Quality Control

Sergey Naumenko

2021-03-29

Contents

Overview

- Principal Investigator: Beth Overmoyer
- Experiment: RNAseq_analysis_of_inflammatory_breast_cancer_hbc04141
- study 6 was excluded

```
library(tidyverse)
library(knitr)
library(DESeq2)
library(DEGreport)
library(ggrepel)
library(pROC)
```

```
library(randomForest)
#devtools::install_github("sachsmc/plotROC")
#library(plotROC)

ggplot2::theme_set(theme_light(base_size = 14))

opts_chunk[["set"]](
    cache = FALSE,
    dev = c("png", "pdf"),
    error = TRUE,
    highlight = TRUE,
    message = FALSE,
    prompt = FALSE,
    tidy = FALSE,
    warning = FALSE)
```

Metadata

```
se <- readRDS("data/bcbio-se.rds")

metadata <- colData(se) %>%
    as_tibble(rownames = NULL) %>%
    dplyr::select(-batch, -phenotype)

metrics <- metadata(se)$metrics %>%
    left_join(metadata, by = c("sample" = "sample"))

metrics$date_of <- as_factor(metrics$date_of)

metadata
```

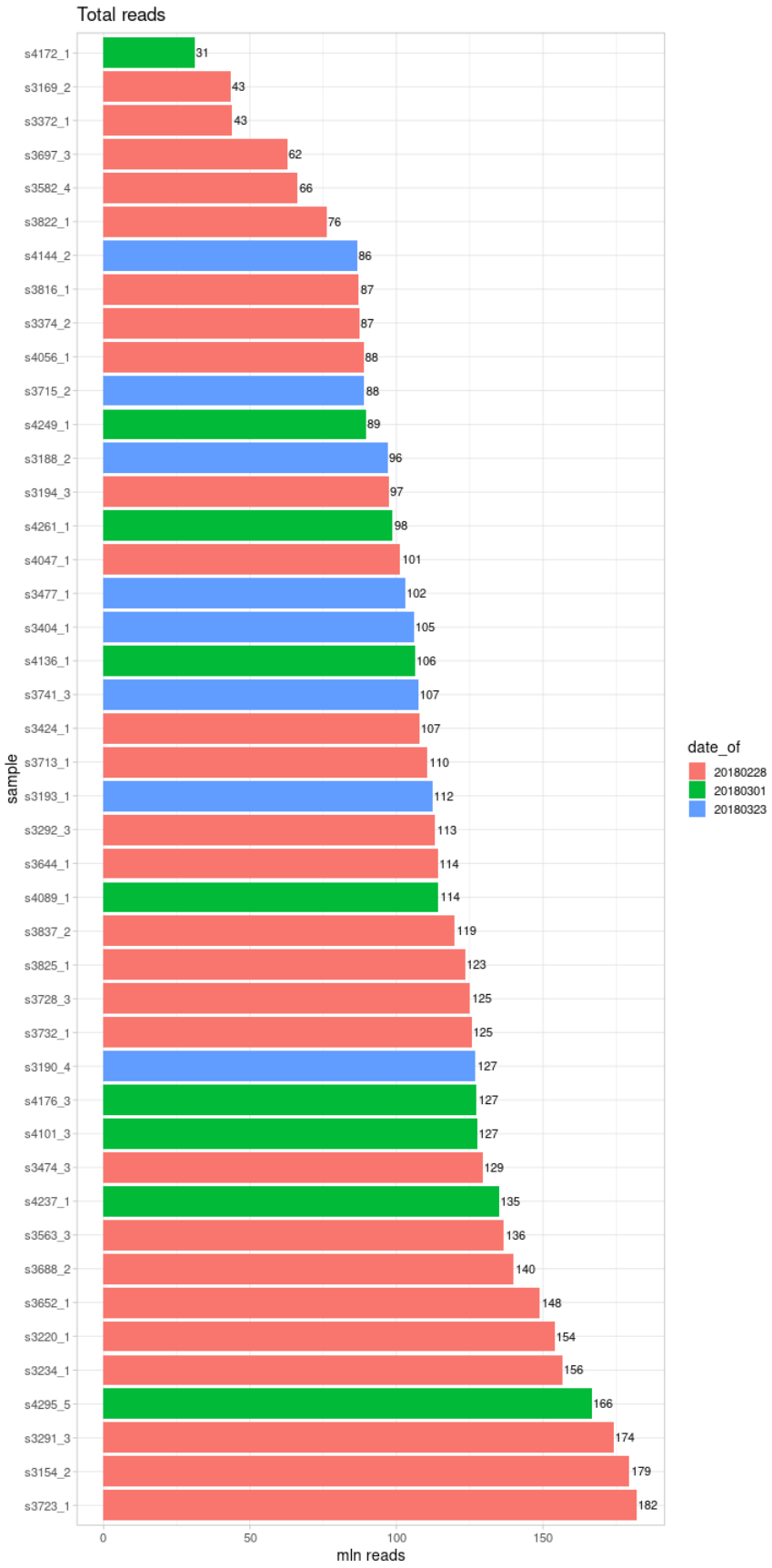
```
## # A tibble: 44 x 8
##    category  date_of er      response study_id treatment tumor_percentage sample
##    <chr>       <dbl> <chr>   <chr>       <dbl> <chr>     <chr>            <chr>
##  1 pre      20180228 Positi~ Yes             1 pre       30-100           s3154~
##  2 post     20180228 Positi~ Yes             1 post      30-100           s3169~
##  3 pre      20180323 Positi~ No              3 pre       1-29             s3188~
##  4 pre      20180323 Positi~ <NA>            2 pre       30-100           s3190~
##  5 post     20180323 Positi~ No              3 post      1-29             s3193~
##  6 post     20180228 Positi~ <NA>            2 post      1-29             s3194~
##  7 pre      20180228 Positi~ Yes             4 pre       30-100           s3220~
##  8 post     20180228 Positi~ Yes             4 post      30-100           s3234~
##  9 pre      20180228 Positi~ Yes             5 pre       1-29             s3291~
## 10 post     20180228 Positi~ Yes             5 post      1-29             s3292~
## # ... with 34 more rows
```

Read metrics

Total reads

```
metrics <- metrics %>%
    group_by(date_of) %>%
```

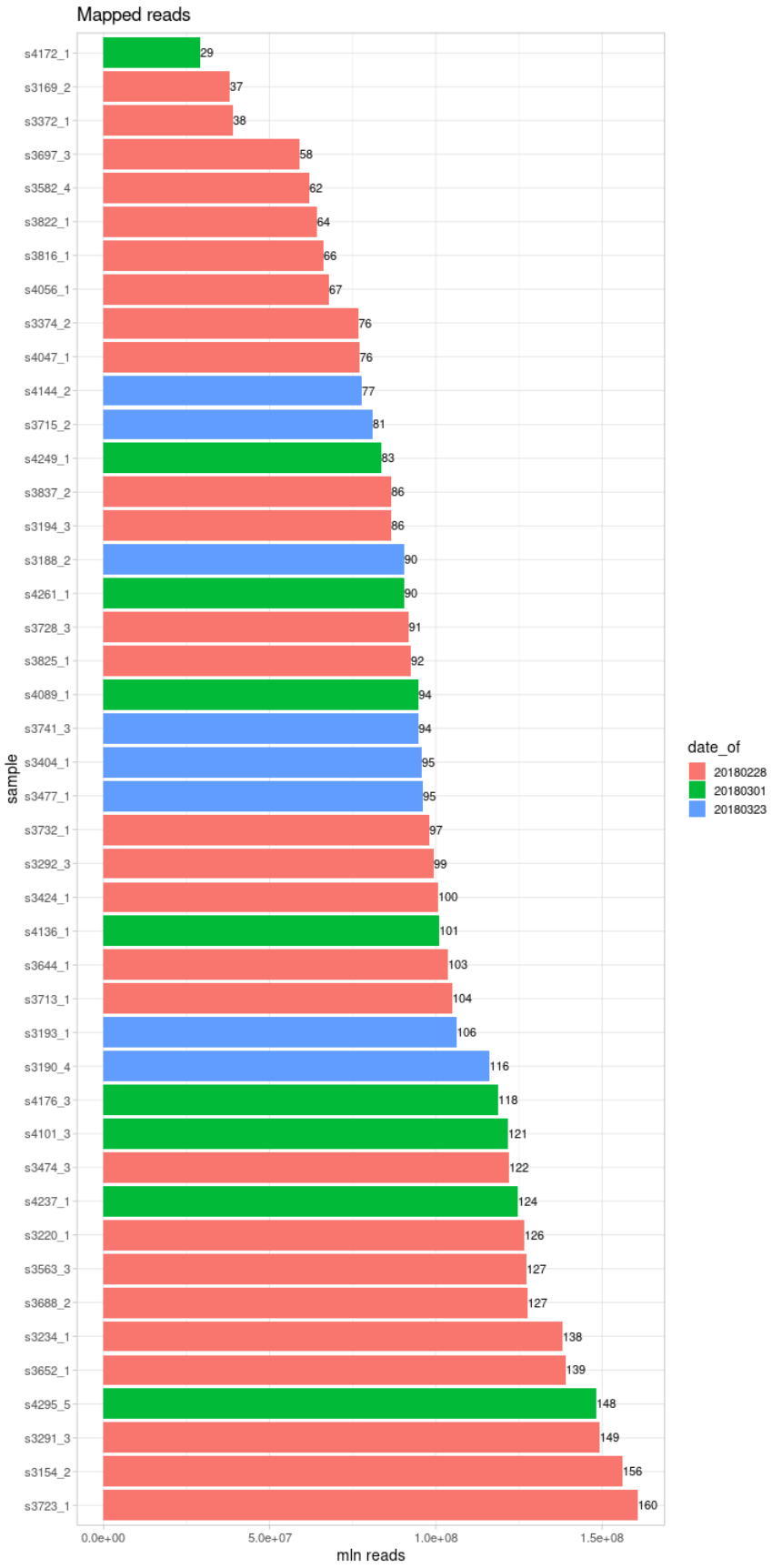
```
    mutate(position = rank(-total_reads))

metrics %>%
    ggplot(aes(x = reorder(sample, -total_reads),
               y = total_reads/1e6L,
               fill = date_of,
               group = position)) +
    geom_bar(stat = "identity") +
    coord_flip() +
    scale_y_continuous(name = "mln reads") +
    geom_text(aes(label = floor(total_reads/1e6L)), hjust = 0, nudge_y = 0.5)+
    xlab("sample") +
    ggtitle("Total reads")
```


