Quality Control

Sergey Naumenko

2021-07-01

Contents

Overview

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

Metadata

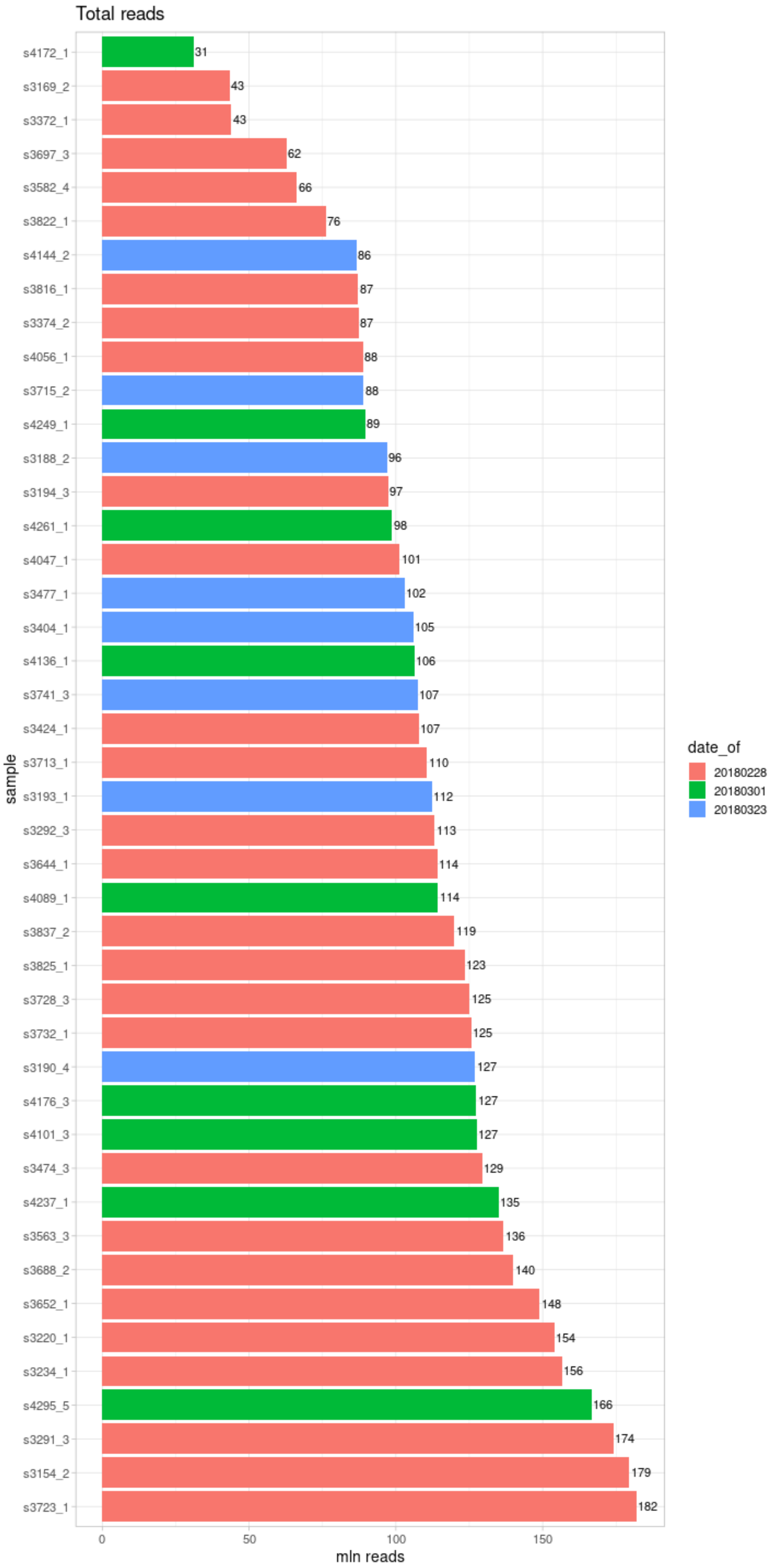
```
se <- readRDS("data/bcbio-se_all_samples.rds")
se <- se[, se$study_id != 6]
mdata <- colData(se)
mdata <- subset(mdata, subset = study_id != 6)
mdata$response <- str_replace(mdata$response, "Yes", "pCR")
mdata$response <- str_replace(mdata$response, "No", "non-pCR")
mdata$response <- replace_na(mdata$response, "non-pCR")
colData(se) <- mdata

mdata <- metadata(se)
mdata$metrics <- mdata$metrics %>% dplyr::filter(!sample %in% c("s3370_1", "s3373_3"))
metadata(se) <- mdata

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~ pCR             1 pre       30-100           s3154~
##  2 post     20180228 Positi~ pCR             1 post      30-100           s3169~
##  3 pre      20180323 Positi~ non-pCR         3 pre       1-29             s3188~
##  4 pre      20180323 Positi~ non-pCR         2 pre       30-100           s3190~
##  5 post     20180323 Positi~ non-pCR         3 post      1-29             s3193~
##  6 post     20180228 Positi~ non-pCR         2 post      1-29             s3194~
##  7 pre      20180228 Positi~ pCR             4 pre       30-100           s3220~
##  8 post     20180228 Positi~ pCR             4 post      30-100           s3234~
##  9 pre      20180228 Positi~ pCR             5 pre       1-29             s3291~
## 10 post     20180228 Positi~ pCR             5 post      1-29             s3292~
## # ... with 34 more rows
```

Read metrics

Total reads

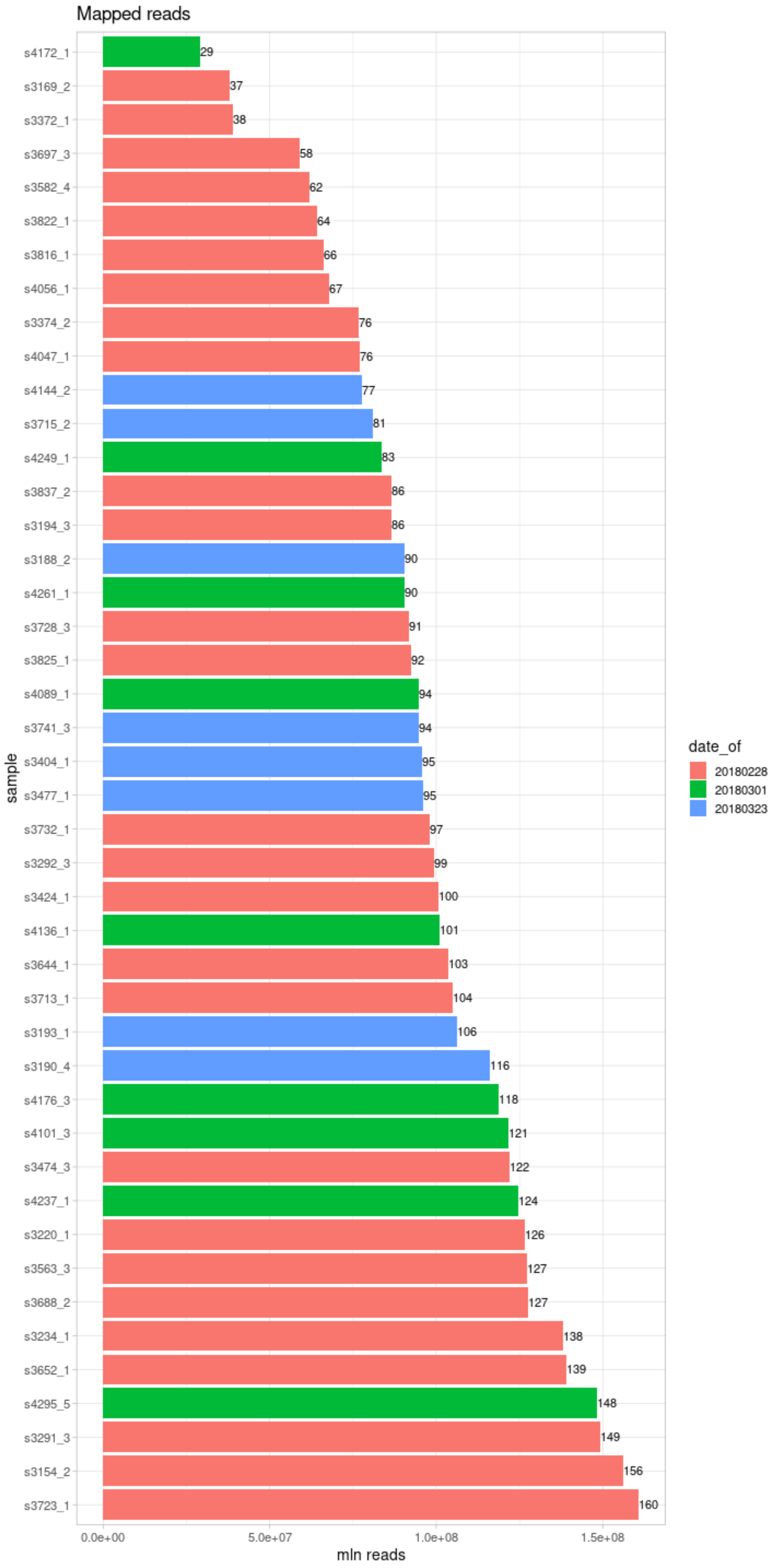
```r
metrics <- metrics %>%
    group_by(date_of) %>%
    mutate(position = rank(-total_reads))

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
sample
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
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.

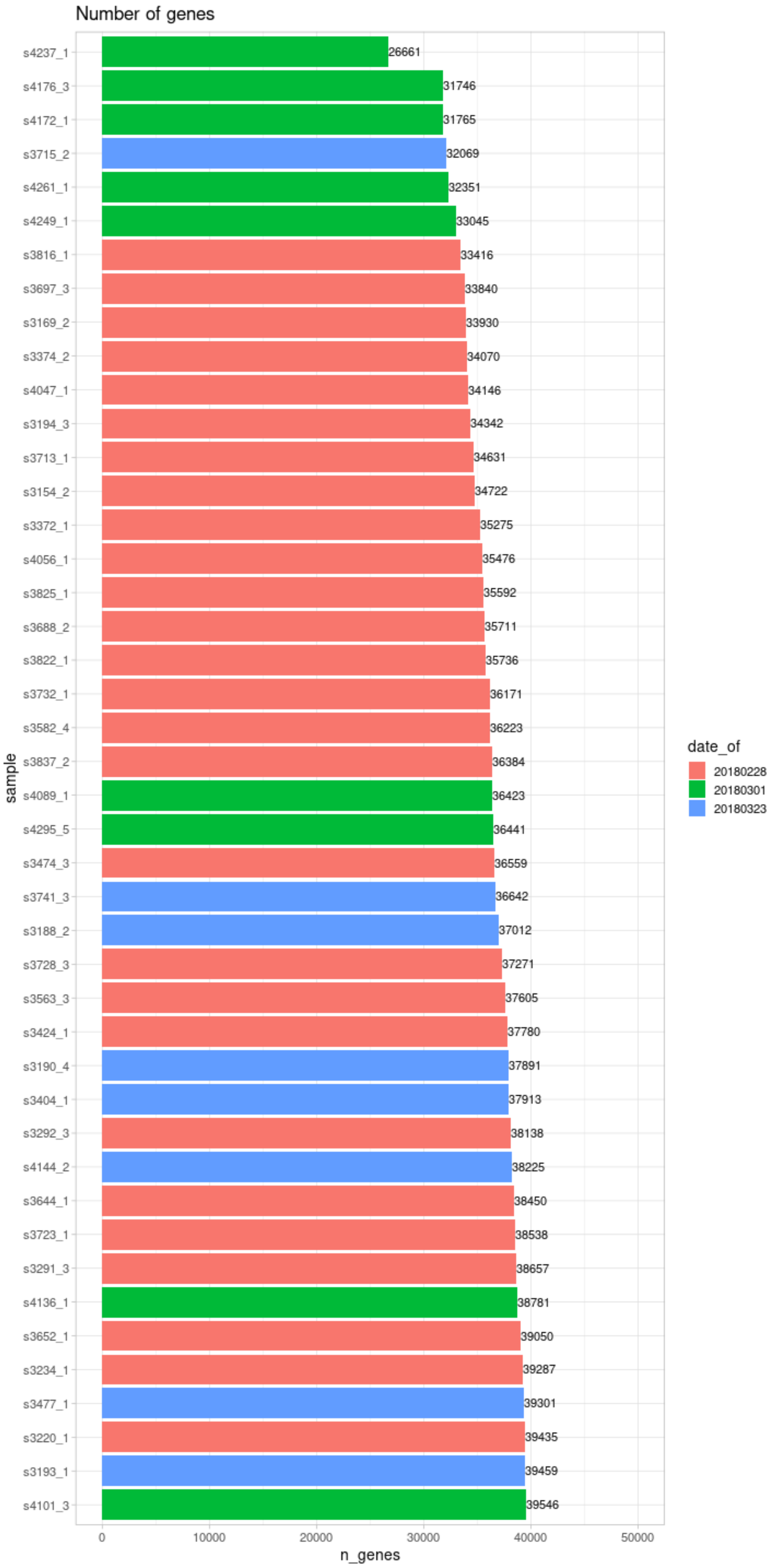
```r
metrics %>%
    ggplot(aes(x = reorder(sample, -mapped_reads),
               y = mapped_reads, fill = date_of)) +
    geom_bar(stat = "identity") +
    coord_flip() +
    scale_y_continuous(name = "mln reads") +
    geom_text(aes(label = floor(mapped_reads/1e6L)), hjust = 0, nudge_y = 0.5)+
  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


