Results

mcc

3/13/2020

Results

I have studied six different M.L. algorithms using protein amino acid percent composition data from two classes. Class number 1 is Myoglobin proteins, which is the positive control set. While the second class is a control group (0) of human proteins that do not have Fe binding centers.

Group	Class	N of Class	Range of Groups
Controls	0 or (-)	1216	1, ..., 1216
Myoglobin	1 or (+)	1124	1217, ..., 2340

The Six M.L Algorithms consist of:

Name	Type	Output Used For Graphing
Principal Component Analysis	Unsupervised	Anomalies > Abs(3σ)
Logistic Regression	Supervised	FP & FN
SVM-linear	Supervised	FP & FN
SVM-polynomial kernel	Supervised	FP & FN
SVM-radial basis function kernel	Supervised	FP & FN
Neural Network	Supervised	FP & FN

==========

Scatter Plots of Anomalies Vs. FP & FN Outputs

To obtain False-Positive (FP) and False-Negatives (FN) from sets:

1. False-Postives $\stackrel{\text{def}}{=}$ $\{\text{obs} = 0 \quad \wedge \quad \text{pred} = 1\}$
2. False-Negatives $\stackrel{\text{def}}{=}$ $\{\text{obs} = 1 \quad \wedge \quad \text{pred} = 0\}$

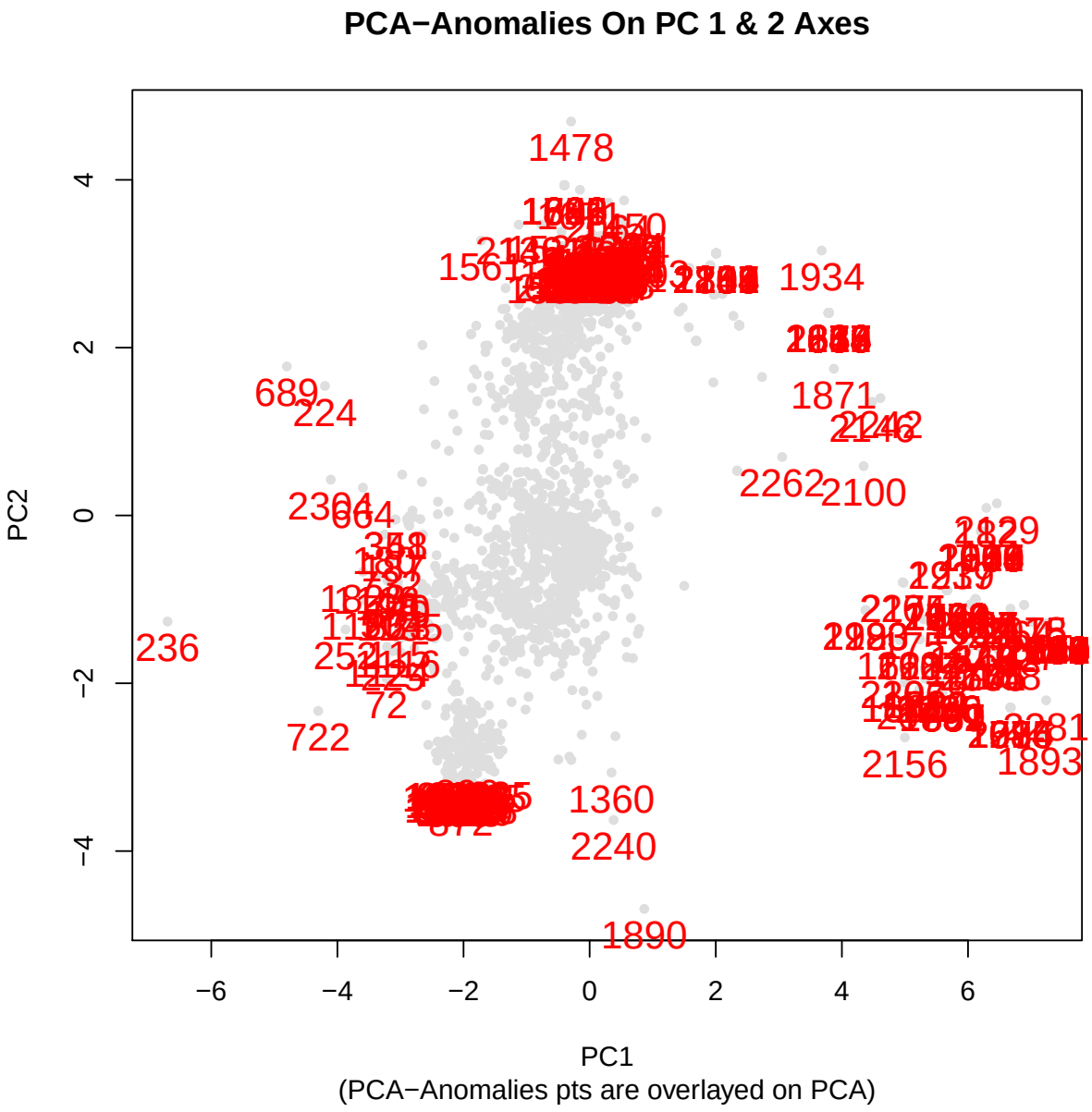
Anomalies Inner Joined with PC

```
# Load Libraries
Libraries = c("knitr", "readr")
for(p in Libraries){
    library(p, character.only = TRUE)
}
opts_chunk$set(fig.align = "center", cache=TRUE)
```