Total reads
s4172_1 31
s3169_2 43
s3372_1 43
s3697_3 62
s3582_4 66
s3822_1 76
s4144_2 86
s3816_1 87
s3374_2 87
s4056_1 88
s3715_2 88
s4249_1 89
s3188_2 96
s3194_3 97
s4261_1 98
s4047_1 101
s3477_1 102
s3404_1 105
s4136_1 106
s3741_3 107
s3424_1 107
s3713_1 110
s3193_1 112
s3292_3 113
s3644_1 114
s4089_1 114
s3837_2 119
s3825_1 123
s3728_3 125
s3732_1 125
s3190_4 127
s4176_3 127
s4101_3 127
s3474_3 129
s4237_1 135
s3563_3 136
s3688_2 140
s3652_1 148
s3220_1 154
s3234_1 156
s4295_5 166
s3291_3 174
s3154_2 179
s3723_1 182
sample
0
50
100
150
mln reads
date_of
20180228
20180301
20180323

Mapped reads

The number of mapped reads should correspond to the number of total reads.

```
metrics %>%
    ggplot(aes(x = reorder(sample, -mapped_reads),
               y = mapped_reads, fill = date_of)) +
    geom_bar(stat = "identity") +
    coord_flip() +
    scale_y_continuous(name = "mln reads") +
    geom_text(aes(label = floor(mapped_reads/1e6L)), hjust = 0, nudge_y = 0.5)+
    xlab("sample") +
    ggtitle("Mapped reads")
```


Mapped reads
sample
s4172_1
s3169_2
s3372_1
s3697_3
s3582_4
s3822_1
s3816_1
s4056_1
s3374_2
s4047_1
s4144_2
s3715_2
s4249_1
s3837_2
s3194_3
s3188_2
s4261_1
s3728_3
s3825_1
s4089_1
s3741_3
s3404_1
s3477_1
s3732_1
s3292_3
s3424_1
s4136_1
s3644_1
s3713_1
s3193_1
s3190_4
s4176_3
s4101_3
s3474_3
s4237_1
s3220_1
s3563_3
s3688_2
s3234_1
s3652_1
s4295_5
s3291_3
s3154_2
s3723_1
29
37
38
58
62
64
66
67
76
76
77
81
83
86
86
90
90
91
92
94
94
95
95
97
99
100
101
103
104
106
116
118
121
122
124
126
127
127
138
139
148
149
156
160
0.0e+00
5.0e+07
1.0e+08
1.5e+08
mln reads
date_of
20180228
20180301
20180323

```
## Mapping rate
#The genomic mapping rate represents the percentage of reads mapping to the reference genome. Low mappi
#{r plot_mapping_rate, fig.width = 10, fig.height = 20}
#metrics %>%
#    ggplot(aes(x = reorder(sample, -mapped_reads_pct),
#               y = mapped_reads_pct, fill = tissue)) +
#        geom_bar(stat = "identity") +
#    coord_flip() +
#    geom_text(aes(label = floor(mapped_reads_pct)), hjust = 0, nudge_y = 0.5)+
#    xlab("sample") +
#    ggtitle("Mapping rate")
```

Number of genes detected

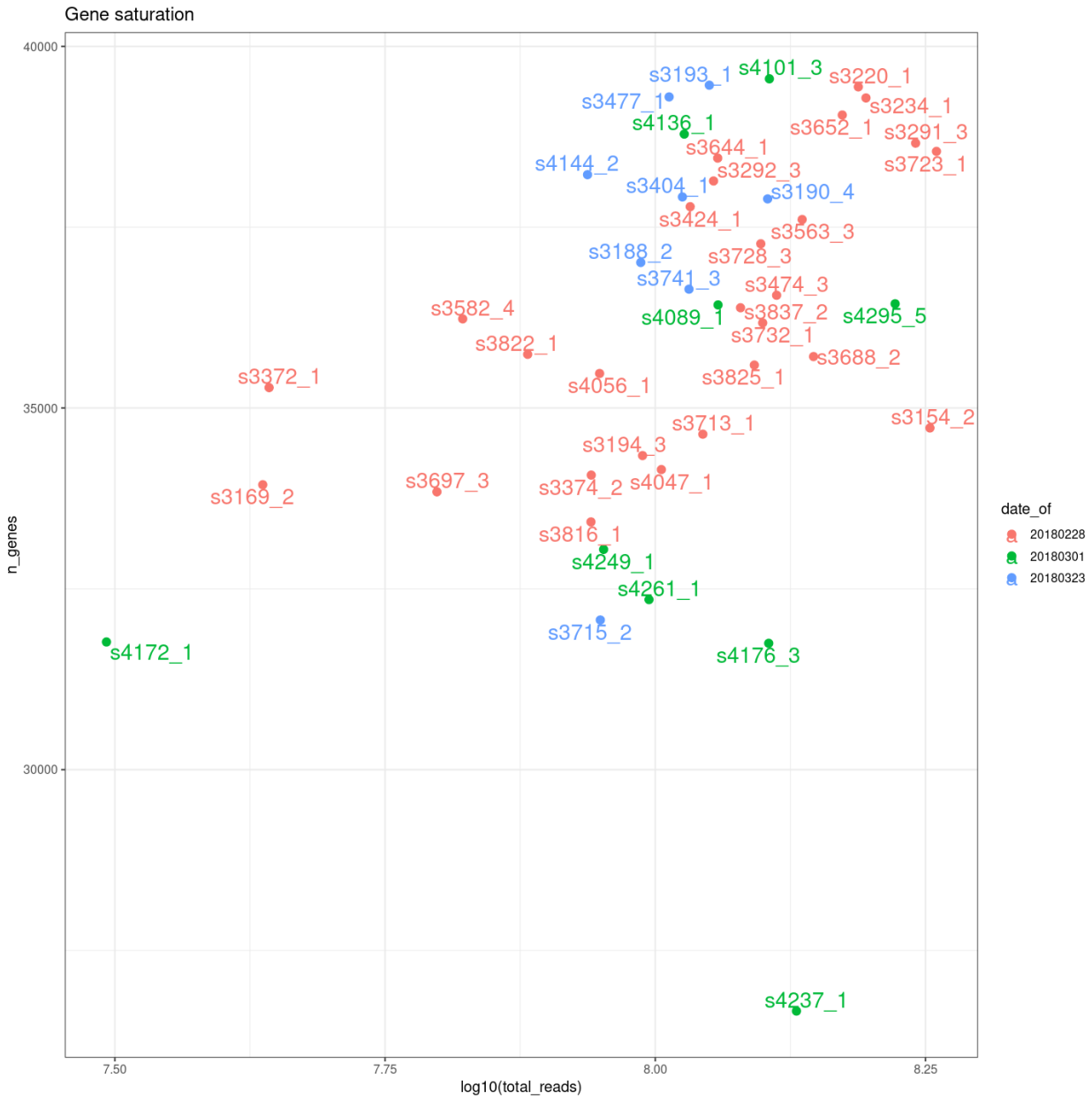
```
genes_detected <- colSums(assays(se)[["raw"]] > 0) %>% enframe()
colnames(genes_detected) <- c("sample", "n_genes")

metrics <- metrics %>%
    left_join(genes_detected, by = c("sample" = "sample"))

metrics %>%
    ggplot(aes(x = reorder(sample, -n_genes),
               y = n_genes, fill = date_of)) +
    geom_bar(stat = "identity") +
    coord_flip() +
    geom_text(aes(label = n_genes), hjust = 0, nudge_y = 0.5)+
    xlab("sample") +
    ylim(0, 50000) +
    ggtitle("Number of genes")
```


Number of genes
sample
n_genes
0
10000
20000
30000
40000
50000
s4237_1 26662
s4176_3 31747
s4172_1 31765
s3715_2 32069
s4261_1 32351
s4249_1 33045
s3816_1 33425
s3697_3 33841
s3169_2 33939
s3374_2 34072
s4047_1 34149
s3194_3 34342
s3713_1 34638
s3154_2 34724
s3372_1 35282
s4056_1 35477
s3825_1 35594
s3688_2 35711
s3822_1 35742
s3732_1 36177
s3582_4 36232
s3837_2 36386
s4089_1 36425
s4295_5 36441
s3474_3 36559
s3741_3 36643
s3188_2 37012
s3728_3 37271
s3563_3 37606
s3424_1 37783
s3190_4 37891
s3404_1 37918
s3292_3 38139
s4144_2 38226
s3644_1 38455
s3723_1 38548
s3291_3 38662
s4136_1 38787
s3652_1 39051
s3234_1 39287
s3477_1 39301
s3220_1 39440
s3193_1 39463
s4101_3 39551
date_of
20180228
20180301
20180323

