TCGA Kidney Tumor Classification

Dong Hyun Kang

2025-04-17

Contents

1 Executive Summary

This report investigates the classification of kidney cancer subtypes using RNA-Seq gene expression data from TCGA. The goal is to identify expression patterns that distinguish the three main subtypes: KIRC, KIRP, and KICH. Principal Component Analysis (PCA) is used to reduce dimensionality and reveal key axes of variation. Three machine learning models—Random Forest, Logistic Regression, and Support Vector Machine—are trained to predict subtype from gene expression. Importantly, the analysis narrows down to a 3-gene model that retains high accuracy, showing potential for clinical translation.

2 Introduction

This analysis aims to classify subtypes of kidney cancer using gene expression data from The Cancer Genome Atlas (TCGA). The subtypes—KIRC, KIRP, and KICH—are known to have distinct clinical outcomes and molecular features. To understand the underlying biological variation and develop predictive models, we apply principal component analysis (PCA) and machine learning classifiers. We also identify a minimal set of predictive genes shared across different models.

This report classifies three TCGA kidney cancer subtypes — **KIRC**, **KIRP**, and **KICH** — using RNA-Seq transcriptome data. PCA and machine learning methods are applied to detect subtype-separating signals. A minimal 3-gene signature is also evaluated.

3 Data Preparation

The dataset contains pre-processed transcriptome quantification data (TPM values) from TCGA kidney tumor samples. These data were downloaded and curated externally, filtered to include only primary tumor samples, and saved in `.rds` format for convenient loading.

```
expr_kirc_df <- readRDS("data/expr_kirc_df.rds")
expr_kirp_df <- readRDS("data/expr_kirp_df.rds")
expr_kich_df <- readRDS("data/expr_kich_df.rds")

expr_all_wide <- bind_rows(expr_kirc_df, expr_kirp_df, expr_kich_df) %>%
  relocate(sample_id, tumor_type)
```