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

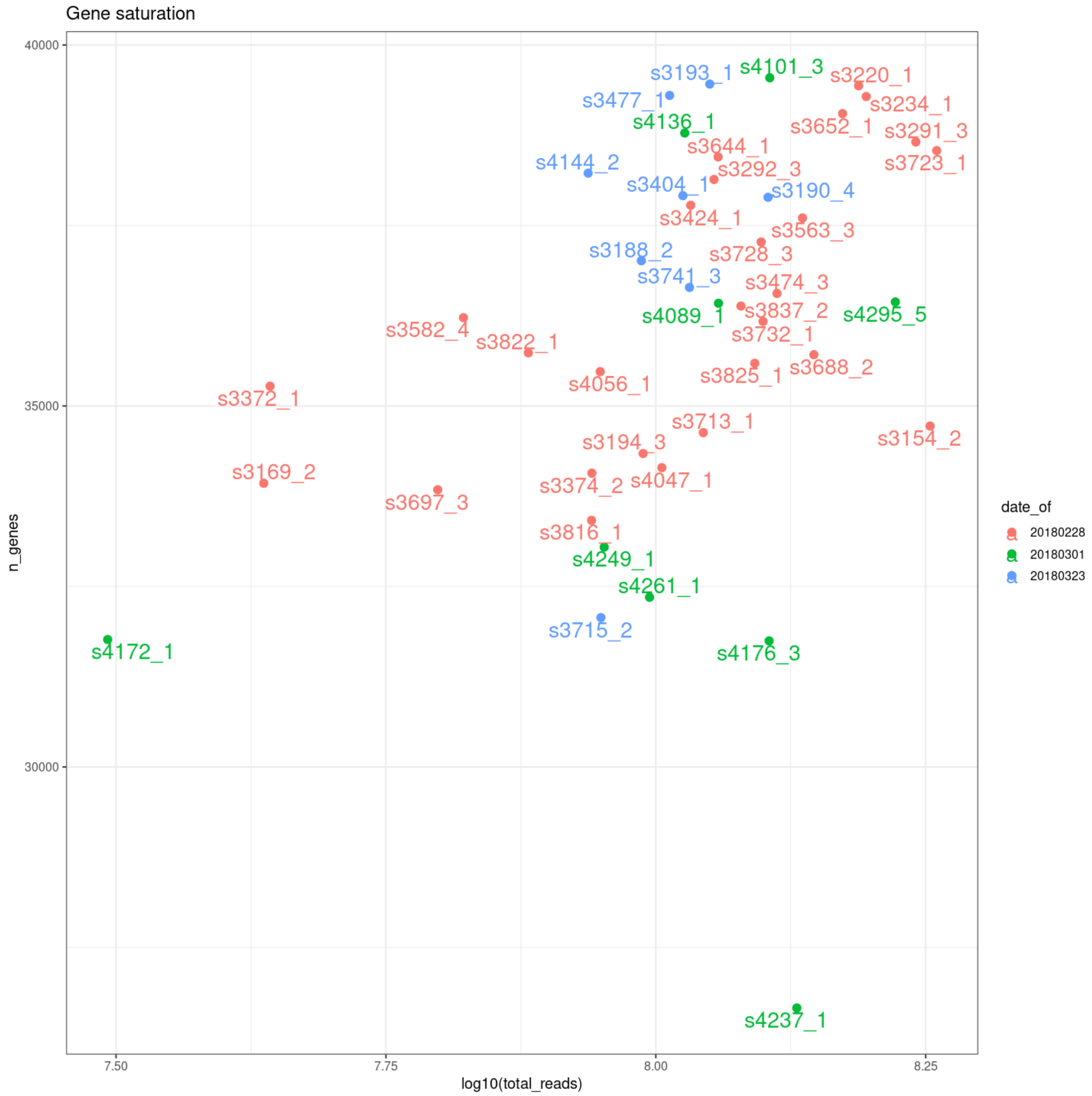
```
genes_detected <- colSums(assays(se)[["raw"]] > 0) %>% enframe()
colnames(genes_detected) <- c("sample", "n_genes")

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
s4237_1 26661
s4176_3 31746
s4172_1 31765
s3715_2 32069
s4261_1 32351
s4249_1 33045
s3816_1 33416
s3697_3 33840
s3169_2 33930
s3374_2 34070
s4047_1 34146
s3194_3 34342
s3713_1 34631
s3154_2 34722
s3372_1 35275
s4056_1 35476
s3825_1 35592
s3688_2 35711
s3822_1 35736
s3732_1 36171
s3582_4 36223
s3837_2 36384
s4089_1 36423
s4295_5 36441
s3474_3 36559
s3741_3 36642
s3188_2 37012
s3728_3 37271
s3563_3 37605
s3424_1 37780
s3190_4 37891
s3404_1 37913
s3292_3 38138
s4144_2 38225
s3644_1 38450
s3723_1 38538
s3291_3 38657
s4136_1 38781
s3652_1 39050
s3234_1 39287
s3477_1 39301
s3220_1 39435
s3193_1 39459
s4101_3 39546
sample
0
10000
20000
30000
40000
50000
n_genes
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.

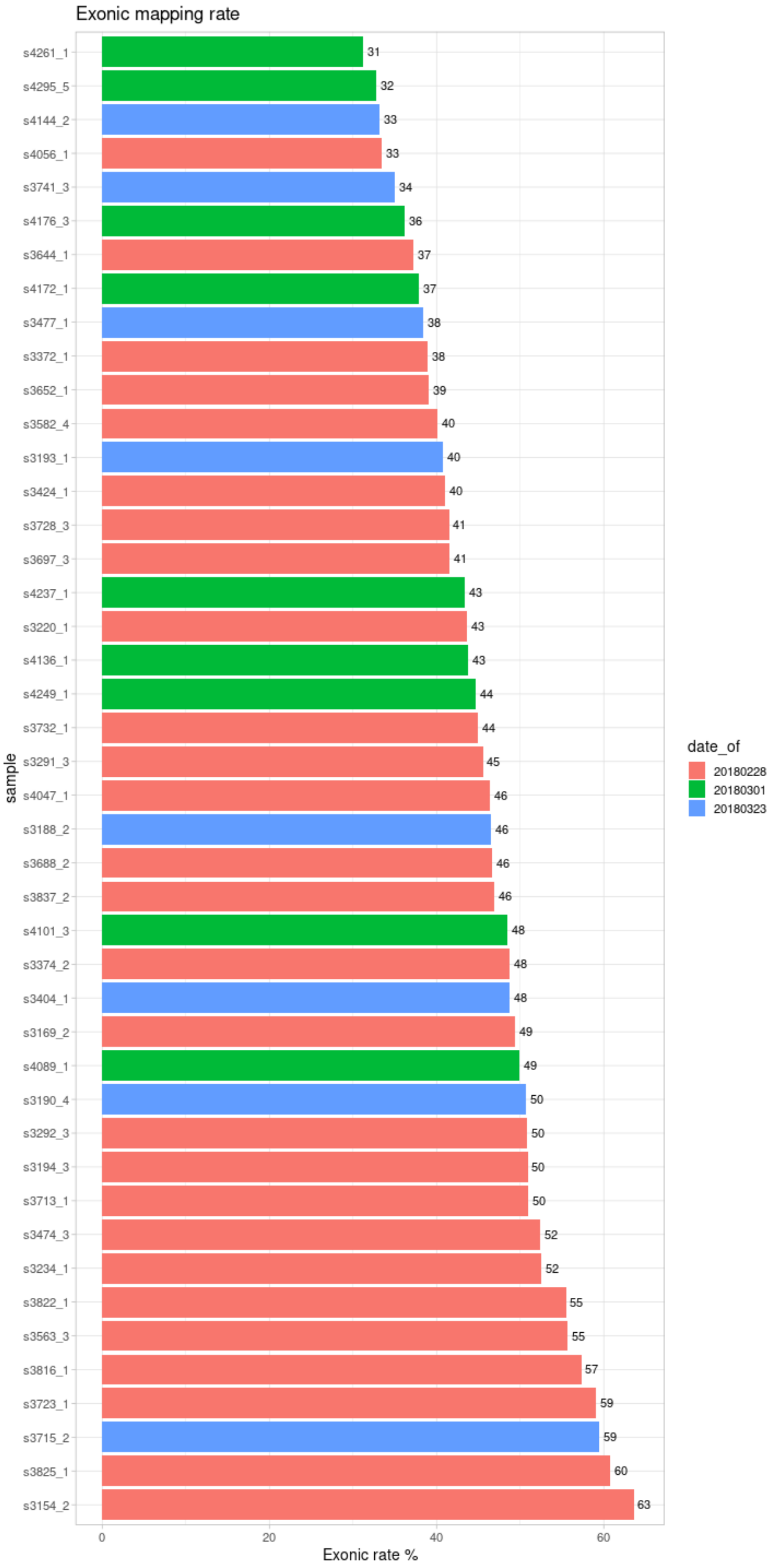
```
metrics %>%
    ggplot(aes(x = log10(total_reads),
               y = n_genes,
               color = date_of)) +
        geom_point(size = 5)+
        geom_text_repel(aes(label = sample), size = 10)+
    theme(axis.text = element_text(size = 20),
          axis.title = element_text(size = 20, face="bold"),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