Gene detection saturation

We should observe a linear trend in the number of genes detected with the number of mapped reads, which indicates that the sample input was not overloaded.

```
metrics %>%
    ggplot(aes(x = log10(total_reads),
               y = n_genes,
               color = date_of)) +
        geom_point(size = 5)+
        geom_text_repel(aes(label = sample), size = 10)+
    theme(axis.text = element_text(size = 20),
          axis.title = element_text(size = 20, face="bold"),
          legend.text = element_text(size = rel(1))) +
    theme_bw(base_size = 20) +
    ggtitle("Gene saturation")
```



Exonic mapping rate

Ideally, at least 60% of total reads should map to exons.

```
metrics %>%
    ggplot(aes(x = reorder(sample, -exonic_rate),
               y = exonic_rate * 100,
               fill = date_of)) +
        geom_bar(stat = "identity") +
    geom_text(aes(label = floor(exonic_rate*100)), hjust = 0, nudge_y = 0.5)+
    xlab("sample") +
    ylab("Exonic rate %") +
    ggtitle("Exonic mapping rate") +
    coord_flip()
```


Exonic mapping rate
sample
s4261_1 31
s4295_5 32
s4144_2 33
s4056_1 33
s3741_3 34
s4176_3 36
s3644_1 37
s4172_1 37
s3477_1 38
s3372_1 38
s3652_1 39
s3582_4 40
s3193_1 40
s3424_1 40
s3728_3 41
s3697_3 41
s4237_1 43
s3220_1 43
s4136_1 43
s4249_1 44
s3732_1 44
s3291_3 45
s4047_1 46
s3188_2 46
s3688_2 46
s3837_2 46
s4101_3 48
s3374_2 48
s3404_1 48
s3169_2 49
s4089_1 49
s3190_4 50
s3292_3 50
s3194_3 50
s3713_1 50
s3474_3 52
s3234_1 52
s3822_1 55
s3563_3 55
s3816_1 57
s3723_1 59
s3715_2 59
s3825_1 60
s3154_2 63
0
20
40
60
Exonic rate %
date_of
20180228
20180301
20180323

Intronic mapping rate

The majority of reads should map to exons and not introns.

```
metrics %>%
    ggplot(aes(x = reorder(sample, -intronic_rate),
               y = intronic_rate * 100,
               fill = date_of)) +
        geom_bar(stat = "identity") +
        coord_flip() +
        geom_text(aes(label = floor(intronic_rate*100)),
                  hjust = 0,
                  nudge_y = 0.5) +
        xlab("sample") +
        ylab("Intronic rate %")
```


s3154_2 32
s3825_1 35
s3715_2 36
s3723_1 36
s3816_1 38
s3822_1 39
s3563_3 39
s3474_3 42
s3234_1 42
s3713_1 43
s3190_4 44
s3292_3 44
s3194_3 44
s3169_2 44
s4089_1 45
s3374_2 46
s3404_1 46
s4101_3 46
s4047_1 47
s3837_2 48
s3188_2 48
s3688_2 48
s3732_1 48
s3291_3 49
s4249_1 50
s3220_1 51
s4237_1 51
s4136_1 51
s3728_3 52
s3193_1 53
s3424_1 53
s3697_3 53
s3582_4 54
s3372_1 55
s4172_1 55
s3652_1 55
s3477_1 56
s4176_3 57
s3644_1 57
s3741_3 58
s4056_1 59
s4144_2 60
s4295_5 61
s4261_1 62
sample
0
20
40
60
Intronic rate %
date_of
20180228
20180301
20180323

```
        ggtitle("Intronic mapping rate")
```

```
## $title
## [1] "Intronic mapping rate"
##
## attr(,"class")
## [1] "labels"
```

rRNA mapping rate

Samples should have a ribosomal RNA (rRNA) contamination rate below 10%.

```
metrics %>%
    ggplot(aes(x = reorder(sample, -r_rna_rate),
               y = r_rna_rate * 100,
               fill = date_of)) +
        geom_bar(stat = "identity") +
    geom_text(aes(label = round(r_rna_rate*100,2)), hjust = 0, nudge_y = 0.5)+
    xlab("sample") +
    ylab("rRNA rate, %")+
    ylim(0, 1) +
    ggtitle("rRNA mapping rate") +
    coord_flip()
```


rRNA mapping rate
sample
s3193_1
s3477_1
s4144_2
s3715_2
s3188_2
s3404_1
s4056_1
s3194_3
s3697_3
s3837_2
s3816_1
s3728_3
s4136_1
s3582_4
s4089_1
s3474_3
s3688_2
s4172_1
s4261_1
s3154_2
s4249_1
s3822_1
s3825_1
s4176_3
s3169_2
s3220_1
s3190_4
s4237_1
s4047_1
s3723_1
s3374_2
s3234_1
s4295_5
s4101_3
s3713_1
s3292_3
s3741_3
s3652_1
s3563_3
s3291_3
s3732_1
s3424_1
s3372_1
s3644_1
0.02
0.03
0.04
0.06
0.06
0.07
0.08
0.08
0.08
0.09
0.09
0.09
0.09
0.09
0.1
0.1
0.11
0.11
0.11
0.12
0.12
0.13
0.13
0.13
0.13
0.13
0.15
0.15
0.15
0.16
0.17
0.17
0.17
0.18
0.18
0.18
0.19
0.22
0.24
0.25
0.26
0.26
0.27
0.31
0.00
0.25
0.50
0.75
1.00
rRNA rate, %
date_of
20180228
20180301
20180323

5'->3' bias

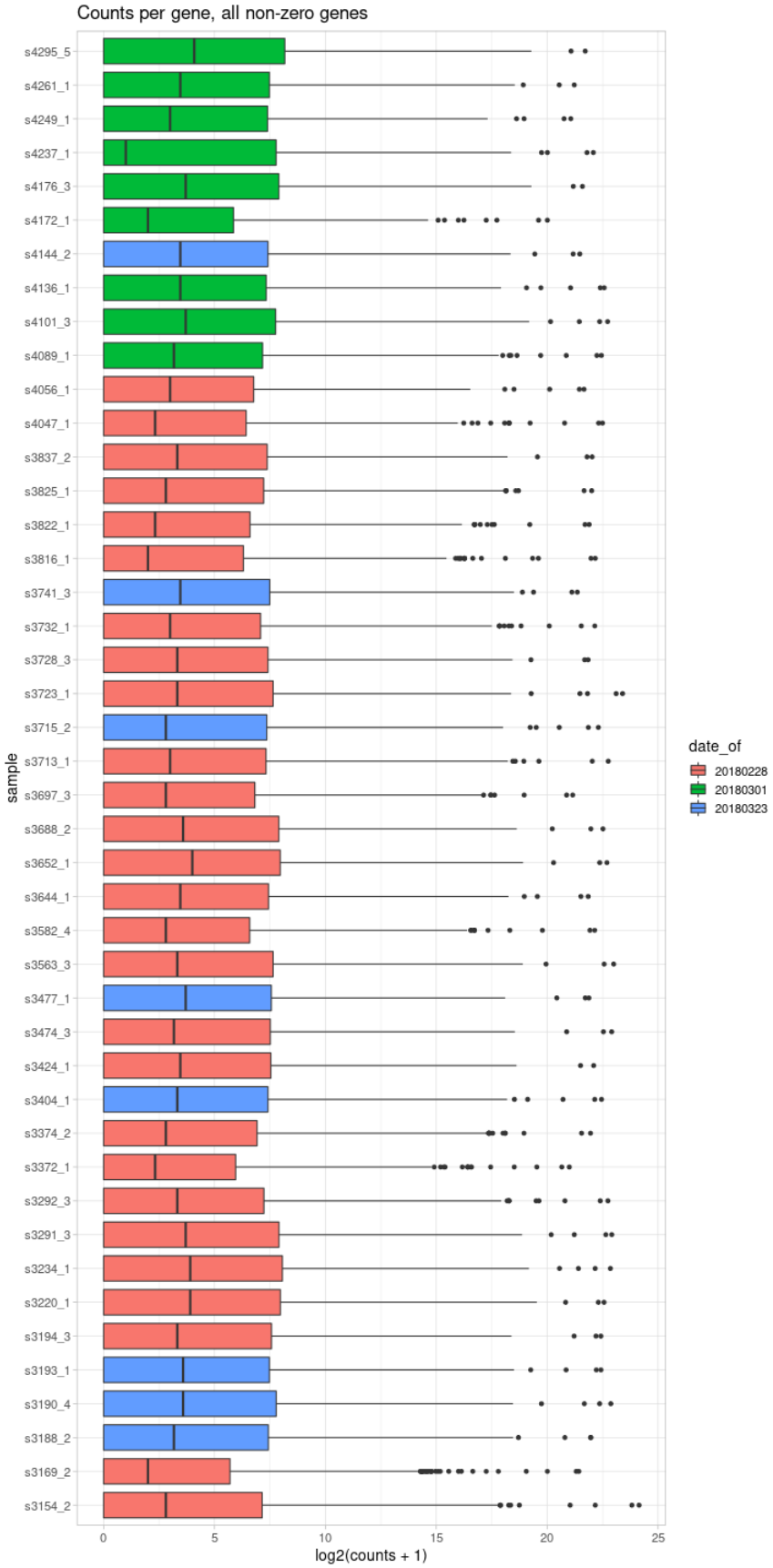
```
metrics %>%
    ggplot(aes(x = reorder(sample, -x5_3_bias),
               y = x5_3_bias,
               fill = date_of)) +
        geom_bar(stat = "identity") +
    geom_text(aes(label = x5_3_bias), hjust = 0, nudge_y = 0.01)+
    xlab("sample") +
    ylim(0, 1.5)+
    ggtitle("5'-3' bias") +
    coord_flip()
```