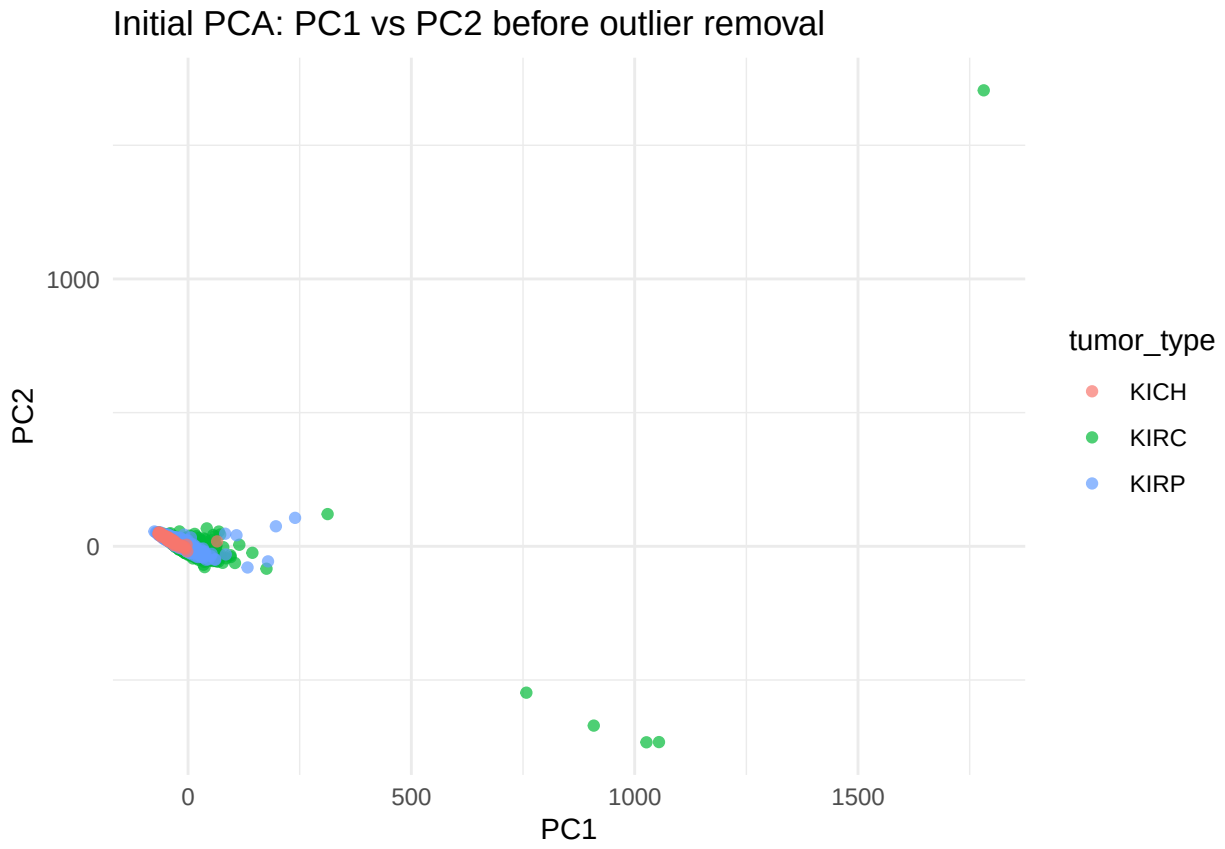
4 PCA-Based Outlier Removal

PCA is performed on the numeric portion of the data to explore global structure and identify potential outliers. Outliers are defined as samples that deviate more than 3 standard deviations from the mean on the first principal component (PC1). These are likely technical or biological outliers and are removed before modeling.

```
numeric_expr <- expr_all_wide %>% dplyr::select(where(is.numeric)) %>% dplyr::select(where(~ sd(.) > 0))
pca_all <- prcomp(numeric_expr, scale. = TRUE)

pca_all_df <- as_tibble(pca_all$x) %>%
  mutate(sample_id = expr_all_wide$sample_id, tumor_type = expr_all_wide$tumor_type)

ggplot(pca_all_df, aes(x = PC1, y = PC2, color = tumor_type)) +
  geom_point(alpha = 0.7) +
  theme_minimal() +
  labs(title = "Initial PCA: PC1 vs PC2 before outlier removal")
```




```
pc1_scores <- pca_all$x[, 1]
thresh_hi <- mean(pc1_scores) + 3 * sd(pc1_scores)
thresh_lo <- mean(pc1_scores) - 3 * sd(pc1_scores)
extreme_pc1 <- expr_all_wide$sample_id[pc1_scores > thresh_hi | pc1_scores < thresh_lo]
expr_clean <- expr_all_wide %>% filter(!sample_id %in% extreme_pc1)
```

5 Exploratory PCA

To explore how gene expression separates tumor types, we plot combinations of principal components (PC1 through PC4). The clearest separation by tumor type was observed in the PC3 and PC4 axes, motivating our downstream gene selection based on their loadings.

```
numeric_expr_clean <- expr_clean %>% dplyr::select(where(is.numeric)) %>% dplyr::select(where(~ sd(.) >
pca_clean <- prcomp(numeric_expr_clean, scale. = TRUE)
pca_clean_df <- as_tibble(pca_clean$x) %>% mutate(tumor_type = expr_clean$tumor_type)

ggplot(pca_clean_df, aes(PC1, PC2, color = tumor_type)) +
  geom_point(alpha = 0.7) +
  theme_minimal() +
  labs(title = "PCA: PC1 vs PC2") +
  labs(caption = "Figure: PC1 and PC2 show poor subtype separation")
```