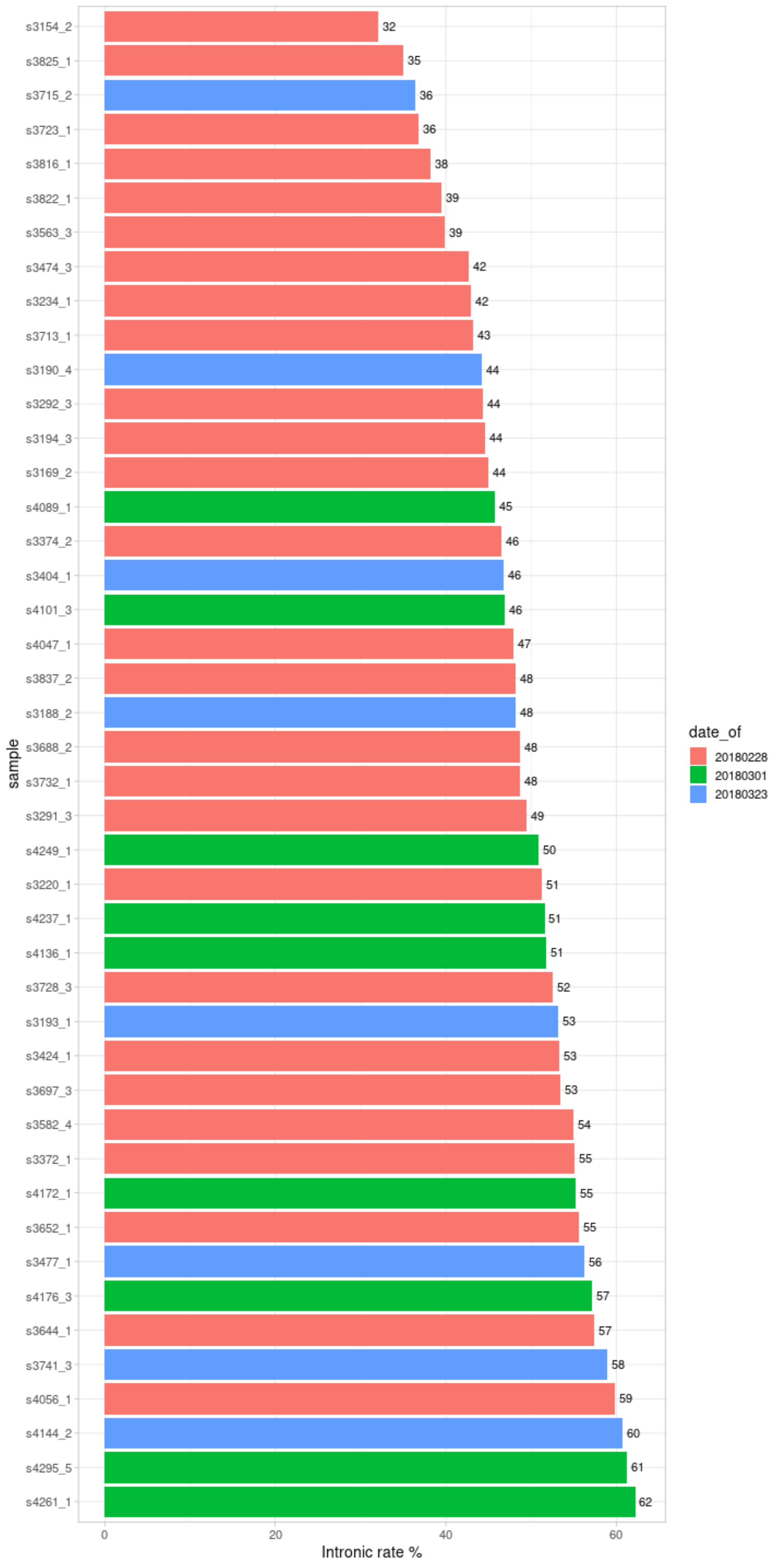
```
metrics %>%
    ggplot(aes(x = reorder(sample, -exonic_rate),
               y = exonic_rate * 100,
               fill = date_of)) +
        geom_bar(stat = "identity") +
    geom_text(aes(label = floor(exonic_rate*100)), hjust = 0, nudge_y = 0.5)+
    xlab("sample") +
    ylab("Exonic rate %")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


sample
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
0
20
40
60
Intronic rate %
date_of
20180228
20180301
20180323

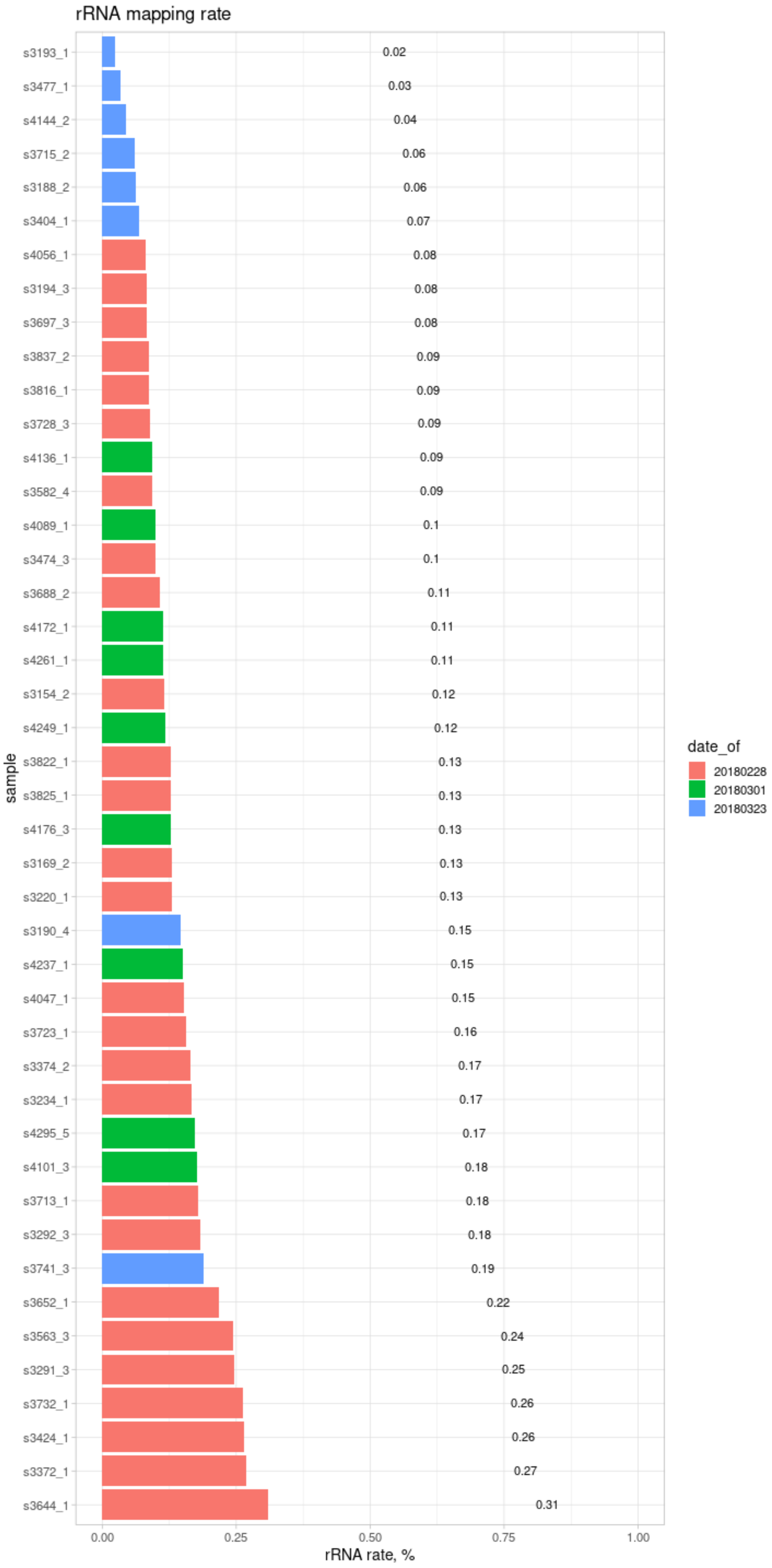
```
        ggtitle("Intronic mapping rate")
```

