Exosomes raw data Quality control

David Cáceres

30/1/2023

Contents

Data read 1

Reads distribution by batch with brain replicate control **2**

Effective reads percent distribution 2

Short reads percent. Less than 17 nucleotides 4

Ultra short reads percent. Less than 15 nucleotides. 6

Reference genome contamination percent. 8

Unmapped reads percent. 11

Bacteria reads percent. Contamination. 13

Virus reads percent. Contamination 15

Total reads in absolute value. 17

Expression Matrix **19**

Builiding a expression matrix for microRNA data 19

Genome Distribution **23**

Building a expression matrix for Genome distribution 23

Genome Distribution 23

Multi dimensional Scaling **25**

DESeq2 object 25

Multi dimenstional scaling plots 25

Principal components correlation 30

Data read

```
setwd("~/Exosomas/QC/Data/")
tipos_exo <- read.table(file = "readsPerc.tsv",
    header = TRUE, sep = "\t")
brainq_reads <- read.table(file = "brain_reads.tsv",
    header = TRUE, sep = "\t")
brainq_short <- read.table(file = "brainq_short.tsv",
    header = TRUE, sep = "\t")
brainq_ushort <- read.table(file = "brainq_ushort.tsv",
    header = TRUE, sep = "\t")
short <- read.table(file = "shortReadsPerc.tsv",
    header = TRUE, sep = "\t")
ushort <- read.table(file = "ultrashort.tsv",
    header = TRUE, sep = "\t")
ref.genome <- read.table(file = "ref.genome.tsv",
    header = TRUE, sep = "\t")
unmapped <- read.table(file = "unmapped.tsv",
    header = TRUE, sep = "\t")
contaminacion <- read.table(file = "Contaminacion.tsv",
    header = TRUE, sep = "\t")
brainq_contaminacion <- read.table(file = "brainq_contaminacion.tsv",
    header = TRUE, sep = "\t")
reads_total <- read.table(file = "Total_reads.tsv",
    header = TRUE, sep = "\t")
```

Reads distribution by batch with brain replicate control

Effective reads percent distribution

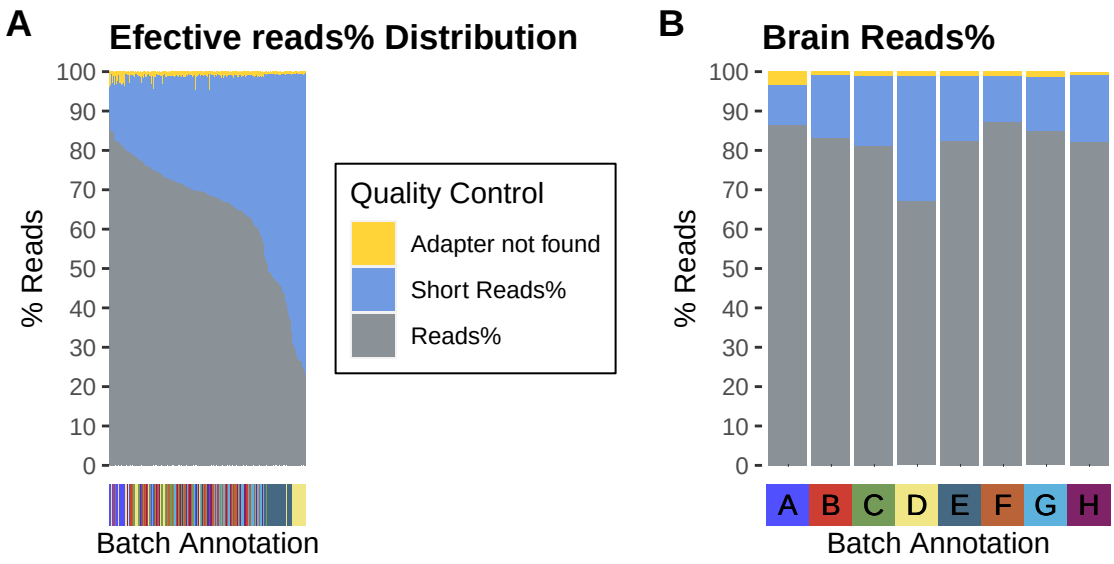
We used miRNAQC to get quality reports and datasets used to pre-process the raw data and check it quality. arn.ugr.es/mirnaqc/