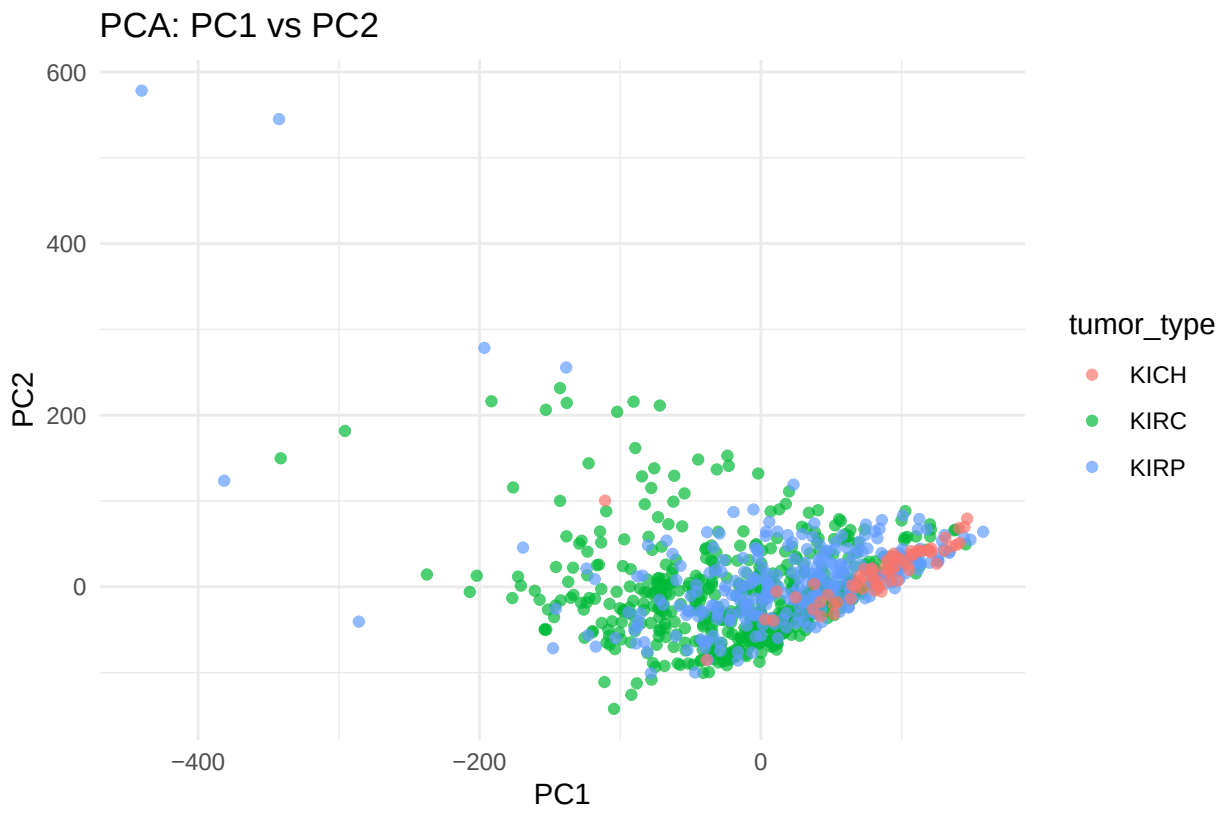


Figure: PC1 and PC2 show poor subtype separation

```
ggplot(pca_clean_df, aes(PC2, PC3, color = tumor_type)) +
  geom_point(alpha = 0.7) +
  theme_minimal() +
  labs(title = "PCA: PC2 vs PC3") +
  labs(caption = "Figure: PC2 and PC3 show moderate separation with some overlap")
```