5'-3' bias
s4047_1
s3715_2
s3741_3
s3291_3
s3563_3
s3404_1
s3372_1
s3234_1
s3688_2
s3292_3
s4176_3
s4172_1
s4136_1
s3837_2
s3732_1
s3220_1
s3193_1
s4056_1
s3816_1
s3477_1
s3169_2
s3154_2
s4249_1
s3190_4
s3188_2
s4295_5
s3723_1
s4089_1
s3825_1
s3697_3
s3644_1
s4144_2
s3822_1
s3424_1
s3374_2
s3194_3
s4101_3
s3713_1
s3652_1
s4237_1
s3728_3
s3582_4
s4261_1
s3474_3
0.96
0.96
0.97
0.97
0.98
0.98
0.99
0.99
1
1
1.01
1.01
1.01
1.01
1.01
1.01
1.01
1.02
1.02
1.02
1.02
1.02
1.03
1.03
1.03
1.04
1.04
1.05
1.05
1.05
1.05
1.06
1.06
1.06
1.06
1.06
1.07
1.07
1.07
1.08
1.08
1.08
1.11
1.15
sample
0.0
0.5
1.0
1.5
x5_3_bias
date_of
20180228
20180301
20180323

Counts per gene - all genes

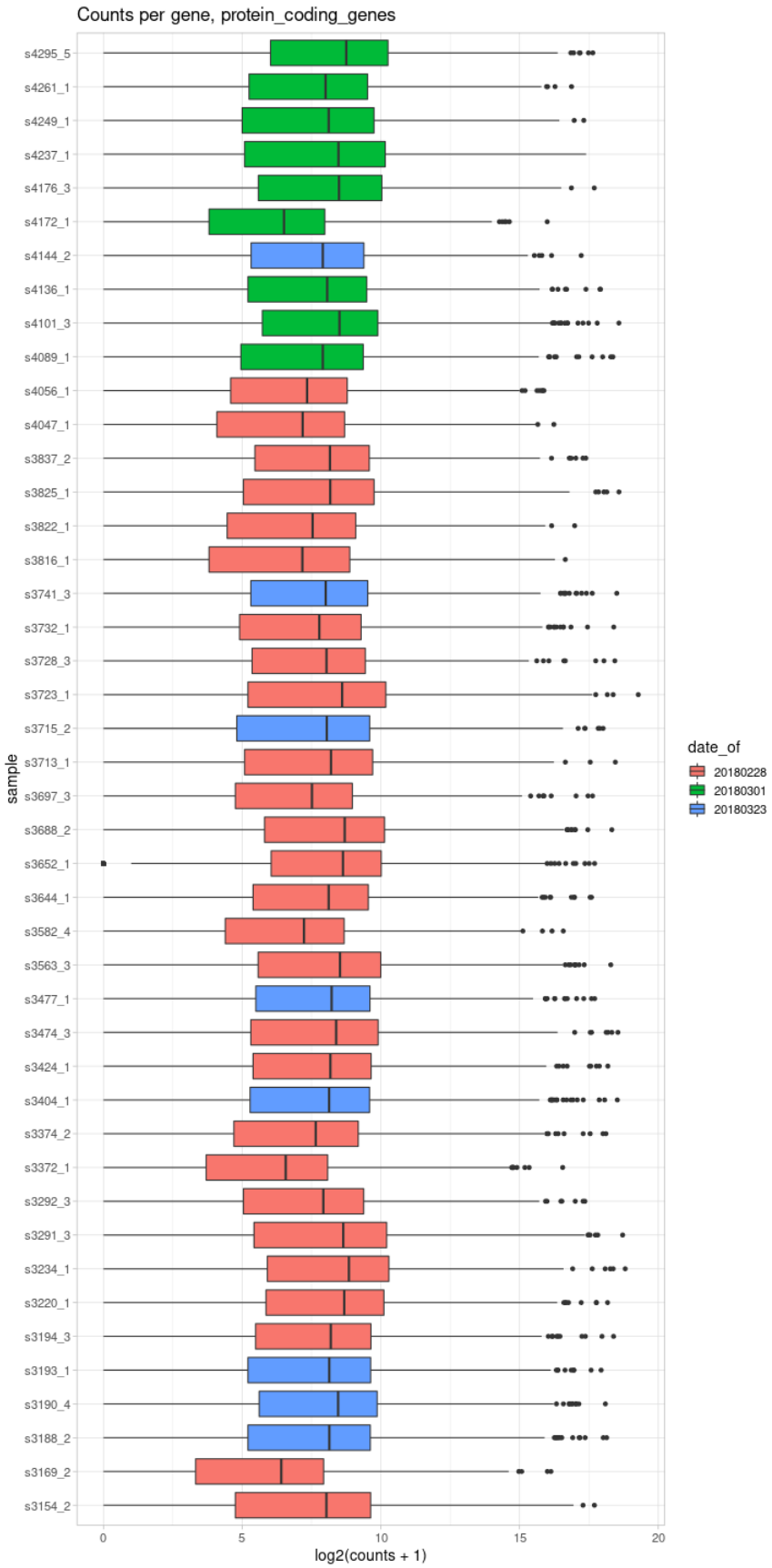
We expected similar spread for every sample.

```
metadata$date_of <- as_factor(metadata$date_of)
metrics_small <- metrics %>% dplyr::select(sample, date_of)
assays(se)[["raw"]] %>%
    as_tibble() %>%
    dplyr::filter(rowSums(.)!=0) %>%
    gather(sample, counts) %>%
    left_join(metadata, by = c("sample" = "sample")) %>%
    ggplot(aes(sample, log2(counts+1), fill = date_of)) +
    geom_boxplot() +
    coord_flip() +
    ggtitle("Counts per gene, all non-zero genes")
```


Counts per gene, all non-zero genes
sample
s4295_5
s4261_1
s4249_1
s4237_1
s4176_3
s4172_1
s4144_2
s4136_1
s4101_3
s4089_1
s4056_1
s4047_1
s3837_2
s3825_1
s3822_1
s3816_1
s3741_3
s3732_1
s3728_3
s3723_1
s3715_2
s3713_1
s3697_3
s3688_2
s3652_1
s3644_1
s3582_4
s3563_3
s3477_1
s3474_3
s3424_1
s3404_1
s3374_2
s3372_1
s3292_3
s3291_3
s3234_1
s3220_1
s3194_3
s3193_1
s3190_4
s3188_2
s3169_2
s3154_2
0
5
10
15
20
25
log2(counts + 1)
date_of
20180228
20180301
20180323

Counts per gene - protein coding genes

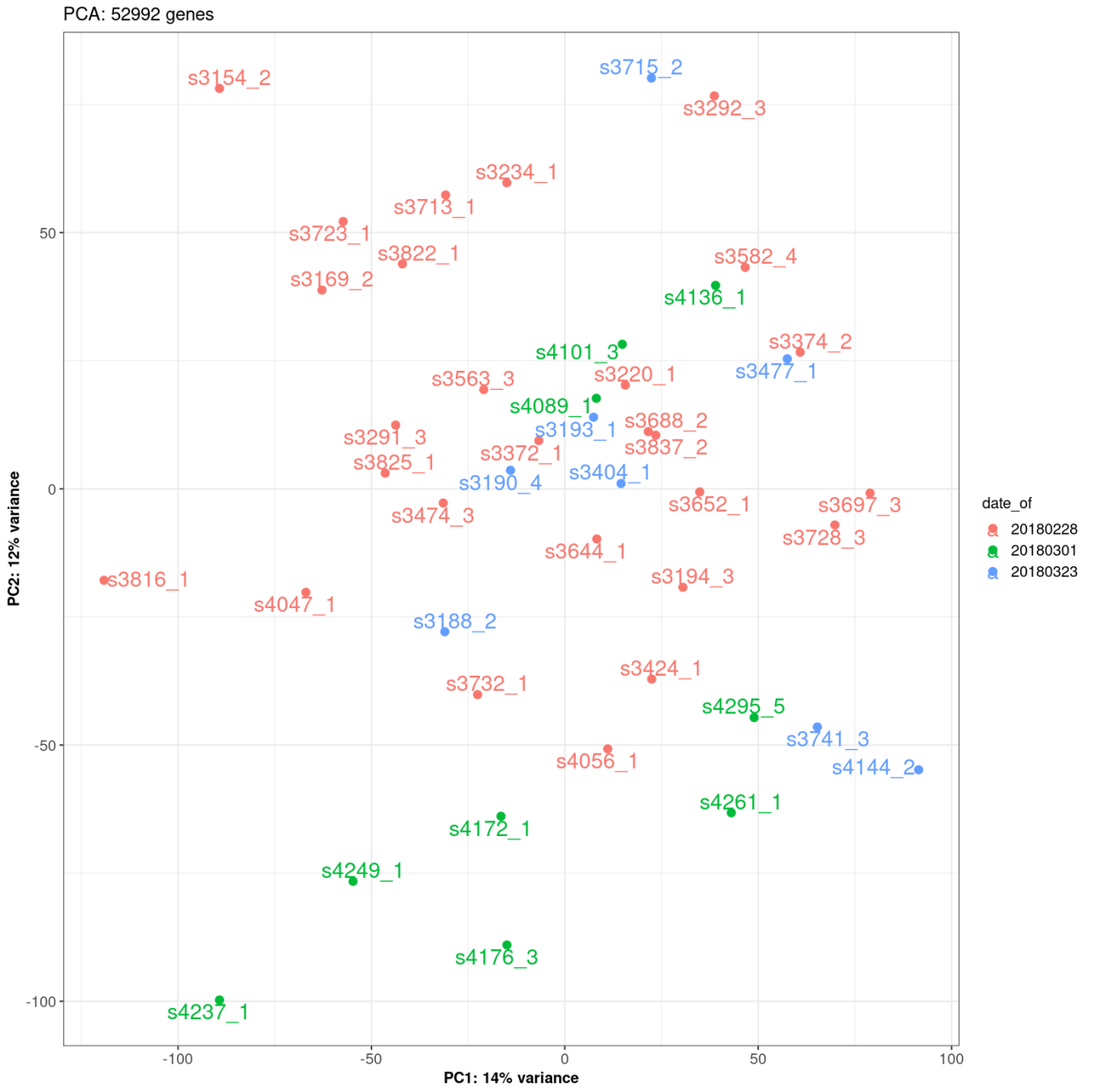
```
protein_coding_genes <- read_csv("tables/ensembl_w_description.protein_coding.csv")
assays(se)[["raw"]] %>%
    as_tibble(rownames = "ensembl_gene_id") %>%
    dplyr::filter(ensembl_gene_id %in% protein_coding_genes$ensembl_gene_id) %>%
    dplyr::select(-ensembl_gene_id) %>%
    dplyr::filter(rowSums(.)!=0) %>%
    gather(sample, counts) %>%
    left_join(metadata, by = c("sample" = "sample")) %>%
    dplyr::mutate(date_of = as.factor(date_of)) %>%
    ggplot(aes(sample, log2(counts+1), fill = date_of)) +
    geom_boxplot() +
    coord_flip() +
    ggtitle("Counts per gene, protein_coding_genes")
```