Initially we plotted the effective reads percent vs the sample distribution by batch. Also plotted the percernt of short reads and not adapter found. As reference we plotted the brain replicate data.

```
# Adding shortnames

values <- c("A", "B", "C", "D",
    "E", "F", "G", "H")
tipos_exo$batch <- as.factor(tipos_exo$batch)
tipos_exo$batch_shortname <- values[tipos_exo$batch]

# Percent of effective reads

tipos_exo <- tipos_exo |>
    mutate(sample1 = reorder(sample,
        -ifelse(!X %in% "readsPerc",
            0, value), FUN = sum))

p1 <- ggplot(tipos_exo, aes(x = sample1,
    y = value, fill = factor(X,
        levels = c("readsAdapterNotFoundPerc",
            "shortReads", "readsPerc")),
```

```
    width = 1.05)) + geom_bar(stat = "identity",
    position = position_stack()) +
    scale_y_continuous(expand = c(0,
        0), breaks = seq(0, 100,
        by = 10)) + labs(x = NULL,
    y = "% Reads", title = "Efective reads% Distribution",
    fill = "Quality Control") +
    theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.ticks.length.x = unit(0,
            "pt")) + theme(panel.background = element_rect(fill = "transparent"),
    plot.background = element_rect(fill = "transparent",
        color = NA), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    legend.background = element_rect(fill = "transparent"),
    legend.box.background = element_rect(fill = "transparent"),
    plot.title = element_text(face = "bold")) +
    scale_fill_simpsons(labels = c("Adapter not found",
        "Short Reads%", "Reads%"))
tipos_exo1 <- tipos_exo |>
    distinct(sample, sample1, batch_shortname)

p_axis <- ggplot(tipos_exo1, aes(x = sample1,
    y = factor(1), fill = batch_shortname)) +
    geom_tile(width = 1) + theme_void() +
    theme(axis.title.x = element_text()) +
    theme(legend.position = "none") +
    labs(x = "Batch Annotation",
        fill = "Batch")

p1_q <- p1/p_axis + plot_layout(heights = c(8,
    1)) + scale_fill_igv()

# Brain

p_reads_brain <- ggplot(brainq_reads,
    aes(x = batch, y = value1,
        fill = factor(variable1,
            levels = c("readsAdapterNotFoundPerc",
                "shortReads", "readsPerc")))) +
    geom_bar(stat = "identity") +
    scale_y_continuous(expand = c(0,
        0), breaks = seq(0, 100,
        by = 10)) + labs(x = NULL,
    y = "% Reads", title = "Brain Reads%") +
    theme(axis.text.x = element_blank(),
        axis.ticks.length.x = unit(0,
            "pt")) + scale_fill_simpsons() +
    theme(panel.background = element_rect(fill = "transparent"),
        plot.background = element_rect(fill = "transparent",
            color = NA), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.background = element_rect(fill = "transparent"),
```

```
        legend.box.background = element_rect(fill = "transparent"),
        plot.title = element_text(face = "bold")) +
    theme(legend.position = "none")
p_axis_reads_brain <- ggplot(brainq_reads,
    aes(x = batch, y = factor(1),
        fill = batch_shortname)) +
    geom_tile(width = 1) + theme_void() +
    theme(axis.title.x = element_text()) +
    geom_text(aes(label = batch_shortname)) +
    labs(x = "Batch Annotation",
        fill = "Batch Annotation") +
    theme(legend.position = "none")
p3_qb <- p_reads_brain/p_axis_reads_brain +
    plot_layout(heights = c(8,
        1)) + scale_fill_igv()

pt1 <- plot_grid(p1_q, p3_qb, rel_widths = c(1.7,
    1.3), labels = c("A", "B"))

pt1
```