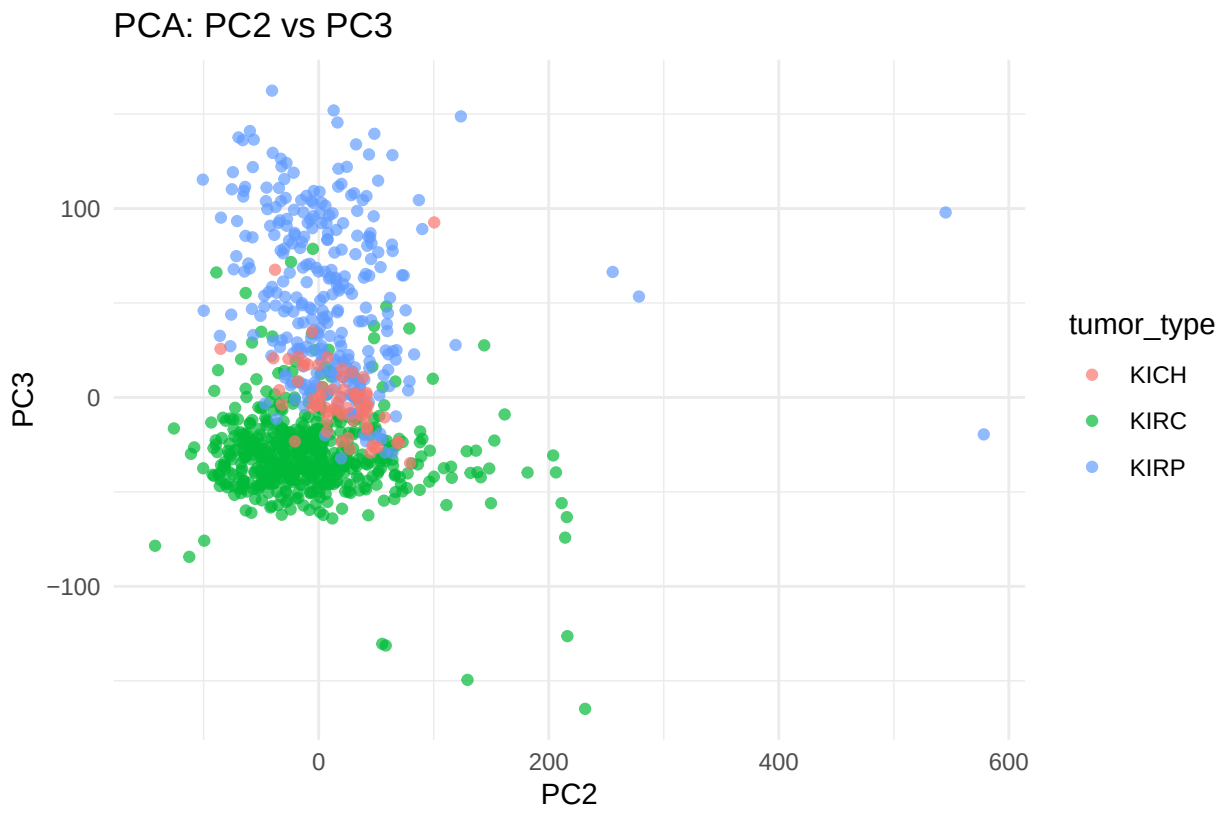


Figure: PC2 and PC3 show moderate separation with some overlap

```
ggplot(pca_clean_df, aes(PC3, PC4, color = tumor_type)) +
  geom_point(alpha = 0.7) +
  theme_minimal() +
  labs(title = "PCA: PC3 vs PC4") +
  labs(caption = "Figure: PC3 and PC4 clearly separate tumor subtypes, guiding downstream gene selectio
```