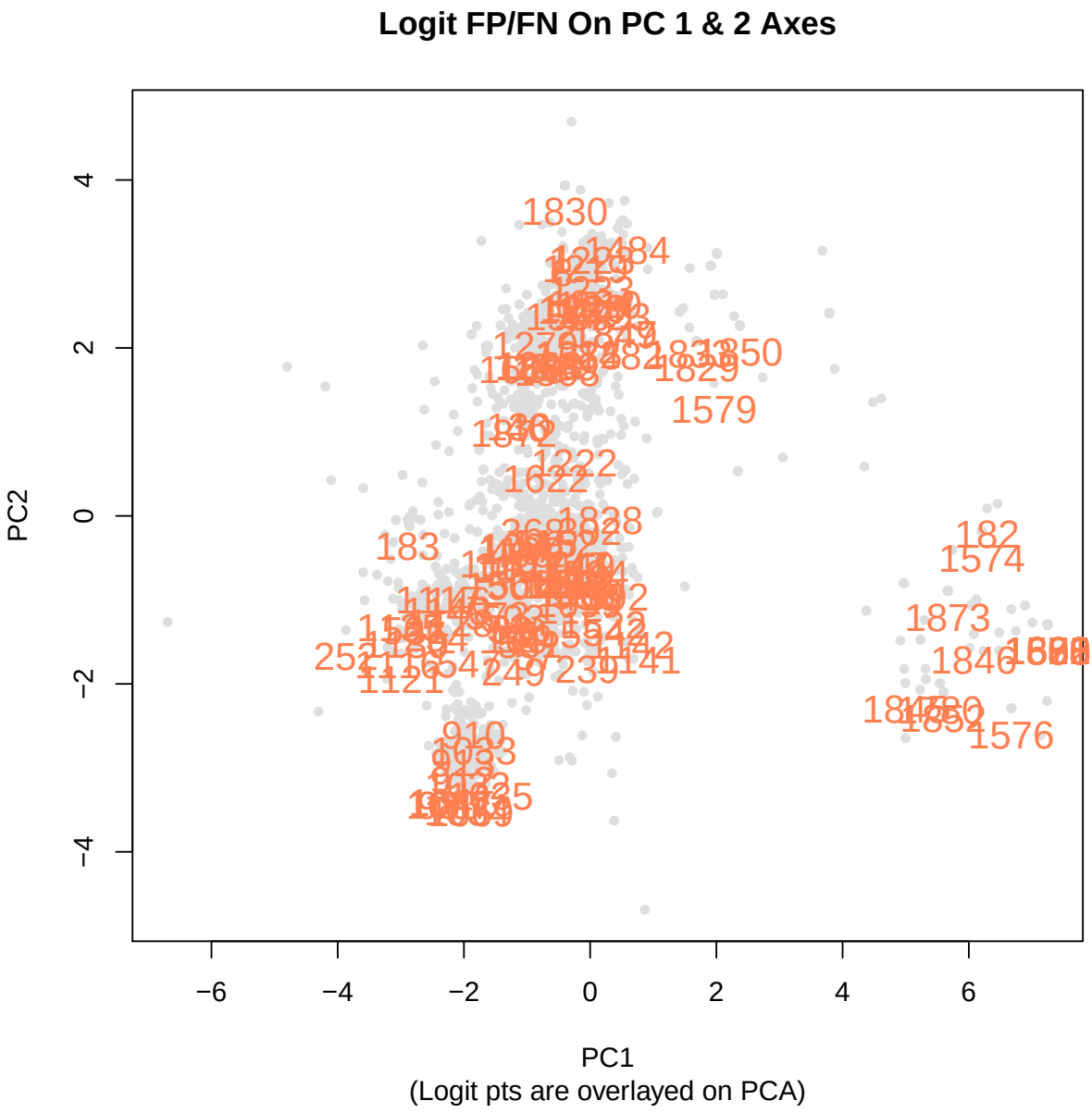
```r
# Prepare PCA: PC1 and PC2 for all 2340 proteins
norm_c_m_20aa <- read_csv("./00-data/03-ml_results/norm_c_m_20aa.csv")

pca_values <- prcomp(norm_c_m_20aa)

row_pc12 <- cbind(rowNum = 1:2340, PC1 = pca_values$x[,1], PC2 = pca_values$x[,2])
# dim(row_pc12)
write.table(row_pc12,
            file = "./00-data/05-joins_plots/row_pc12.txt",
            sep = ",",
            row.names = F)
```

```r
# Inner-join (merge) with PC 1&2 (row_pc12)
logit_pc    <- merge(x = logit_nums,    y = row_pc12, by = "rowNum")
pca_pc      <- merge(x = pca_outliers,  y = row_pc12, by = "rowNum")
nn_pc       <- merge(x = nn_nums,       y = row_pc12, by = "rowNum")
svm_lin_pc  <- merge(x = svm_lin_nums,  y = row_pc12, by = "rowNum")
svm_poly_pc <- merge(x = svm_poly_nums, y = row_pc12, by = "rowNum")
svm_rbf_pc  <- merge(x = svm_rbf_nums,  y = row_pc12, by = "rowNum")
```