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

rRNA mapping rate

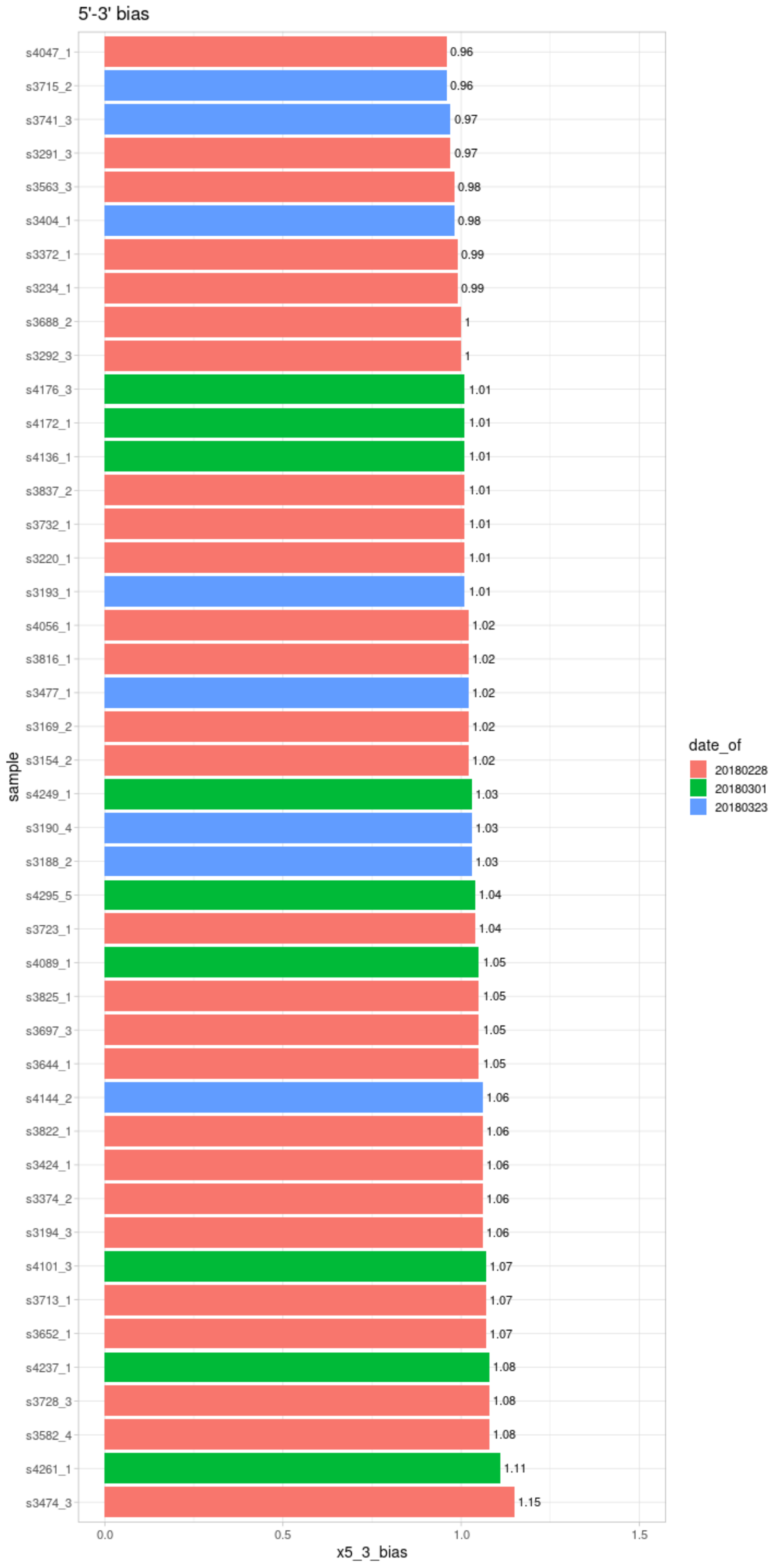
Samples should have a ribosomal RNA (rRNA) contamination rate below 10%.

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

5’->3’ bias

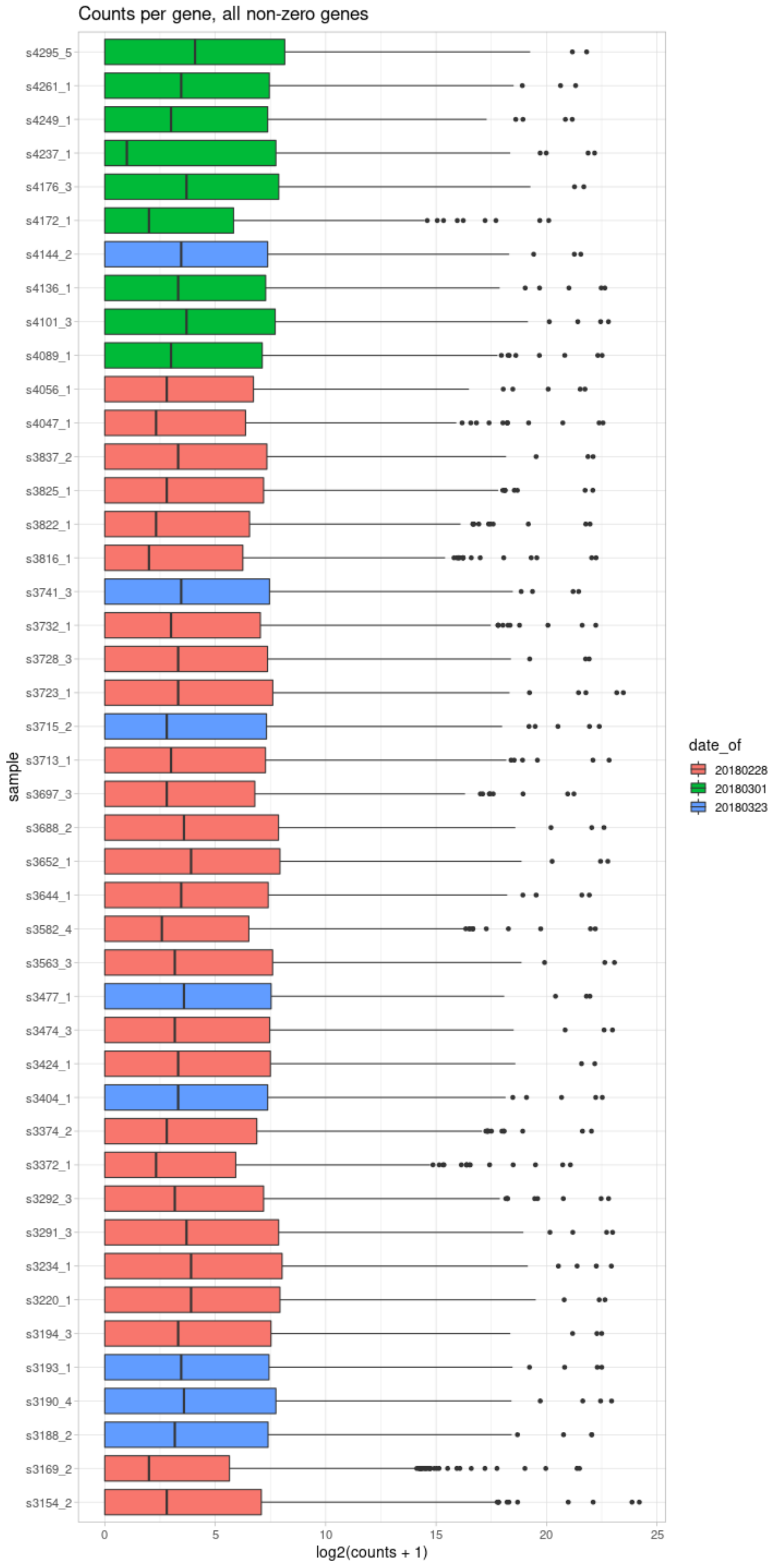
```
metrics %>%
    ggplot(aes(x = reorder(sample, -x5_3_bias),
               y = x5_3_bias,
               fill = date_of)) +
        geom_bar(stat = "identity") +
    geom_text(aes(label = x5_3_bias), hjust = 0, nudge_y = 0.01)+
    xlab("sample") +
    ylim(0, 1.5)+
    ggtitle("5'-3' bias") +
 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