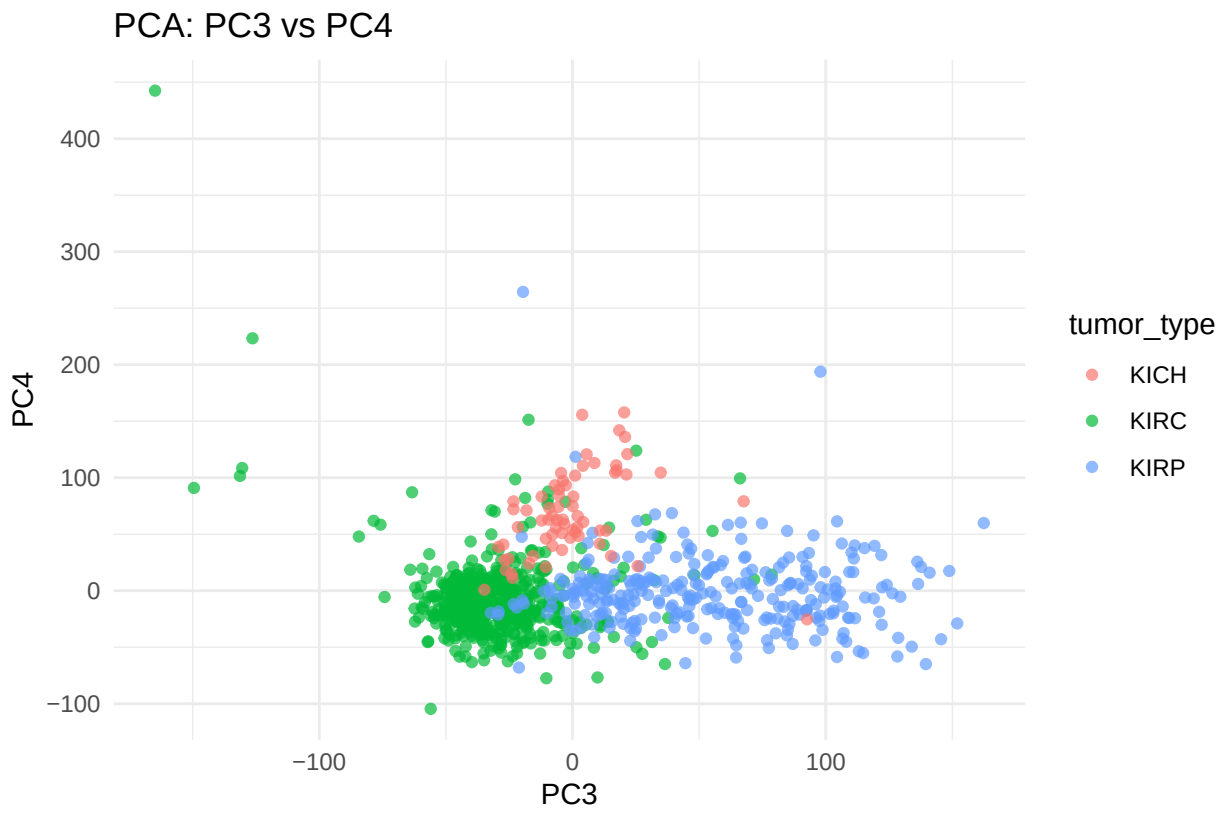


Figure: PC3 and PC4 clearly separate tumor subtypes, guiding downstream gene selection

6 Gene Selection

We select the top 10 genes with the highest absolute loadings from PC3 and PC4. These genes contribute most to the axes that separate tumor subtypes. Their union forms the basis for downstream modeling.

```
rotation <- pca_clean$rotation
top_pc3_genes <- sort(abs(rotation[, 3]), decreasing = TRUE)[1:10]
top_pc4_genes <- sort(abs(rotation[, 4]), decreasing = TRUE)[1:10]
selected_genes <- unique(c(names(top_pc3_genes), names(top_pc4_genes)))
```

7 Modeling

We train three classification models using the selected genes: - Random Forest (RF) - Multinomial Logistic Regression - Support Vector Machine (SVM) with a linear kernel