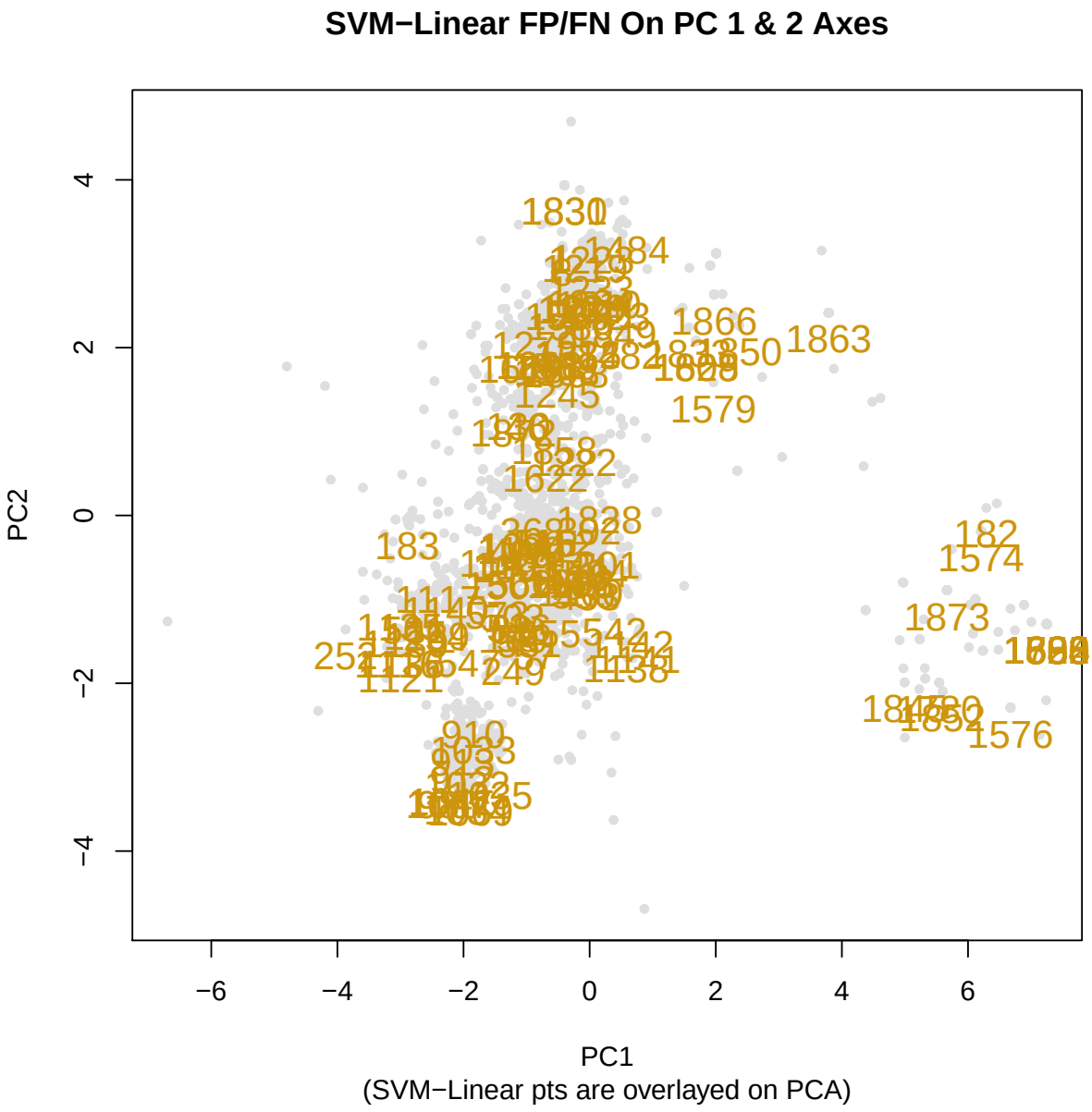
PCA-Anomalies Plot



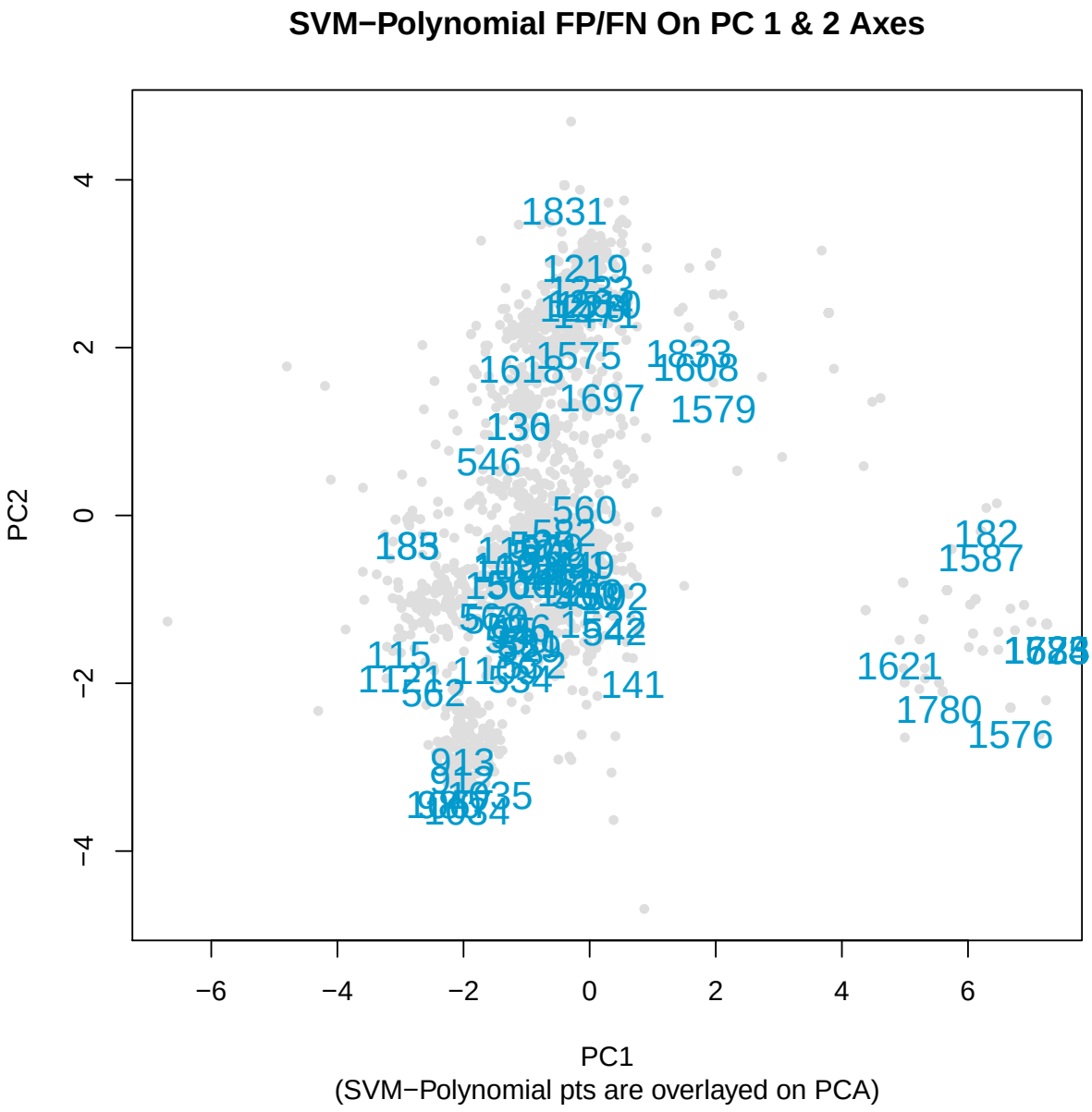
Logit Plot


Logit FP/FN On PC 1 & 2 Axes
PC2
PC1
(Logit pts are overlayed on PCA)
4
2
0
−2
−4
−6
−4
−2
0
2
4
6
1830
1484
1829
1579
1222
1622
183
182
1574
1873
1576