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

Counts per gene - all genes

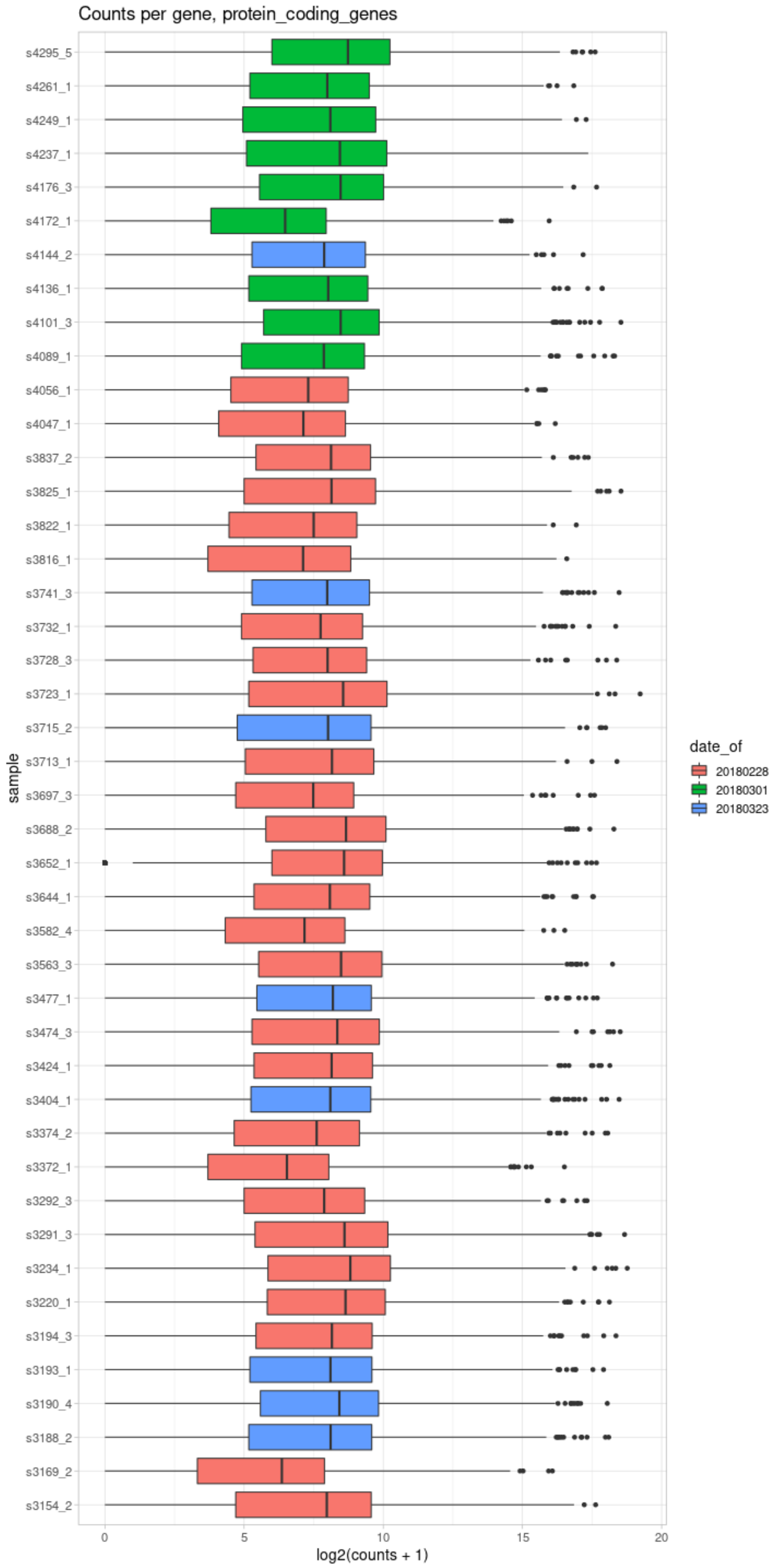
We expected similar spread for every sample.

```r
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

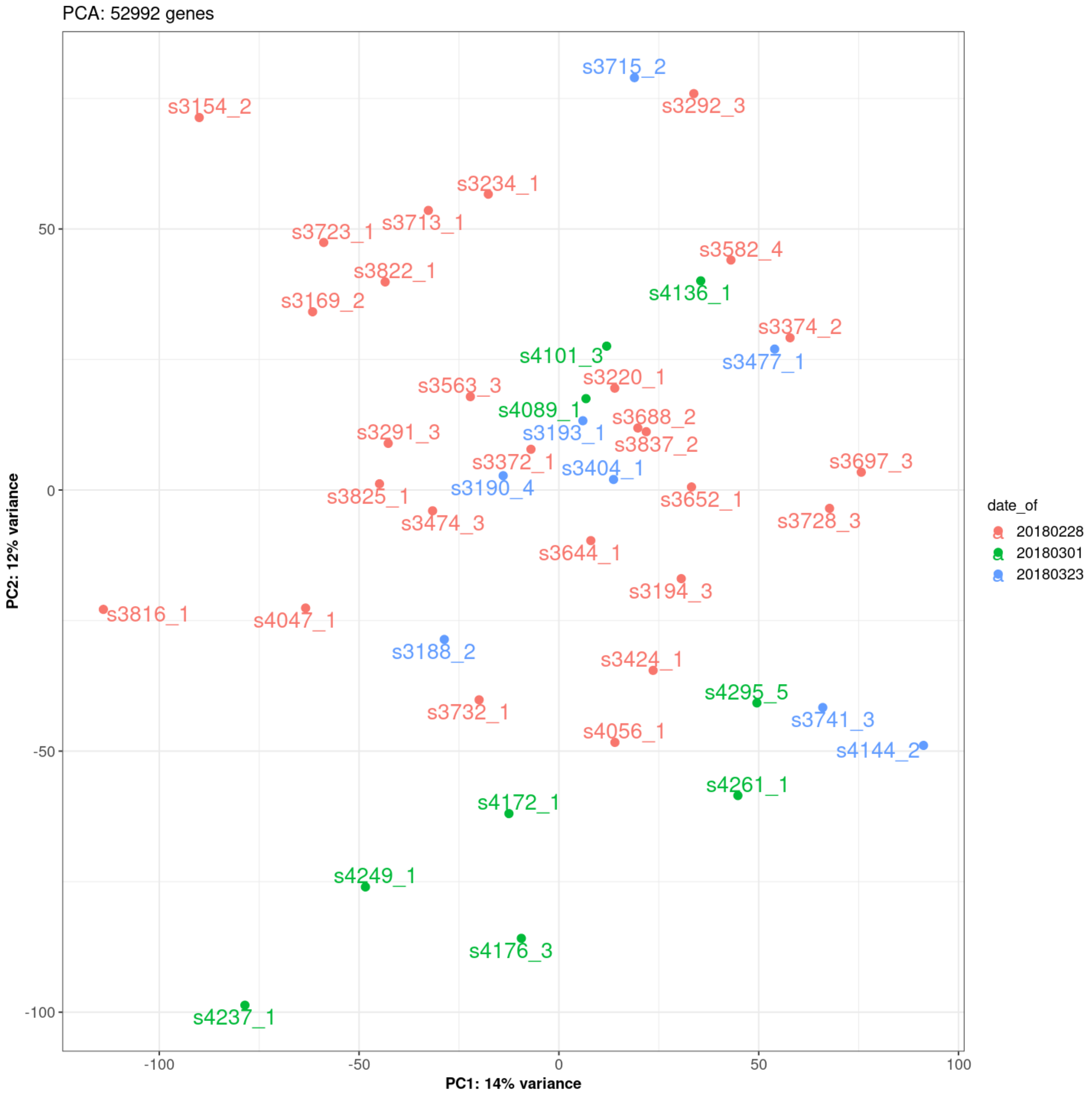
```r
protein_coding_genes <- read_csv("tables/ensembl_w_description.protein_coding.csv")
assays(se)[["raw"]] %>%
    as_tibble(rownames = "ensembl_gene_id") %>%
    dplyr::filter(ensembl_gene_id %in% protein_coding_genes$ensembl_gene_id) %>%
    dplyr::select(-ensembl_gene_id) %>%
    dplyr::filter(rowSums(.)!=0) %>%
    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

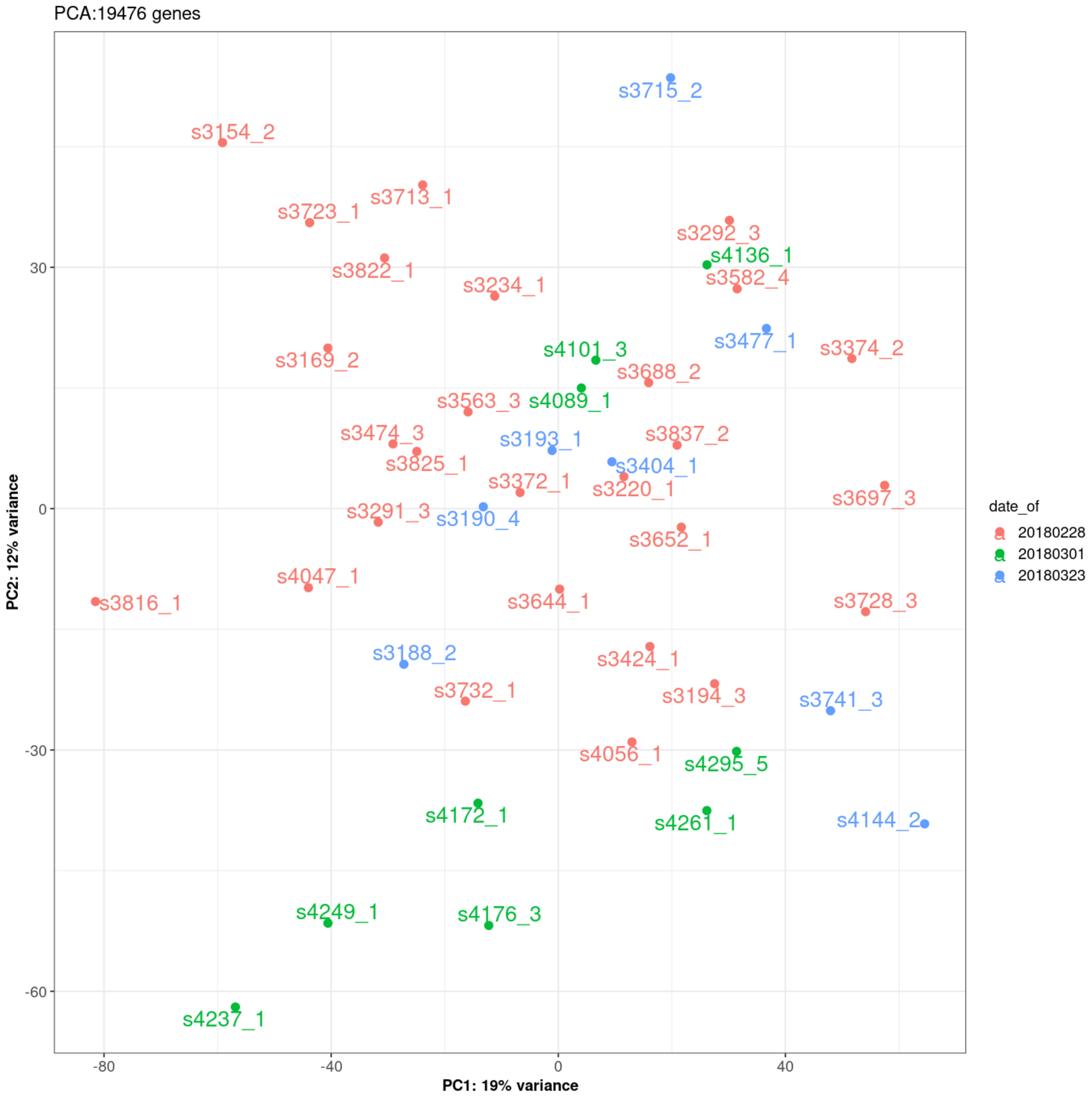
```r
raw_counts <- assays(se)[["raw"]] %>%
    as_tibble() %>%
    dplyr::filter(rowSums(.)!=0) %>%
    as.matrix()

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"]]
```