Short reads percent. Less than 17 nucleotides

Plot of the short reads obtained from mirnaQC

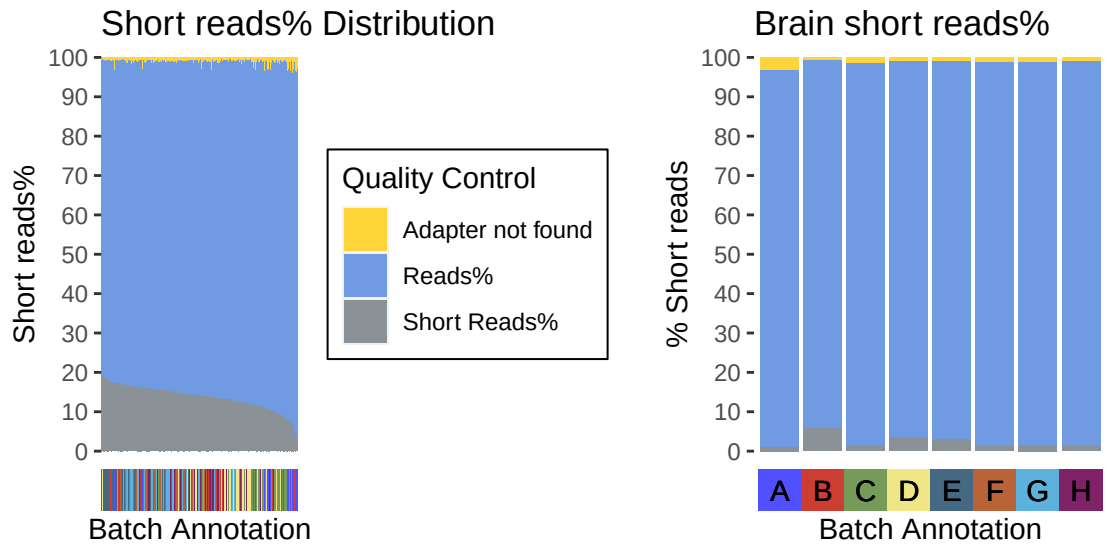
```
# Short Reads%
short <- short |>
    mutate(sample1 = reorder(sample,
        -ifelse(!X %in% "shortReadsPerc",
            0, value), FUN = sum))

p_short <- ggplot(short, aes(x = sample1,
    y = value, fill = factor(X,
```

```
        levels = c("readsAdapterNotFoundPerc",
            "reads", "shortReadsPerc")),
    width = 1.05)) + geom_bar(stat = "identity") +
    scale_y_continuous(expand = c(0,
        0), breaks = seq(0, 100,
        by = 10)) + labs(x = NULL,
    y = "Short reads%", title = "Short reads% Distribution",
    fill = "Quality Control") +
    theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.ticks.length.x = unit(0,
            "pt")) + theme(panel.background = element_rect(fill = "transparent"),
    plot.background = element_rect(fill = "transparent",
        color = NA), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    legend.background = element_rect(fill = "transparent"),
    legend.box.background = element_rect(fill = "transparent")) +
    scale_fill_simpsons(labels = c("Adapter not found",
        "Reads%", "Short Reads%"))

p_axis_short <- ggplot(short, aes(x = sample1,
    y = factor(1), fill = batch)) +
    geom_tile(width = 1) + theme_void() +
    theme(axis.title.x = element_text()) +
    theme(legend.position = "none") +
    labs(x = "Batch Annotation",
        fill = "Batch")
p3q <- p_short/p_axis_short + plot_layout(heights = c(8,
    1)) + scale_fill_igv()

# Brain

p_short_brain <- ggplot(brainq_short,
    aes(x = batch, y = value1,
        fill = factor(variable1,
            levels = c("readsAdapterNotFoundPerc",
                "reads", "shortReadsPerc")))) +
    geom_bar(stat = "identity") +
    scale_y_continuous(expand = c(0,
        0), breaks = seq(0, 100,
        by = 10)) + labs(x = NULL,
    y = "% Short reads", title = "Brain short reads%",
    fill = "QC") + theme(axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.ticks.length.x = unit(0,
        "pt")) + scale_fill_simpsons() +
    theme(panel.background = element_rect(fill = "transparent"),
        plot.background = element_rect(fill = "transparent",
            color = NA), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.background = element_rect(fill = "transparent"),
        legend.box.background = element_rect(fill = "transparent")) +
    theme(legend.position = "none")
```

```
p_axis_short_brain <- ggplot(brainq_short,
    aes(x = batch, y = factor(1),
        fill = batch_shortname)) +
    geom_tile(width = 1) + theme_void() +
    theme(axis.title.x = element_text()) +
    geom_text(aes(label = batch_shortname)) +
    labs(x = "Batch Annotation",
        fill = "Batch Annotation") +
    theme(legend.position = "none")
p4_qb <- p_short_brain/p_axis_short_brain +
    plot_layout(heights = c(8,
        1)) + scale_fill_igv()

pt2 <- plot_grid(p3q, p4_qb, rel_widths = c(1.7,
    1.3))
```

```
Warning: 'position_stack()' requires non-overlapping x intervals
```

```
pt2
```



Ultra short reads percent. Less than 15 nucleotides.