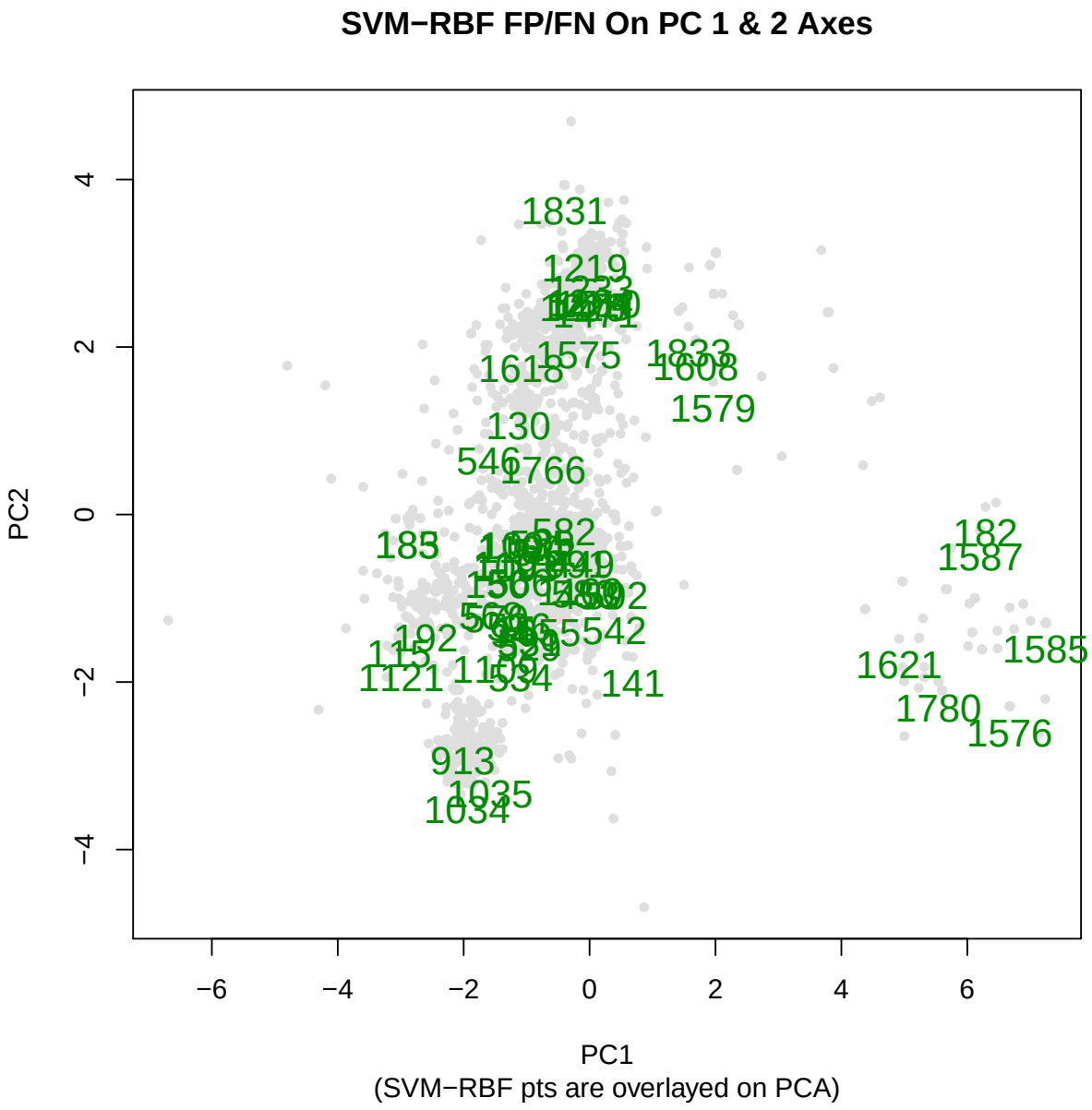
SVM-Linear Plot



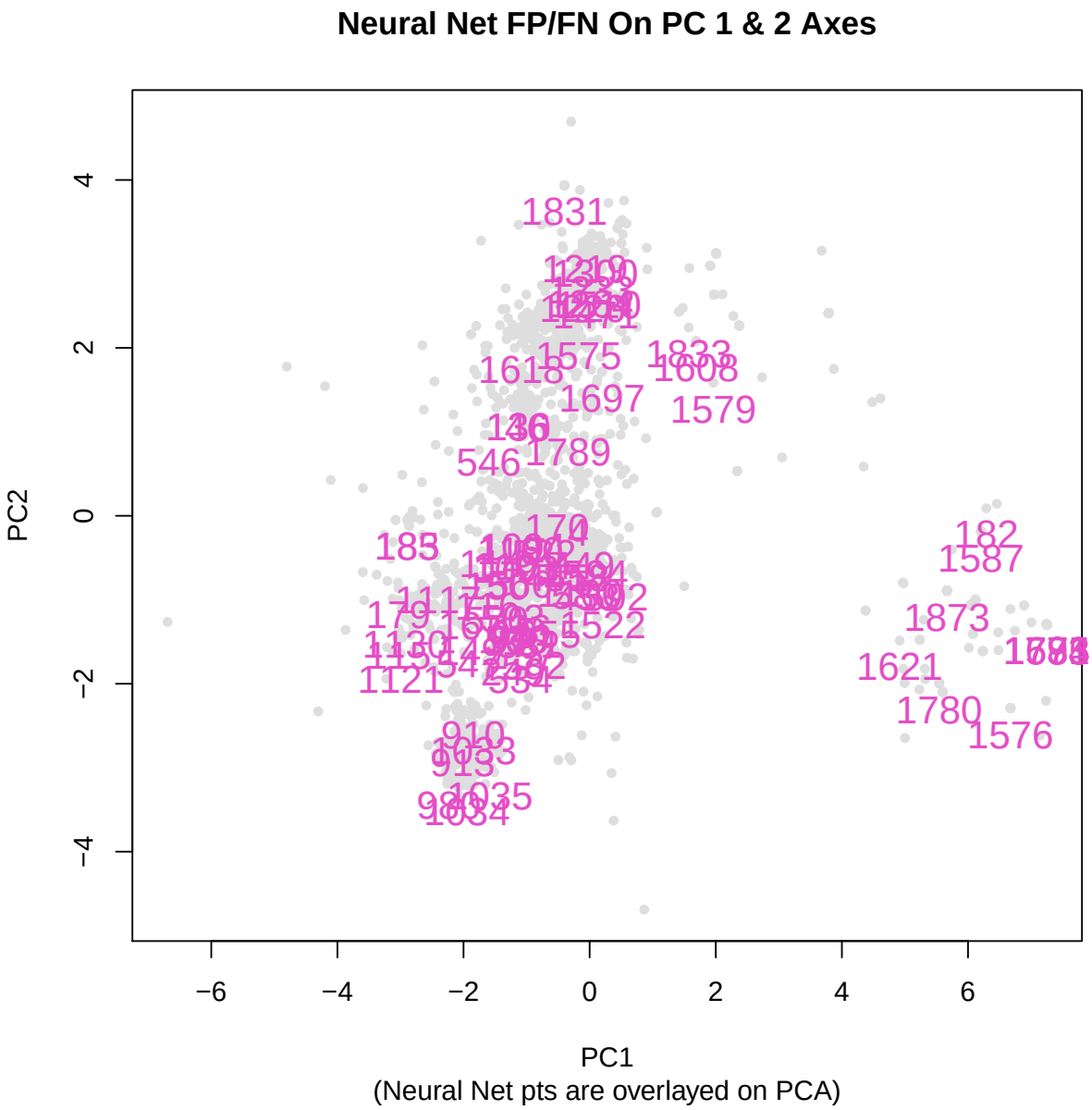
SVM-Polynomial Plot



SVM-Radial Basis Function Plot



Neural Network Function Plot



Statistical Learning Method Vs Total Number of FP/FN

Statistical Method	Unique
Principal Componnent Analysis	460
Logit	119
SVM Linear	125
SVM Polynomial	70
SVM Radial Basis Function	58
Deep Learning	79

Comparison of Machine Learning Accuracies

```
# Load Libraries
Libraries <- c("doMC", "knitr", "readr", "caret", "nnet", "caretEnsemble", "e1071", "kernlab")
for (p in Libraries) {
    library(p, character.only = TRUE)
}
knitr::opts_chunk$set(echo = TRUE, fig.align="center")
```

```
# Import data & data handling
c_m_TRANSFORMED <- read_csv("./00-data/02-aac_dpc_values/c_m_TRANSFORMED.csv",
                            col_types = cols(Class = col_factor(levels = c("0", "1")),
                                             PID = col_skip(),
                                             TotalAA = col_skip()))
```

```
# Partition data into training and testing sets
set.seed(1000)
index <- createDataPartition(c_m_TRANSFORMED$Class, p = 0.8, list = FALSE)

training_set <- c_m_TRANSFORMED[ index,]
test_set     <- c_m_TRANSFORMED[-index,]

Class_test <- as.factor(test_set$Class)
```

Stacking Algorithms - Run multiple algorithms in one call.

```
## Warning in caretList(Class ~ ., data = training_set, trControl = trainControl, :
## Duplicate entries in methodList. Using unqiue methodList values.
```