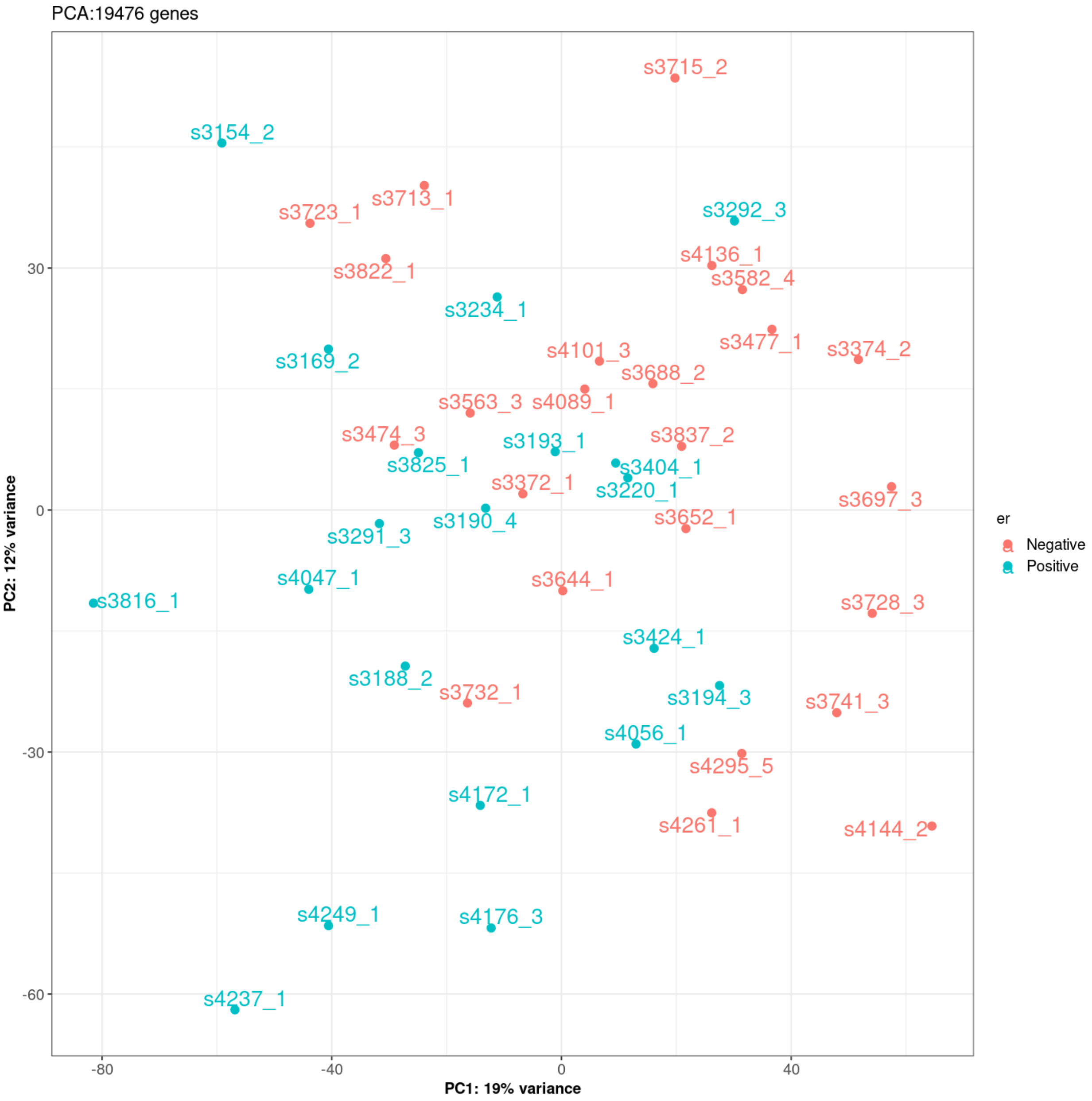
```r
pca_data <- pca[["data"]] %>% as_tibble() %>%
    dplyr::select(sample, PC1, PC2, date_of)

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

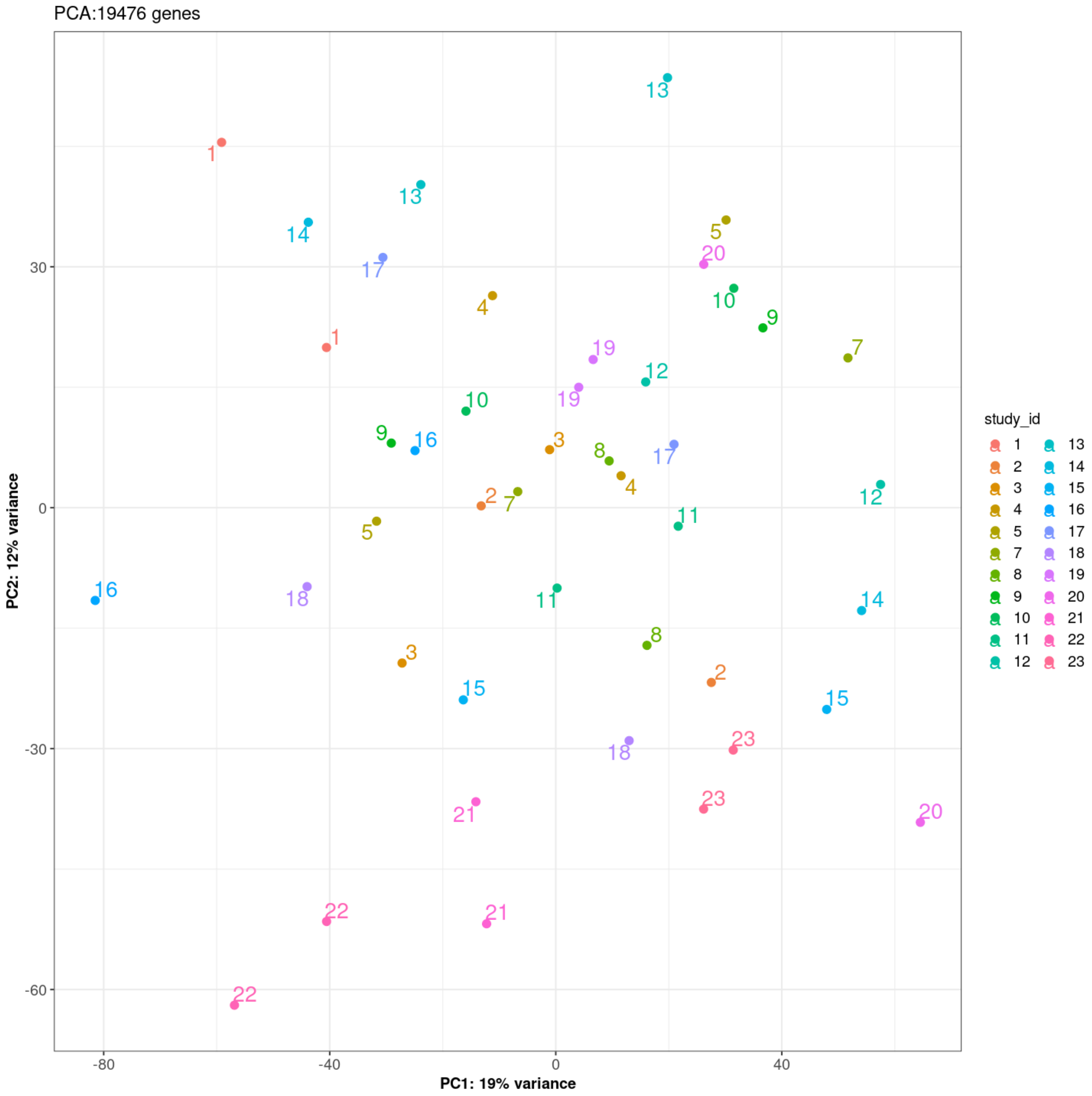
```
raw_counts <- assays(se)[["raw"]] %>%
    as_tibble(rownames = "ensembl_gene_id") %>%
    dplyr::filter(ensembl_gene_id %in% protein_coding_genes$ensembl_gene_id) %>%
    column_to_rownames(var = "ensembl_gene_id") %>%
    dplyr::filter(rowSums(.)!=0) %>%
    as.matrix()

vst <- vst(raw_counts)

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

```
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:19476 genes
s3715_2
s3154_2
s3713_1
s3723_1
s3292_3
s4136_1
s3822_1
s3582_4
s3234_1
s3477_1
s3169_2
s4101_3
s3374_2
s3688_2
s3563_3
s4089_1
s3474_3
s3837_2
s3193_1
s3825_1
s3404_1
s3372_1
s3220_1
s3697_3
s3652_1
s3190_4
s3291_3
s4047_1
s3644_1
s3816_1
s3728_3
s3424_1
s3188_2
s3732_1
s3194_3
s3741_3
s4056_1
s4295_5
s4172_1
s4261_1
s4144_2
s4249_1
s4176_3
s4237_1
30
0
-30
-60
PC2: 12% variance
-80
-40
0
40
PC1: 19% variance
er
Negative
Positive

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

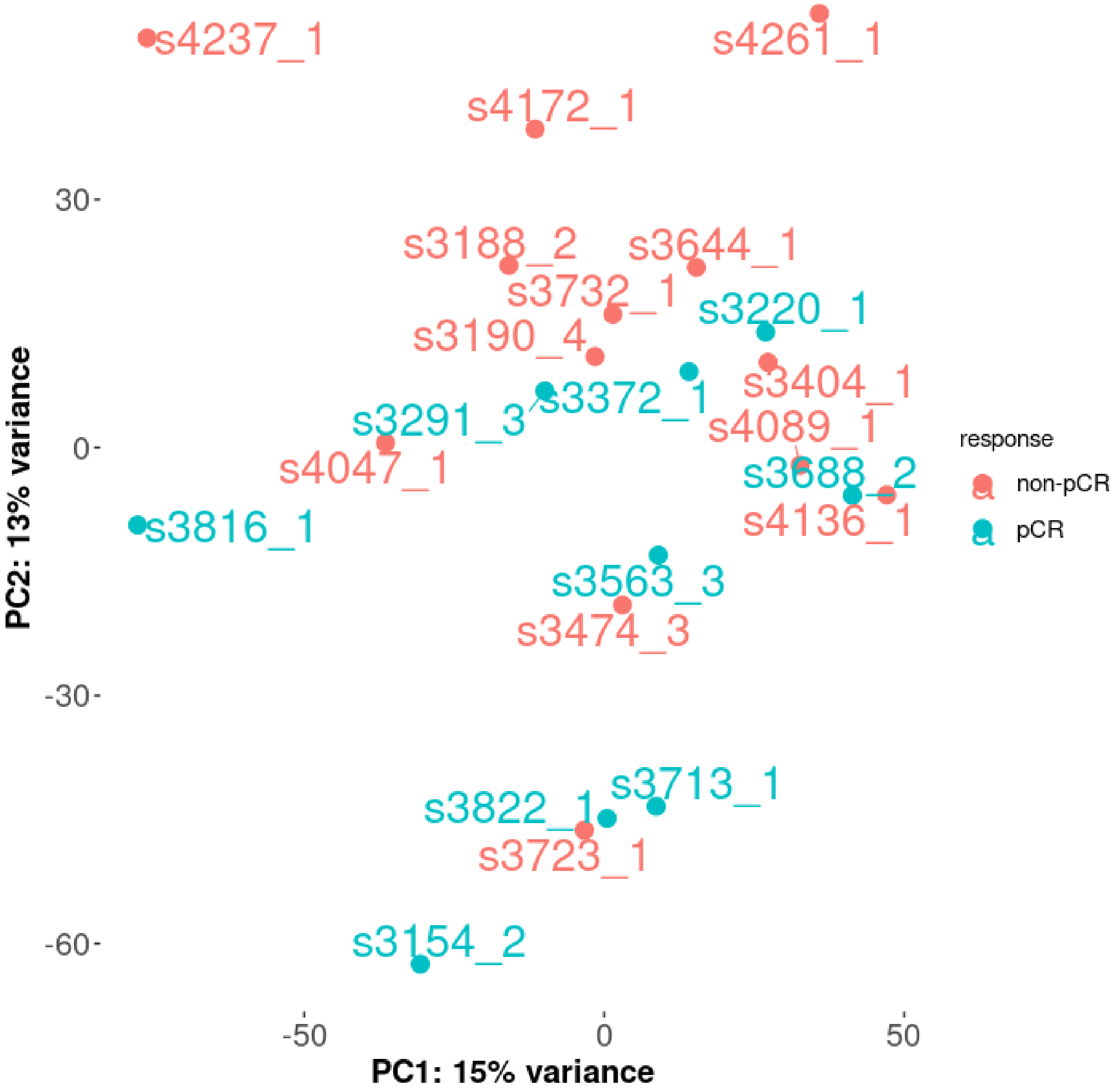
```r
raw_counts <- assays(se)[["raw"]] %>%
    as_tibble(rownames = "ensembl_gene_id") %>%
    dplyr::filter(ensembl_gene_id %in% protein_coding_genes$ensembl_gene_id) %>%
    column_to_rownames(var = "ensembl_gene_id") %>%
    dplyr::filter(rowSums(.)!=0) %>%
    as.matrix()