Counts per gene, protein_coding_genes
s4295_5
s4261_1
s4249_1
s4237_1
s4176_3
s4172_1
s4144_2
s4136_1
s4101_3
s4089_1
s4056_1
s4047_1
s3837_2
s3825_1
s3822_1
s3816_1
s3741_3
s3732_1
s3728_3
s3723_1
s3715_2
s3713_1
s3697_3
s3688_2
s3652_1
s3644_1
s3582_4
s3563_3
s3477_1
s3474_3
s3424_1
s3404_1
s3374_2
s3372_1
s3292_3
s3291_3
s3234_1
s3220_1
s3194_3
s3193_1
s3190_4
s3188_2
s3169_2
s3154_2
sample
0
5
10
15
20
log2(counts + 1)
date_of
20180228
20180301
20180323

Sample similarity analysis

Principal component analysis (PCA) - non zero genes

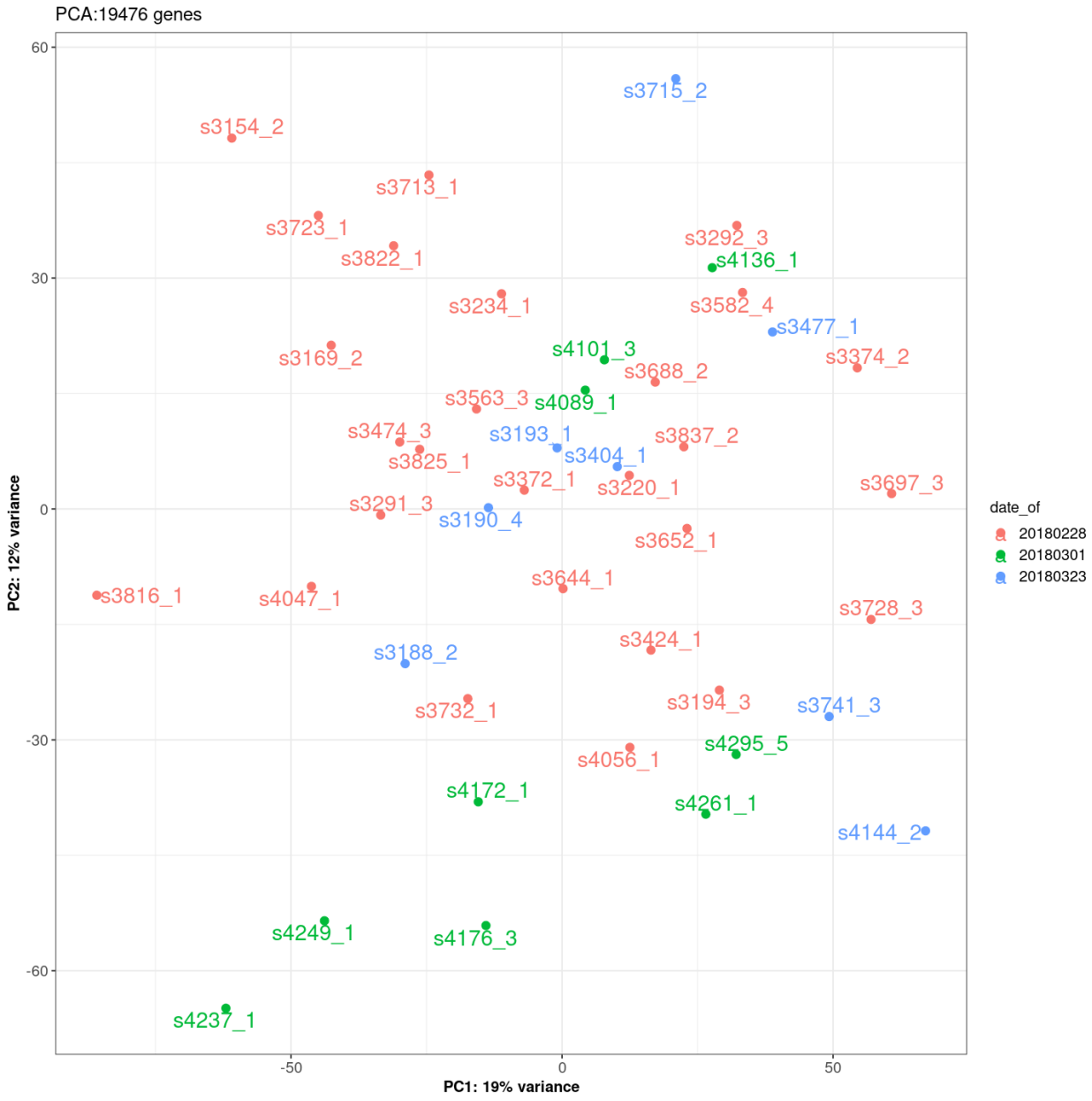
```
raw_counts <- assays(se)[["raw"]] %>%
    as_tibble() %>%
    dplyr::filter(rowSums(.)!=0) %>%
    as.matrix()

vst <- vst(raw_counts)

pca <- degPCA(vst, colData(se), condition = "date_of", name = "sample", data = T)[["plot"]]
pca_labels <- pca[["labels"]]
pca_data <- pca[["data"]] %>% as_tibble() %>%
    dplyr::select(sample, PC1, PC2, date_of)

pca_data$date_of <- as.factor(pca_data$date_of)

pca_data %>%
    ggplot(aes(x = PC1, y = PC2, color = date_of, label = sample)) +
    geom_point(size = 5) +
    geom_text_repel(size = 10) +
    xlab(pca_labels$x) +
    ylab(pca_labels$y) +
    ggtitle(paste0("PCA: ", nrow(vst), " genes")) +
    theme_bw(base_size = 20) +
    theme(axis.text = element_text(size = 20),
          axis.title = element_text(size = 20, face="bold"),
          legend.text = element_text(size = rel(1)))
```



Principal component analysis (PCA) - protein coding genes

```
raw_counts <- assays(se)[["raw"]] %>%
    as_tibble(rownames = "ensembl_gene_id") %>%
    dplyr::filter(ensembl_gene_id %in% protein_coding_genes$ensembl_gene_id) %>%
    column_to_rownames(var = "ensembl_gene_id") %>%
    dplyr::filter(rowSums(.)!=0) %>%
    as.matrix()

vst <- vst(raw_counts)

pca <- degPCA(vst, colData(se), condition = "date_of", name = "sample", data = T)[["plot"]]
pca_labels <- pca[["labels"]]
```