The dataset is split 80/20 into training and test sets. Expression values are log-transformed to stabilize variance.

```
model_data <- expr_clean %>%
  dplyr::select(all_of(selected_genes), tumor_type) %>%
  mutate(across(all_of(selected_genes), ~log2(. + 1)))

set.seed(42)
```

```r
split_idx <- createDataPartition(model_data$tumor_type, p = 0.8, list = FALSE)
train <- model_data[split_idx, ]; test <- model_data[-split_idx, ]
train$tumor_type <- factor(train$tumor_type); test$tumor_type <- factor(test$tumor_type)

# RF
rf_model <- randomForest(tumor_type ~ ., data = train)
pred_rf <- predict(rf_model, test)
conf_rf <- confusionMatrix(pred_rf, test$tumor_type)

# Logit
logit_model <- multinom(tumor_type ~ ., data = train)
```

```
## # weights:  66 (42 variable)
## initial  value 785.507786
## iter  10 value 245.523709
## iter  20 value 159.841044
## iter  30 value 145.007757
## iter  40 value 142.919961
## iter  50 value 142.601118
## iter  60 value 142.597836
## final  value 142.597826
## converged
```

```r
pred_logit <- predict(logit_model, test)
conf_logit <- confusionMatrix(pred_logit, test$tumor_type)

# SVM
train_matrix <- as.matrix(train[, setdiff(names(train), "tumor_type")])
svm_model <- ksvm(train_matrix, train$tumor_type, type = "C-svc", kernel = "vanilladot")
```

```
##  Setting default kernel parameters
```

```r
test_matrix <- as.matrix(test[, setdiff(names(test), "tumor_type")])
pred_svm <- predict(svm_model, test_matrix)
conf_svm <- confusionMatrix(pred_svm, test$tumor_type)
```

8 Top Gene Comparison

To identify stable biomarkers, we extract the top 10 most important genes from each model. We then compute their intersection to find genes ranked consistently highly across all three models.

```r
rf_importance <- importance(rf_model)[, 1]
top10_rf <- sort(rf_importance, decreasing = TRUE)[1:10]
logit_importance <- apply(abs(summary(logit_model)$coefficients), 2, max)
top10_logit <- sort(logit_importance, decreasing = TRUE)[1:10]
W <- colSums(coef(svm_model)[[1]] * svm_model@xmatrix[[1]])
names(W) <- colnames(train_matrix)
top10_svm <- sort(abs(W), decreasing = TRUE)[1:10]
top10_overlap <- Reduce(intersect, list(names(top10_rf), names(top10_logit), names(top10_svm)))
top10_overlap
```

```
## [1] "ENSG00000167646.14" "ENSG00000169727.12" "ENSG00000204237.5"
```