vst <- vst(raw_counts)

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

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

pca_data %>%
    ggplot(aes(x = PC1, y = PC2, color = study_id, label = study_id)) +
    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:19476 genes
13
1
13
14
5
20
17
4
10
9
1
7
19
12
19
10
9
16
3
8
17
4
2
7
12
5
11
16
18
11
14
8
3
2
15
15
23
18
21
23
20
22
21
22
30
0
-30
-60
-80
-40
0
40
PC2: 12% variance
PC1: 19% variance
study_id
1
2
3
4
5
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

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)

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)

# ggtitle(paste0("PCA:", nrow(vst), " genes")) +
fig3b <- 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) +
    theme_bw(base_size = 15) +
    theme(axis.text = element_text(size = 20),
          axis.title = element_text(size = 20, face="bold"),
          legend.text = element_text(size = rel(1)),
          panel.border = element_blank(),
          panel.grid.minor = element_blank(),
          panel.grid.major = element_blank())

saveRDS(fig3b, "data/fig3b.RDS")
fig3b
```

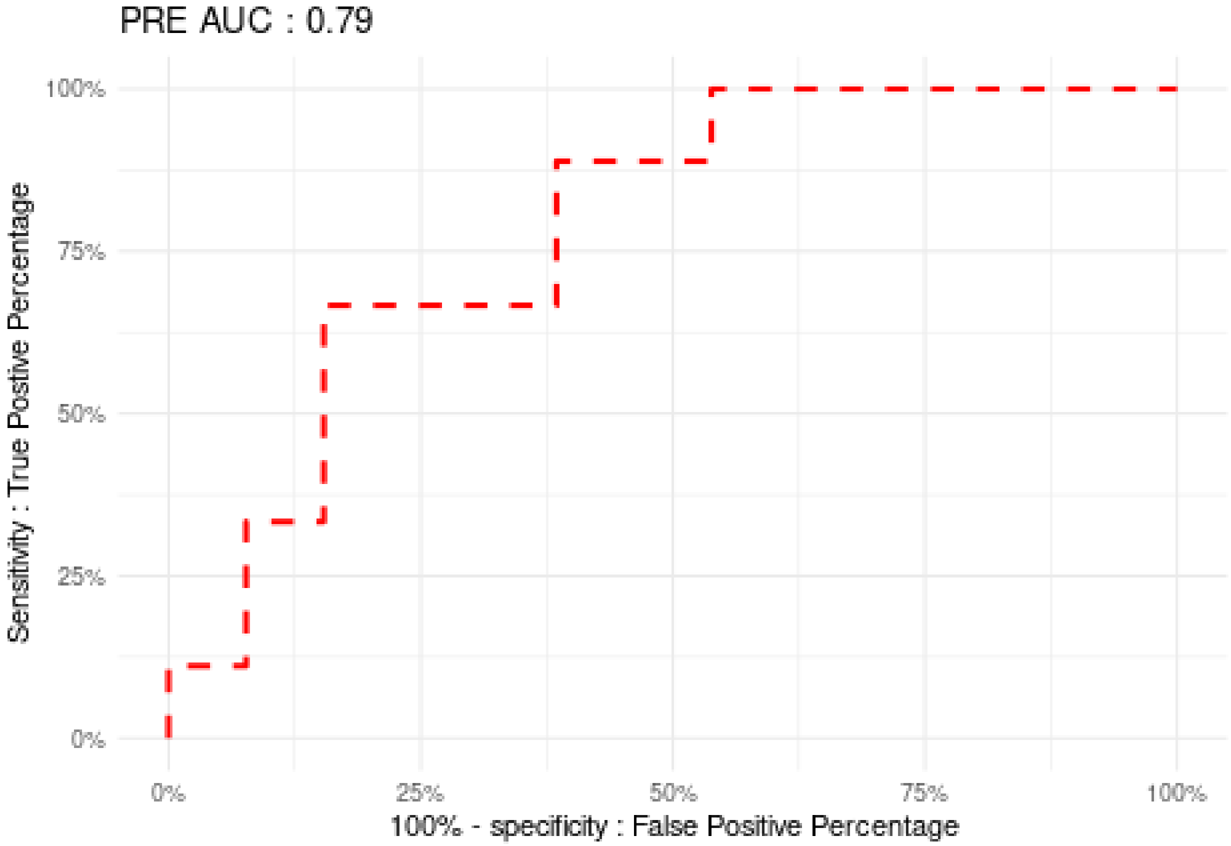

s4237_1
s4261_1
s4172_1
s3188_2
s3644_1
s3732_1
s3220_1
s3190_4
s3404_1
s3372_1
s3291_3
s4089_1
s4047_1
s3688_2
s4136_1
s3816_1
s3563_3
s3474_3
s3713_1
s3822_1
s3723_1
s3154_2
30
0
-30
-60
-50
0
50
PC1: 15% variance
PC2: 13% variance
response
non-pCR
pCR

PRE - ROC

```
#pca_data$PC2
#pca_data$response
roc_pre <- roc(pca_data_pre$response, pca_data_pre$PC2, percent = FALSE)
ci.auc(roc_pre)
```

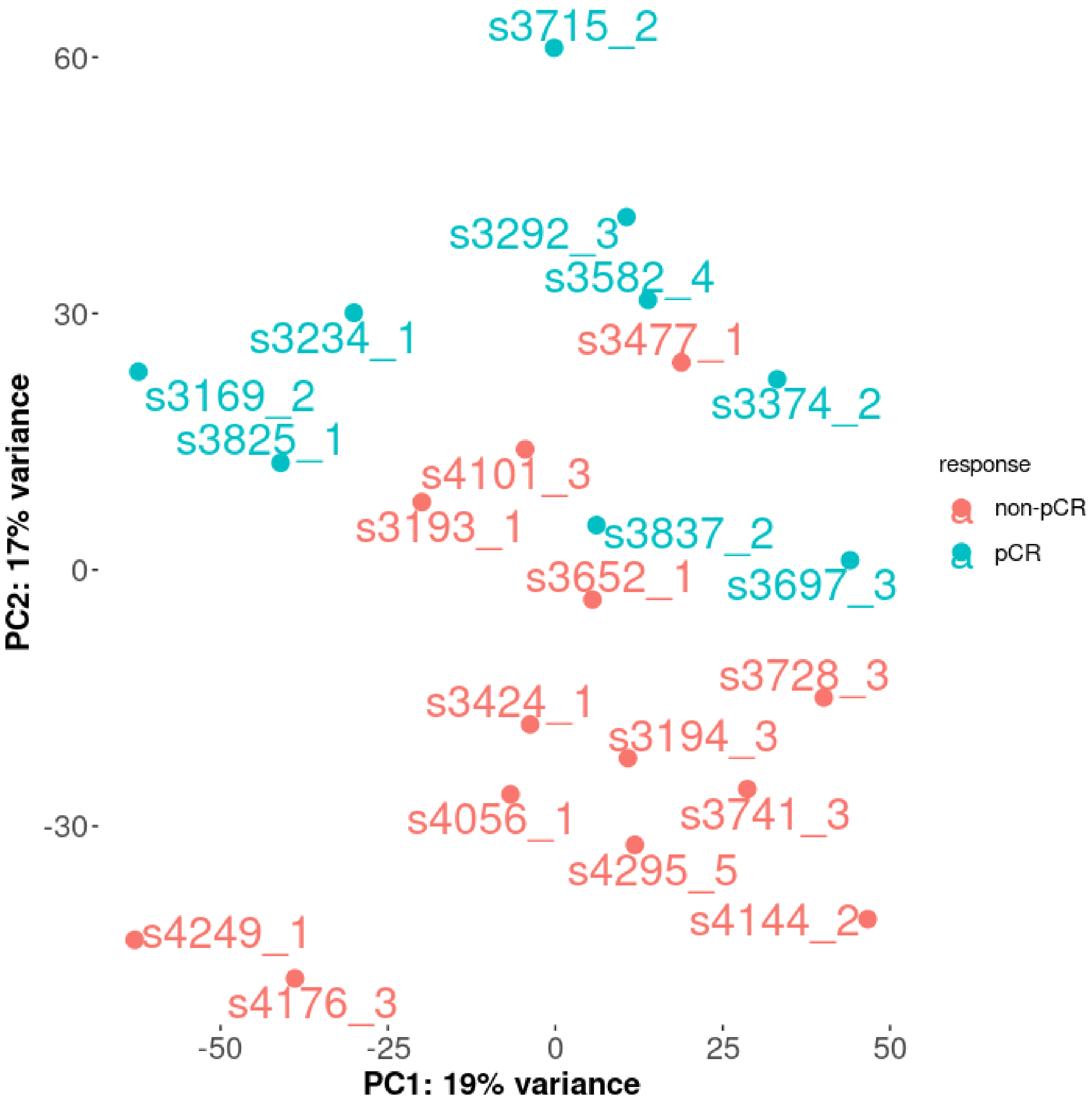
```
## 95% CI: 0.5905-0.9822 (DeLong)
```