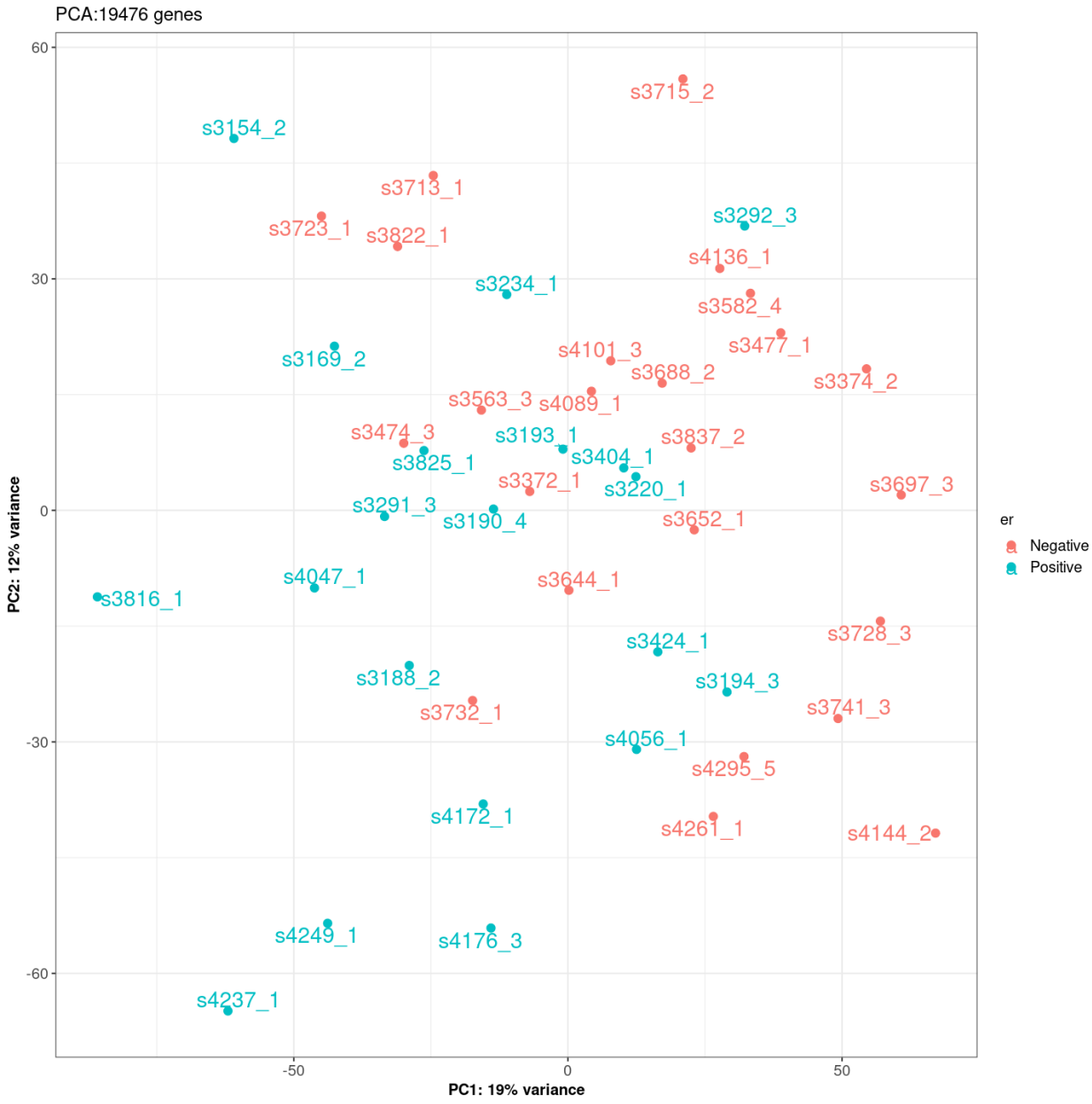
```r
pca_data <- pca[["data"]] %>% as_tibble() %>%
    dplyr::select(sample, PC1, PC2, date_of)

pca_data$date_of <- as.factor(pca_data$date_of)

pca_data %>%
    ggplot(aes(x = PC1, y = PC2, color = date_of, label = sample)) +
    geom_point(size = 5) +
    geom_text_repel(size = 10) +
    xlab(pca_labels$x) +
    ylab(pca_labels$y) +
    ggtitle(paste0("PCA:", nrow(vst), " genes")) +
    theme_bw(base_size = 20) +
    theme(axis.text = element_text(size = 20),
          axis.title = element_text(size = 20, face="bold"),
          legend.text = element_text(size = rel(1)))
```



Principal component analysis (PCA) - protein coding genes - er

```
raw_counts <- assays(se)[["raw"]] %>%
    as_tibble(rownames = "ensembl_gene_id") %>%
    dplyr::filter(ensembl_gene_id %in% protein_coding_genes$ensembl_gene_id) %>%
    column_to_rownames(var = "ensembl_gene_id") %>%
    dplyr::filter(rowSums(.)!=0) %>%
    as.matrix()

vst <- vst(raw_counts)

pca <- degPCA(vst, colData(se), condition = "er", name = "sample", data = T)[["plot"]]
pca_labels <- pca[["labels"]]
```

```r
pca_data <- pca[["data"]] %>% as_tibble() %>%
    dplyr::select(sample, PC1, PC2, er)

pca_data$er <- as.factor(pca_data$er)

pca_data %>%
    ggplot(aes(x = PC1, y = PC2, color = er, label = sample)) +
    geom_point(size = 5) +
    geom_text_repel(size = 10) +
    xlab(pca_labels$x) +
    ylab(pca_labels$y) +
    ggtitle(paste0("PCA:", nrow(vst), " genes")) +
    theme_bw(base_size = 20) +
    theme(axis.text = element_text(size = 20),
          axis.title = element_text(size = 20, face="bold"),
          legend.text = element_text(size = rel(1)))
```