9 Minimal 3-Gene Signature

The genes **DNAAF3**, **GPS1**, and **OXLD1** were consistently top-ranked across all three models. We train new RF, logistic regression, and SVM models using only these three genes and compare their performance against the full gene set models.

```
top3_genes <- c("ENSG00000167646.14", "ENSG00000169727.12", "ENSG00000204237.5")
model_top3 <- expr_clean %>%
  dplyr::select(all_of(top3_genes), tumor_type) %>%
  mutate(across(all_of(top3_genes), ~log2(. + 1)))

split_idx3 <- createDataPartition(model_top3$tumor_type, p = 0.8, list = FALSE)
train3 <- model_top3[split_idx3, ]; test3 <- model_top3[-split_idx3, ]
train3$tumor_type <- factor(train3$tumor_type); test3$tumor_type <- factor(test3$tumor_type)

rf3 <- randomForest(tumor_type ~ ., data = train3)
logit3 <- multinom(tumor_type ~ ., data = train3)
```

```
## # weights:  15 (8 variable)
## initial  value 785.507786
## iter  10 value 276.627032
## iter  20 value 251.200126
## iter  30 value 248.572533
## final  value 248.569718
## converged
```

```
train3_matrix <- as.matrix(train3[, setdiff(names(train3), "tumor_type")])
test3_matrix <- as.matrix(test3[, setdiff(names(test3), "tumor_type")])
svm3 <- ksvm(train3_matrix, train3$tumor_type, type = "C-svc", kernel = "vanilladot")
```

```
##  Setting default kernel parameters
```

```
list(
  RF = confusionMatrix(predict(rf3, test3), test3$tumor_type),
  Logit = confusionMatrix(predict(logit3, test3), test3$tumor_type),
  SVM = confusionMatrix(predict(svm3, test3_matrix), test3$tumor_type)
)
```

```
## $RF
## Confusion Matrix and Statistics
##
##           Reference
## Prediction KICH KIRC KIRP
##       KICH    7    3    0
##       KIRC    5   98    4
##       KIRP    1    6   54
##
## Overall Statistics
##
##                Accuracy : 0.8933
##                  95% CI : (0.8383, 0.9345)
##     No Information Rate : 0.6011
```

```
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.7959
##
##  Mcnemar's Test P-Value : 0.5934
##
## Statistics by Class:
##
##                      Class: KICH Class: KIRC Class: KIRP
## Sensitivity              0.53846      0.9159      0.9310
## Specificity              0.98182      0.8732      0.9417
## Pos Pred Value           0.70000      0.9159      0.8852
## Neg Pred Value           0.96429      0.8732      0.9658
## Prevalence               0.07303      0.6011      0.3258
## Detection Rate           0.03933      0.5506      0.3034
## Detection Prevalence     0.05618      0.6011      0.3427
## Balanced Accuracy        0.76014      0.8946      0.9364
##
## $Logit
## Confusion Matrix and Statistics
##
##           Reference
## Prediction KICH KIRC KIRP
##       KICH    8    1    1
##       KIRC    4  102    4
##       KIRP    1    4   53
##
## Overall Statistics
##
##                Accuracy : 0.9157
##                  95% CI : (0.8648, 0.9521)
##     No Information Rate : 0.6011
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8374
##
##  Mcnemar's Test P-Value : 0.6149
##
## Statistics by Class:
##
##                      Class: KICH Class: KIRC Class: KIRP
## Sensitivity              0.61538      0.9533      0.9138
## Specificity              0.98788      0.8873      0.9583
## Pos Pred Value           0.80000      0.9273      0.9138
## Neg Pred Value           0.97024      0.9265      0.9583
## Prevalence               0.07303      0.6011      0.3258
## Detection Rate           0.04494      0.5730      0.2978
## Detection Prevalence     0.05618      0.6180      0.3258
## Balanced Accuracy        0.80163      0.9203      0.9361
##
## $SVM
## Confusion Matrix and Statistics
##
##           Reference
```

```
## Prediction KICH KIRC KIRP
##       KICH    4    2    0
##       KIRC    8  101    5
##       KIRP    1    4   53
##
## Overall Statistics
##
##                Accuracy : 0.8876
##                  95% CI : (0.8318, 0.93)
##     No Information Rate : 0.6011
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.7781
##
##  Mcnemar's Test P-Value : 0.1942
##
## Statistics by Class:
##
##                      Class: KICH Class: KIRC Class: KIRP
## Sensitivity              0.30769      0.9439      0.9138
## Specificity              0.98788      0.8169      0.9583
## Pos Pred Value           0.66667      0.8860      0.9138
## Neg Pred Value           0.94767      0.9062      0.9583
## Prevalence               0.07303      0.6011      0.3258
## Detection Rate           0.02247      0.5674      0.2978
## Detection Prevalence     0.03371      0.6404      0.3258
## Balanced Accuracy        0.64779      0.8804      0.9361
```

10 Conclusion

This analysis demonstrates that a small number of genes can be used to effectively classify kidney tumor subtypes. The models trained using only three genes achieved similar performance to those using 20+ genes, indicating the robustness and potential clinical utility of this minimal gene signature.

- PCA revealed PC3/PC4 as separating dimensions.
- DNAAF3, GPS1, and OXLD1 were selected from top overlapping genes.
- Models showed high accuracy even with only 3 genes.

11 References

- TCGA: https://www.cancer.gov/ccg/research/genome-sequencing/tcga
- GDC Portal: https://portal.gdc.cancer.gov/
- BiomaRt: https://bioconductor.org/packages/release/bioc/html/biomaRt.html
- caret: https://topepo.github.io/caret/
- kernlab: https://cran.r-project.org/package=kernlab
- randomForest: https://cran.r-project.org/package=randomForest
- nnet: https://cran.r-project.org/package=nnet