```
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

```r
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)

#ggtitle(paste0("PCA:", nrow(vst), " genes")) +
fig3c <- pca_data_post %>%
    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) +
    theme_bw(base_size = 15) +
    theme(axis.text = element_text(size = 20),
          axis.title = element_text(size = 20, face="bold"),
          legend.text = element_text(size = rel(1)),
          panel.border = element_blank(),
          panel.grid.minor = element_blank(),
          panel.grid.major = element_blank())

saveRDS(fig3c, "data/fig3c.RDS")
fig3c
```

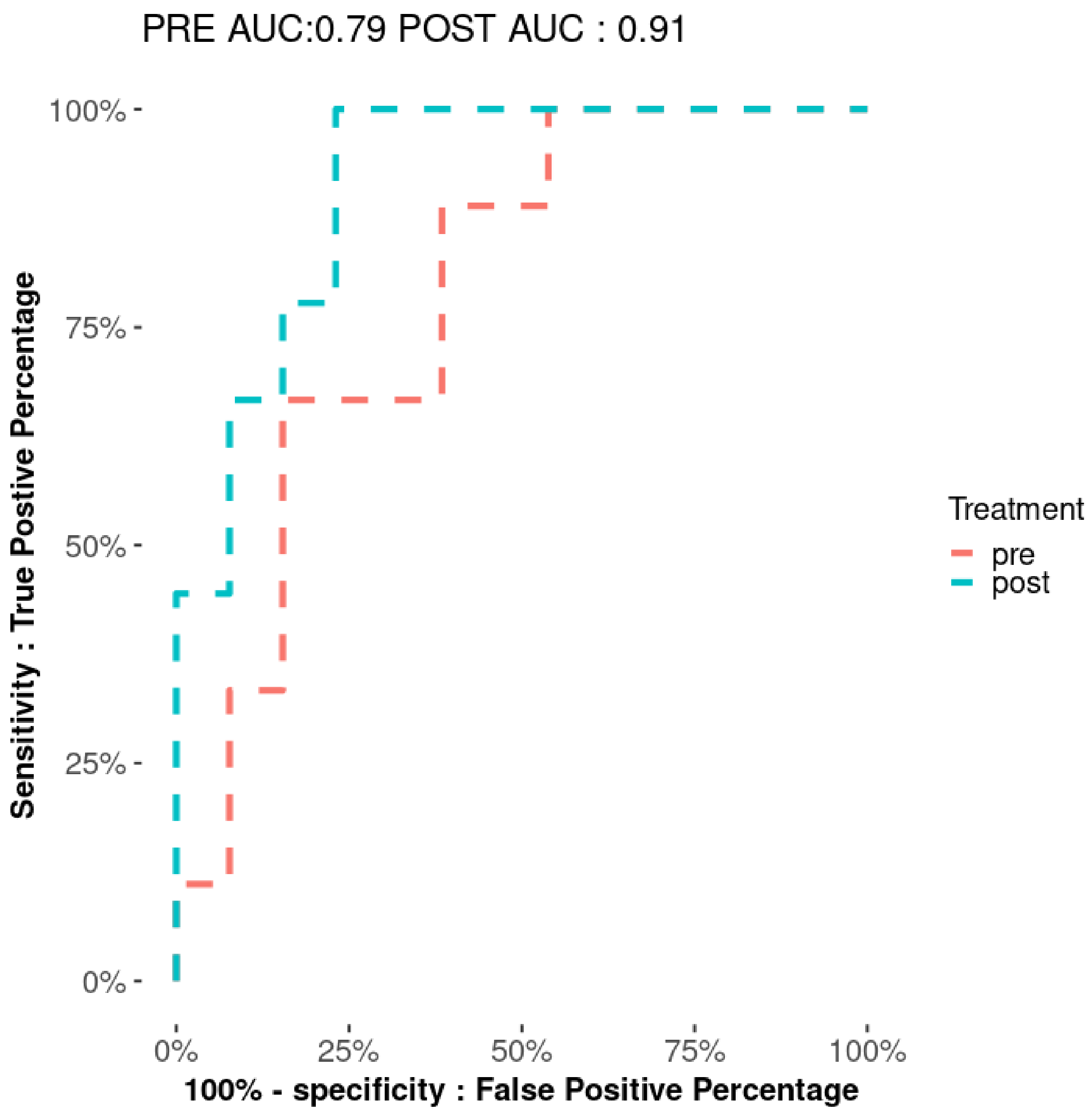

s3715_2
s3292_3
s3582_4
s3234_1
s3477_1
s3169_2
s3374_2
s3825_1
s4101_3
s3193_1
s3837_2
s3652_1
s3697_3
s3728_3
s3424_1
s3194_3
s4056_1
s3741_3
s4295_5
s4144_2
s4249_1
s4176_3
60
30
0
-30
-50
-25
0
25
50
PC1: 19% variance
PC2: 17% variance
response
non-pCR
pCR

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)
ci.auc(roc_post)
```

```
## 95% CI: 0.7993-1 (DeLong)
```

```
fig3d <- pROC::ggroc(list(pre = roc_pre,
           post = roc_post),
           legacy.axes = TRUE, size = 2, 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")) +
  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)),
          panel.border = element_blank(),
          panel.grid.minor = element_blank(),
          panel.grid.major = element_blank())

saveRDS(fig3d, "data/fig3d.RDS")
fig3d
```



CI and roc.test

```
ci.auc(roc_pre)
```

```
## 95% CI: 0.5905-0.9822 (DeLong)
```

```
ci.auc(roc_post)
```

```
## 95% CI: 0.7993-1 (DeLong)
```

```
roc.test(roc_post, roc_pre)
```

```
##
##  Bootstrap test for two correlated ROC curves
```

```
##
## data:  roc_post and roc_pre
## D = 1.1994, boot.n = 2000, boot.stratified = 1, p-value = 0.2304
## alternative hypothesis: true difference in AUC is not equal to 0
## sample estimates:
## AUC of roc1 AUC of roc2
##   0.9145299   0.7863248
```

R session

```
sessionInfo()
```

```
## R version 4.0.5 (2021-03-31)
## 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] pROC_1.17.0.1               ggrepel_0.9.1
##  [3] DEGreport_1.26.0            DESeq2_1.30.1
##  [5] SummarizedExperiment_1.20.0 Biobase_2.50.0
##  [7] MatrixGenerics_1.2.1        matrixStats_0.59.0
##  [9] GenomicRanges_1.42.0        GenomeInfoDb_1.26.7
## [11] IRanges_2.24.1              S4Vectors_0.28.1
## [13] BiocGenerics_0.36.1         knitr_1.33
## [15] forcats_0.5.1               stringr_1.4.0
## [17] dplyr_1.0.7                 purrr_0.3.4
## [19] readr_1.4.0                 tidyr_1.1.3
## [21] tibble_3.1.2                ggplot2_3.3.5
## [23] tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##   [1] colorspace_2.0-2            rjson_0.2.20
##   [3] ellipsis_0.3.2              circlize_0.4.13
##   [5] XVector_0.30.0              ggdendro_0.1.22
##   [7] GlobalOptions_0.1.2         fs_1.5.0
##   [9] clue_0.3-59                 rstudioapi_0.13
##  [11] farver_2.1.0                bit64_4.0.5
##  [13] AnnotationDbi_1.52.0        fansi_0.5.0
##  [15] lubridate_1.7.10            xml2_1.3.2
##  [17] splines_4.0.5               logging_0.10-108
##  [19] mnormt_2.0.2                cachem_1.0.5
##  [21] geneplotter_1.68.0          jsonlite_1.7.2
##  [23] Nozzle.R1_1.1-1             Cairo_1.5-12.2
##  [25] broom_0.7.8                 annotate_1.68.0
##  [27] cluster_2.1.1               dbplyr_2.1.1
##  [29] png_0.1-7                   compiler_4.0.5
##  [31] httr_1.4.2                  backports_1.2.1
##  [33] assertthat_0.2.1           Matrix_1.3-4
##  [35] fastmap_1.1.0              limma_3.46.0
##  [37] cli_2.5.0                  lasso2_1.2-21.1
##  [39] htmltools_0.5.1.1          tools_4.0.5
##  [41] gtable_0.3.0               glue_1.4.2
##  [43] GenomeInfoDbData_1.2.4     Rcpp_1.0.6
##  [45] cellranger_1.1.0           vctrs_0.3.8
##  [47] nlme_3.1-152               psych_2.1.6
##  [49] xfun_0.22                  rvest_1.0.0
##  [51] lifecycle_1.0.0            XML_3.99-0.6
##  [53] edgeR_3.32.1               MASS_7.3-53.1
##  [55] zlibbioc_1.36.0            scales_1.1.1
##  [57] hms_1.1.0                  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_2.2.7
##  [65] highr_0.9                  genefilter_1.72.1
##  [67] BiocParallel_1.24.1        shape_1.4.6
##  [69] rlang_0.4.11               pkgconfig_2.0.3
##  [71] bitops_1.0-7               evaluate_0.14
##  [73] lattice_0.20-41            labeling_0.4.2
##  [75] cowplot_1.1.1              bit_4.0.4
##  [77] tidyselect_1.1.1           plyr_1.8.6
##  [79] magrittr_2.0.1             R6_2.5.0
##  [81] generics_0.1.0             DelayedArray_0.16.3
##  [83] DBI_1.1.1                  pillar_1.6.1
##  [85] haven_2.4.1                withr_2.4.2
##  [87] survival_3.2-10            RCurl_1.98-1.3
##  [89] modelr_0.1.8               crayon_1.4.1
##  [91] utf8_1.2.1                 tmvnsim_1.0-2
##  [93] rmarkdown_2.6              GetoptLong_1.0.5
##  [95] locfit_1.5-9.4             grid_4.0.5
##  [97] readxl_1.3.1               blob_1.2.1
##  [99] ConsensusClusterPlus_1.54.0 reprex_2.0.0
## [101] digest_0.6.27              xtable_1.8-4
## [103] munsell_0.5.0
```