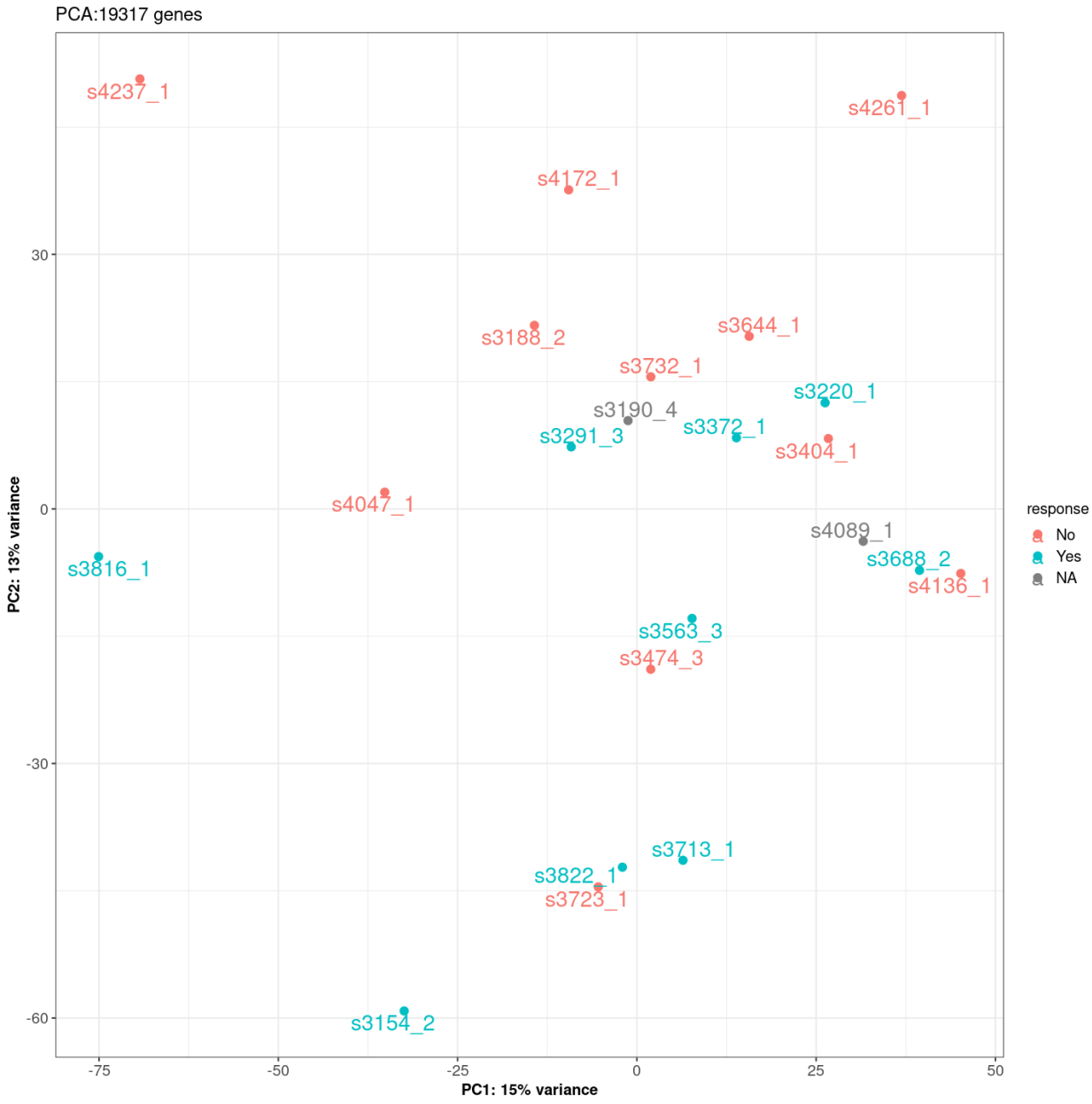


PCA - protein coding genes - PRE/POST - Responce

PRE

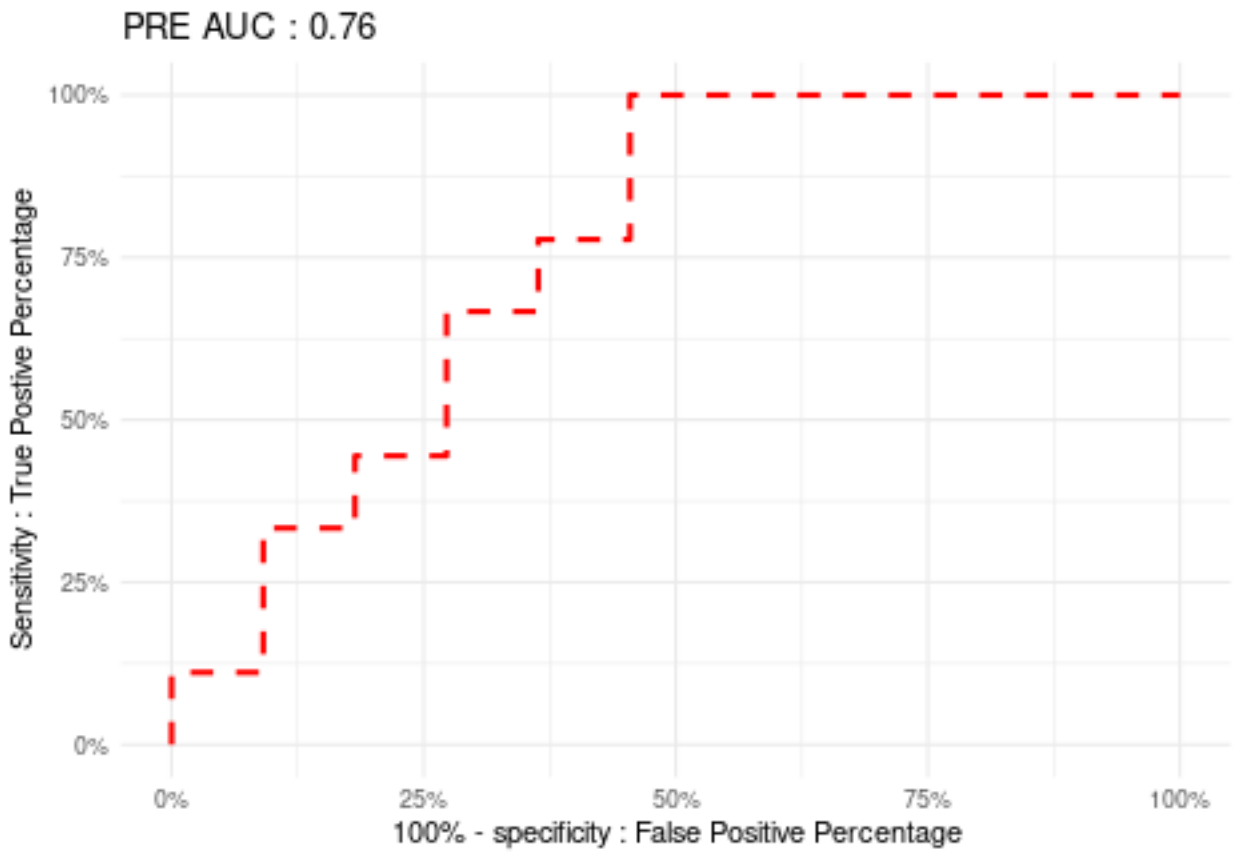
```
se_pre <- se[,se$treatment == "pre"]
raw_counts <- assays(se_pre)[["raw"]] %>%
    as_tibble(rownames = "ensembl_gene_id") %>%
    dplyr::filter(ensembl_gene_id %in% protein_coding_genes$ensembl_gene_id) %>%
    column_to_rownames(var = "ensembl_gene_id") %>%
    dplyr::filter(rowSums(.)!=0) %>%
    as.matrix()

vst <- vst(raw_counts)
```

```r
pca <- degPCA(vst, colData(se_pre), condition = "response", name = "sample", data = T)[["plot"]]
pca_labels <- pca[["labels"]]
pca_data_pre <- pca[["data"]] %>% as_tibble() %>%
    dplyr::select(sample, PC1, PC2, response)

pca_data_pre$response <- as.factor(pca_data_pre$response)

pca_data_pre %>%
    ggplot(aes(x = PC1, y = PC2, color = response, label = sample)) +
    geom_point(size = 5) +
    geom_text_repel(size = 10) +
    xlab(pca_labels$x) +
    ylab(pca_labels$y) +
    ggtitle(paste0("PCA:", nrow(vst), " genes")) +
    theme_bw(base_size = 20) +
    theme(axis.text = element_text(size = 20),
          axis.title = element_text(size = 20, face="bold"),
          legend.text = element_text(size = rel(1)))
```



PRE - ROC

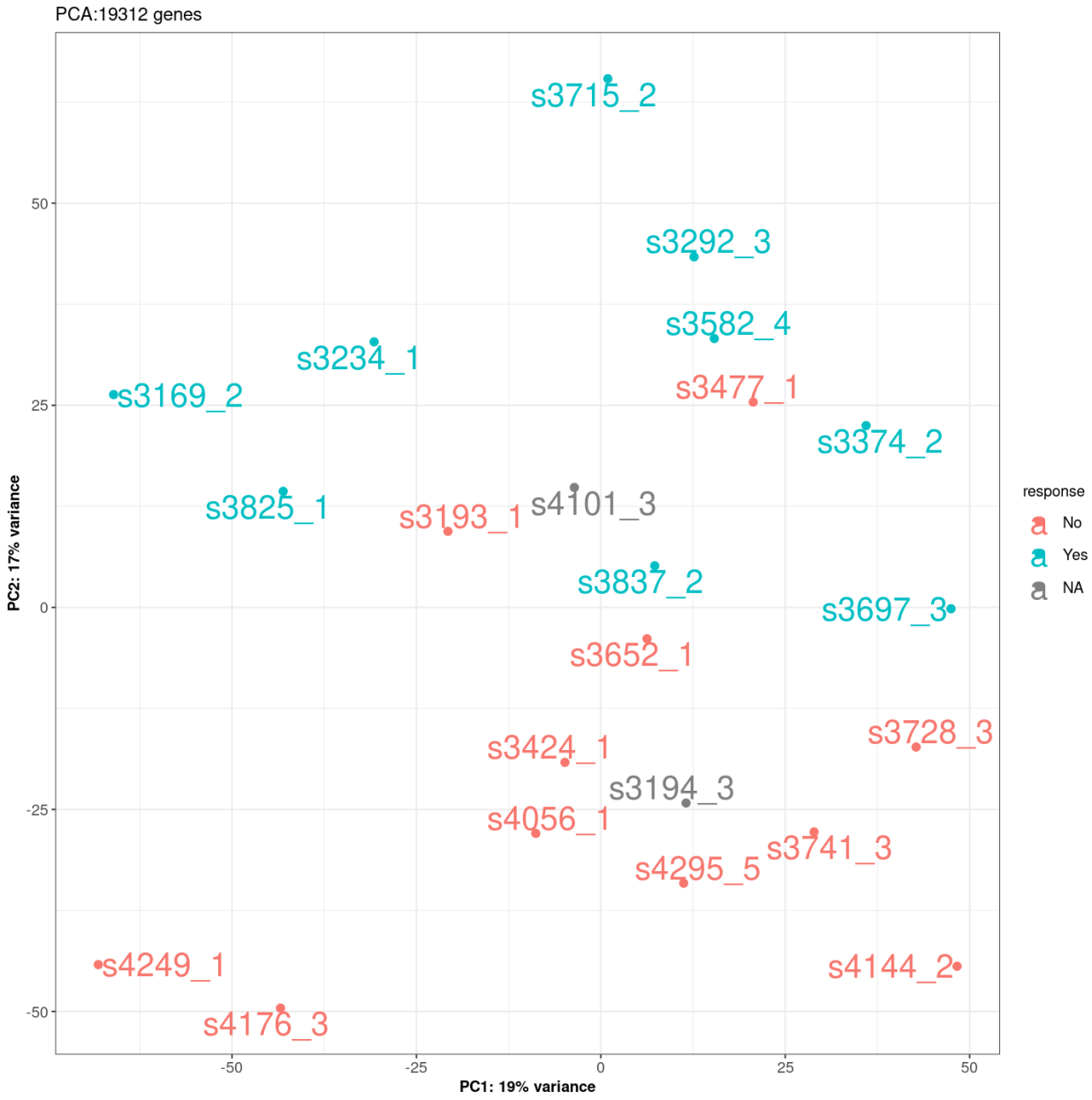
```
#pca_data$PC2
#pca_data$response
roc_pre <- roc(pca_data_pre$response, pca_data_pre$PC2, percent = FALSE)
ggroc(roc_pre,
      colour = "red", linetype = 2, size = 1, legacy.axes = TRUE)+
    theme_minimal() +
  scale_x_continuous(labels = scales::percent) +
  scale_y_continuous(labels = scales::percent) +
  xlab("100% - specificity : False Positive Percentage") +
  ylab("Sensitivity : True Postive Percentage") +
  ggtitle(paste0("PRE AUC : ", round(roc_pre$auc, 2)))
```



POST

```
se_post <- se[,se$treatment == "post"]
raw_counts <- assays(se_post)[["raw"]] %>%
    as_tibble(rownames = "ensembl_gene_id") %>%
    dplyr::filter(ensembl_gene_id %in% protein_coding_genes$ensembl_gene_id) %>%
    column_to_rownames(var = "ensembl_gene_id") %>%
    dplyr::filter(rowSums(.)!=0) %>%
    as.matrix()

vst <- vst(raw_counts)

pca_post <- degPCA(vst, colData(se_post), condition = "response", name = "sample", data = T)[["plot"]]
pca_labels <- pca_post[["labels"]]
pca_data_post <- pca_post[["data"]] %>% as_tibble() %>%
    dplyr::select(sample, PC1, PC2, response)

pca_data_post$response <- as.factor(pca_data_post$response)

pca_data_post %>%
    ggplot(aes(x = PC1, y = PC2, color = response, label = sample)) +
    geom_point(size = 5) +
    geom_text_repel(size = 15) +
    xlab(pca_labels$x) +
    ylab(pca_labels$y) +
```