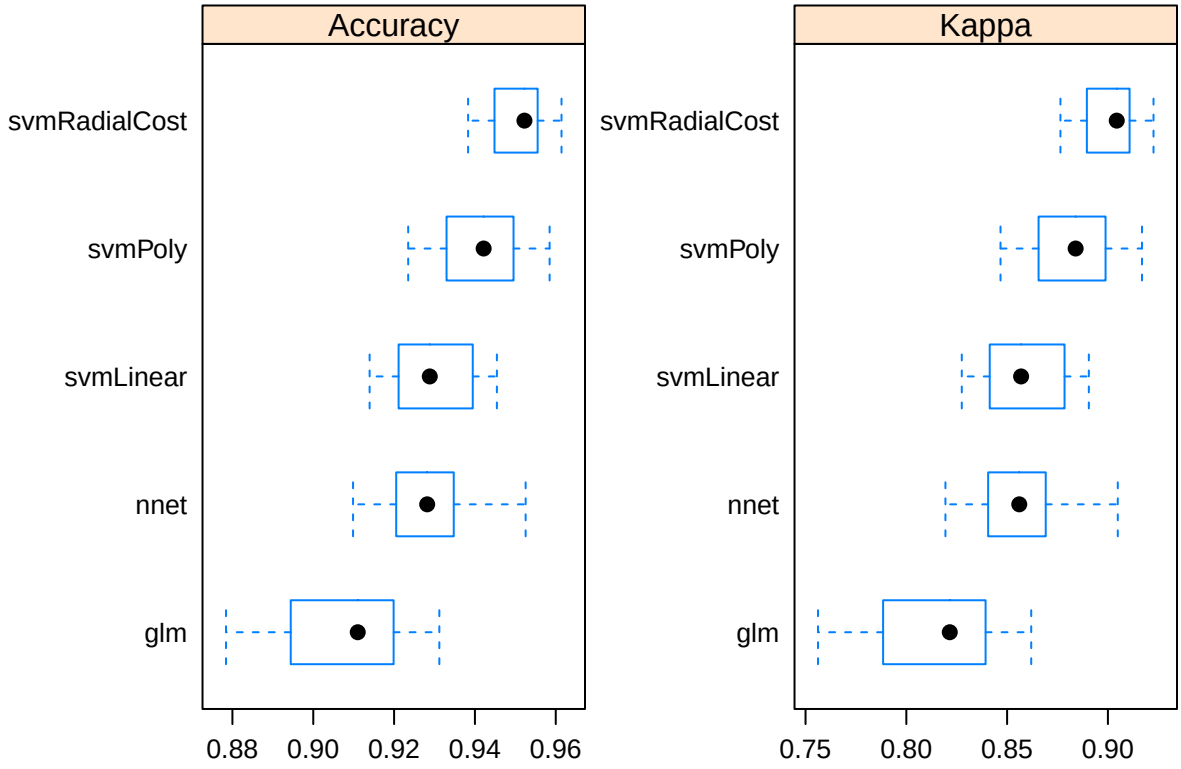
```
## # weights:  287
## initial  value 1622.518111
## iter  10 value 591.200854
## iter  20 value 331.103547
## iter  30 value 281.561829
## iter  40 value 215.474048
## iter  50 value 179.866748
## iter  60 value 111.221062
## iter  70 value 76.163156
## iter  80 value 55.582062
```

```
## iter  90 value 46.458483
## iter 100 value 39.618132
## final  value 39.618132
## stopped after 100 iterations
```

```
summary(results)
```

```
##
## Call:
## summary.resamples(object = results)
##
## Models: glm, nnet, svmLinear, svmPoly, svmRadialCost
## Number of resamples: 10
##
## Accuracy
##                    Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## glm           0.8784104 0.8954626 0.9110056 0.9078127 0.9184104 0.9311981    0
## nnet          0.9098458 0.9208185 0.9281899 0.9286350 0.9334118 0.9525504    0
## svmLinear     0.9139466 0.9215485 0.9288256 0.9292275 0.9373989 0.9454330    0
## svmPoly       0.9234875 0.9346085 0.9421534 0.9415091 0.9485244 0.9584816    0
## svmRadialCost 0.9383155 0.9454330 0.9522539 0.9510603 0.9553610 0.9614243    0
##
## Kappa
##                    Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## glm           0.7561869 0.7905961 0.8215603 0.8152905 0.8365124 0.8619539    0
## nnet          0.8193980 0.8411102 0.8560213 0.8569694 0.8665330 0.9048933    0
## svmLinear     0.8275022 0.8422492 0.8569400 0.8578486 0.8743642 0.8905356    0
## svmPoly       0.8467331 0.8688294 0.8839959 0.8827193 0.8967519 0.9169053    0
## svmRadialCost 0.8764532 0.8906928 0.9043832 0.9019157 0.9104916 0.9226237    0
```

Plot the resamples output to compare the models.



Mean Accuracies of M.L. Techniques, n=10

Rank	M.L. Technique	Mean Accuracy
1	SVM-RBF	0.9510603
2	SVM-Poly	0.9415091
3	SVM-Lin	0.9292275
4	NN w 20 Neurons	0.9286350
5	Logit	0.9078127