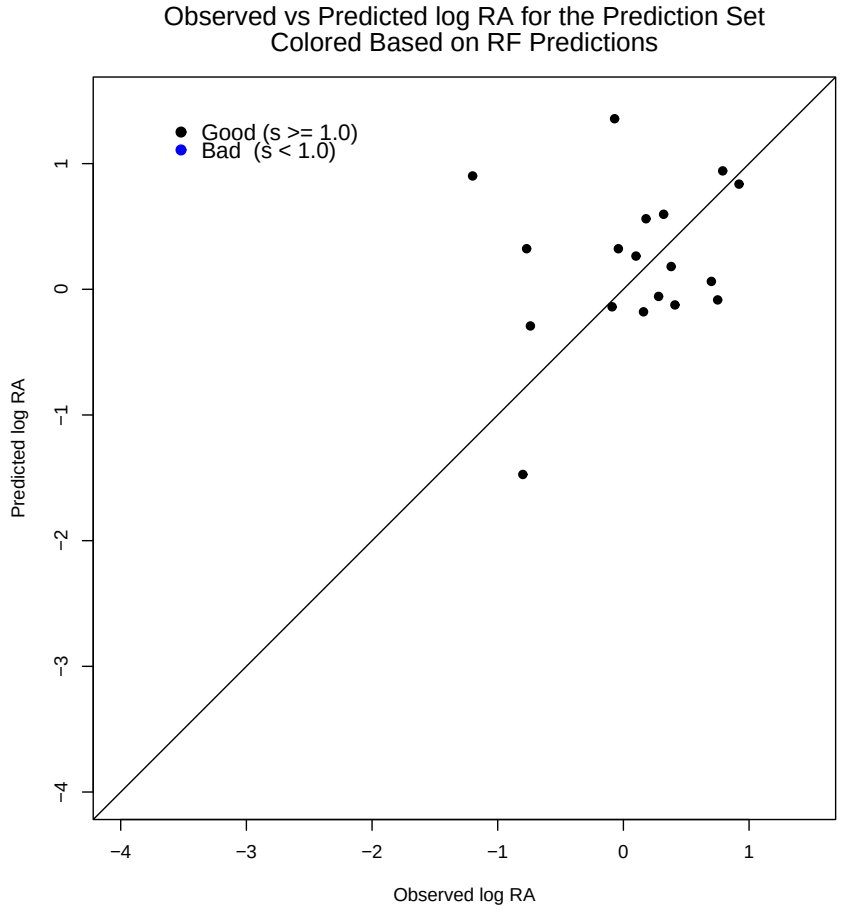
Classification Results - Artemisinin (TSET)





Classification Results - Artemisinin (PSET)





Classification - SVM

- The nice thing about RF is the lack of arbitrary constants!
- Using an SVM appears to give better classification results for the training set
- PSET classification is comparable to the RF

Classification - SVM (Artemisinin)

- Reduced pool was used
- C-classification
- quartic polynomial kernel
- cost = 6.0, coefficient = 2

TSET Confusion Matrix

	b	g	✓
b	50	0	100%
g	0	111	100%

PSET Confusion Matrix

	b	g	✓
b	2	2	50%
g	1	13	92%

Classification - SVM (Tutorial)

- Reduced pool was used
- C-classification
- quartic polynomial kernel
- cost = 6.0, coefficient = 5.5

TSET Confusion Matrix

	b	g	✓
b	51	0	100%
g	0	184	100%

PSET Confusion Matrix

	b	g	✓
b	4	6	40%
g	4	28	87%

Classification Results - SVM

- Artemisinin
  - The *bad* class is poorly predicted but better then RF
  - Using the entire descriptor pool doesn’t help
- Tutorial
  - Training is better than the RF
  - Prediction is not bad, overall rates are similar to RF

Classification

- The problem with SVM's are that there are lots of tunable parameters. Grid search is not a good way to optimize them!
- Training can be made perfect but predictions are not always better than RF
- LDA was also considered:
  - Model descriptors were used
  - Training was poorer than RF and SVM
  - Prediction was comparable (sometimes)

Problems & Further Work

- Two classes might be restrictive. Three or more classes may be useful
- In general the *bad* class is quite small. This is not conducive to good training and leads to poor prediction
- The split point is arbitrary - is there a good way to decide on split points non-visually?
- Not apparent how to actually improve classification rates (AP/fingerprints rather than general descriptors?)
- Are there other regression diagnostics that can be applied here?

Summary

- The whole procedure is very fast
- Minimal user defined choices (with RF)
- Choosing a sensible split point is important
- A good linear model leads to good classification models in general
- Expected to work better with larger datasets