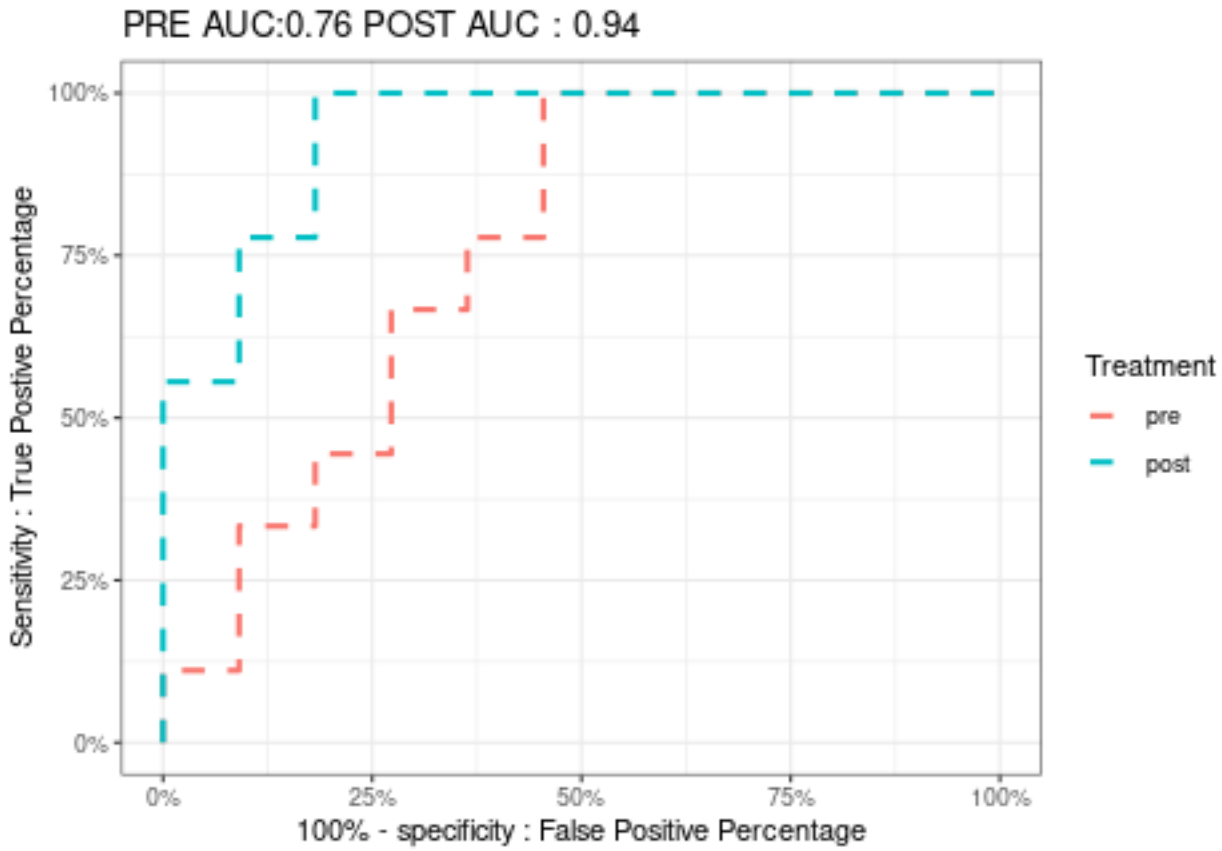
```
    ggtitle(paste0("PCA:", nrow(vst), " genes")) +
    theme_bw(base_size = 20) +
    theme(axis.text = element_text(size = 20),
          axis.title = element_text(size = 20, face="bold"),
          legend.text = element_text(size = rel(1)))
```



POST - ROC

```
#pca_data_post$PC2
#pca_data_post$response
# https://sachsmc.github.io/plotROC/
roc_post <- roc(pca_data_post$response, pca_data_post$PC2, percent = FALSE)
```

```
pROC::ggroc(list(pre = roc_pre,
           post = roc_post),
           legacy.axes = TRUE, size = 1, linetype = 2) +
  scale_x_continuous(labels = scales::percent) +
  scale_y_continuous(labels = scales::percent) +
  xlab("100% - specificity : False Positive Percentage") +
  ylab("Sensitivity : True Postive Percentage") +
  theme_bw() +
  ggtitle(paste0("PRE AUC:", round(roc_pre$auc, 2), " POST AUC : ", round(roc_post$auc, 2))) +
  guides(color = guide_legend(title="Treatment"))
```



R session

```
sessionInfo()
```

```
## R version 4.0.3 (2020-10-10)
## Platform: x86_64-redhat-linux-gnu (64-bit)
## Running under: Fedora 32 (Workstation Edition)
##
## Matrix products: default
## BLAS/LAPACK: /usr/lib64/libopenblas-r0.3.12.so
##
## locale:
##  [1] LC_CTYPE=en_CA.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_CA.UTF-8        LC_COLLATE=en_CA.UTF-8
```

```
##  [5] LC_MONETARY=en_CA.UTF-8    LC_MESSAGES=en_CA.UTF-8
##  [7] LC_PAPER=en_CA.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_CA.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] parallel  stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] randomForest_4.6-14         pROC_1.17.0.1
##  [3] ggrepel_0.9.1               DEGreport_1.26.0
##  [5] DESeq2_1.30.1               SummarizedExperiment_1.20.0
##  [7] Biobase_2.50.0              MatrixGenerics_1.2.1
##  [9] matrixStats_0.58.0          GenomicRanges_1.42.0
## [11] GenomeInfoDb_1.26.2         IRanges_2.24.1
## [13] S4Vectors_0.28.1            BiocGenerics_0.36.0
## [15] knitr_1.30                  forcats_0.5.1
## [17] stringr_1.4.0               dplyr_1.0.5
## [19] purrr_0.3.4                 readr_1.4.0
## [21] tidyr_1.1.3                 tibble_3.1.0
## [23] ggplot2_3.3.3               tidyverse_1.3.0
##
## loaded via a namespace (and not attached):
##   [1] colorspace_2.0-0            rjson_0.2.20
##   [3] ellipsis_0.3.1              circlize_0.4.12
##   [5] XVector_0.30.0              ggdendro_0.1.22
##   [7] GlobalOptions_0.1.2         fs_1.5.0
##   [9] clue_0.3-58                 rstudioapi_0.13
##  [11] farver_2.1.0                bit64_4.0.5
##  [13] AnnotationDbi_1.52.0        fansi_0.4.2
##  [15] lubridate_1.7.10            xml2_1.3.2
##  [17] splines_4.0.3               logging_0.10-108
##  [19] mnormt_2.0.2                cachem_1.0.4
##  [21] geneplotter_1.68.0          jsonlite_1.7.1
##  [23] Nozzle.R1_1.1-1             Cairo_1.5-12.2
##  [25] broom_0.7.5                 annotate_1.68.0
##  [27] cluster_2.1.0               dbplyr_2.1.0
##  [29] png_0.1-7                   compiler_4.0.3
##  [31] httr_1.4.2                  backports_1.2.1
##  [33] assertthat_0.2.1            Matrix_1.2-18
##  [35] fastmap_1.1.0               limma_3.46.0
##  [37] cli_2.3.1                   lasso2_1.2-21.1
##  [39] htmltools_0.5.1.1           tools_4.0.3
##  [41] gtable_0.3.0                glue_1.4.2
##  [43] GenomeInfoDbData_1.2.4      Rcpp_1.0.6
##  [45] cellranger_1.1.0            vctrs_0.3.6
##  [47] nlme_3.1-149                psych_2.0.12
##  [49] xfun_0.19                   rvest_1.0.0
##  [51] lifecycle_1.0.0             XML_3.99-0.5
##  [53] edgeR_3.32.1                MASS_7.3-53
##  [55] zlibbioc_1.36.0             scales_1.1.1
##  [57] hms_1.0.0                   RColorBrewer_1.1-2
##  [59] ComplexHeatmap_2.6.2        yaml_2.2.1
```

```
##  [61] memoise_2.0.0              reshape_0.8.8
##  [63] stringi_1.5.3              RSQLite_2.2.3
##  [65] genefilter_1.72.1          BiocParallel_1.24.1
##  [67] shape_1.4.5                rlang_0.4.10
##  [69] pkgconfig_2.0.3            bitops_1.0-6
##  [71] evaluate_0.14              lattice_0.20-41
##  [73] labeling_0.4.2             cowplot_1.1.1
##  [75] bit_4.0.4                  tidyselect_1.1.0
##  [77] plyr_1.8.6                 magrittr_2.0.1
##  [79] R6_2.5.0                   generics_0.1.0
##  [81] DelayedArray_0.16.2        DBI_1.1.1
##  [83] pillar_1.5.1               haven_2.3.1
##  [85] withr_2.4.1                survival_3.2-7
##  [87] RCurl_1.98-1.2             modelr_0.1.8
##  [89] crayon_1.4.1               utf8_1.1.4
##  [91] tmvnsim_1.0-2              rmarkdown_2.5
##  [93] GetoptLong_1.0.5           locfit_1.5-9.4
##  [95] grid_4.0.3                 readxl_1.3.1
##  [97] blob_1.2.1                 ConsensusClusterPlus_1.54.0
##  [99] reprex_1.0.0               digest_0.6.27
## [101] xtable_1.8-4               munsell_0.5.0
```