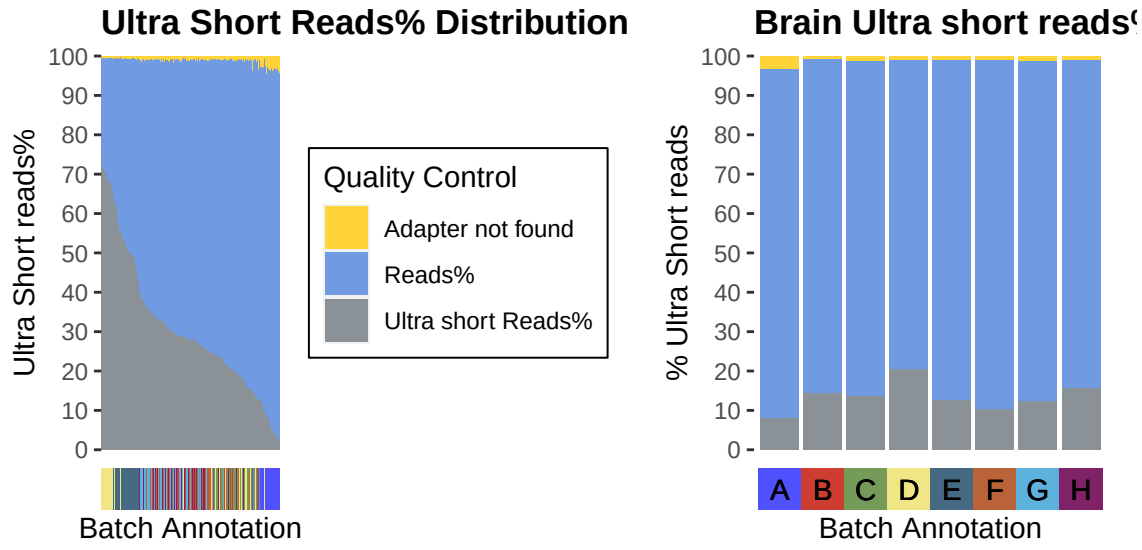
```
# Ultra Short%
ushort <- ushort |>
    mutate(sample1 = reorder(sample,
        -ifelse(!X %in% "ultraShortReadsPerc",
            0, value), FUN = sum))
p_ushort <- ggplot(ushort, aes(x = sample1,
    y = value, fill = factor(X,
        levels = c("readsAdapterNotFoundPerc",
            "reads", "ultraShortReadsPerc")),
```

```
    width = 1.05)) + geom_bar(stat = "identity") +
    scale_y_continuous(expand = c(0,
        0), breaks = seq(0, 100,
        by = 10)) + labs(x = NULL,
    y = "Ultra Short reads%", title = "Ultra Short Reads% Distribution",
    fill = "Quality Control") +
    theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.ticks.length.x = unit(0,
            "pt")) + theme(panel.background = element_rect(fill = "transparent"),
    plot.background = element_rect(fill = "transparent",
        color = NA), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    legend.background = element_rect(fill = "transparent"),
    legend.box.background = element_rect(fill = "transparent"),
    plot.title = element_text(face = "bold")) +
    scale_fill_simpsons(labels = c("Adapter not found",
        "Reads%", "Ultra short Reads%"))

p_axis_ushort <- ggplot(ushort,
    aes(x = sample1, y = factor(1),
        fill = batch)) + geom_tile(width = 1) +
    theme_void() + theme(axis.title.x = element_text()) +
    theme(legend.position = "none") +
    labs(x = "Batch Annotation",
        fill = "Batch")
p5_q <- p_ushort/p_axis_ushort +
    plot_layout(heights = c(8,
        1)) + scale_fill_igv()

# Brain

p_ushort_brain <- ggplot(brainq_ushort,
    aes(x = batch, y = value1,
        fill = factor(variable1,
            levels = c("readsAdapterNotFoundPerc",
                "reads", "ultraShortReadsPerc")))) +
    geom_bar(stat = "identity") +
    scale_y_continuous(expand = c(0,
        0), breaks = seq(0, 100,
        by = 10)) + labs(x = NULL,
    y = "% Ultra Short reads",
    title = "Brain Ultra short reads%",
    fill = "QC") + theme(axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.ticks.length.x = unit(0,
        "pt")) + scale_fill_simpsons() +
    theme(panel.background = element_rect(fill = "transparent"),
        plot.background = element_rect(fill = "transparent",
            color = NA), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.background = element_rect(fill = "transparent"),
        legend.box.background = element_rect(fill = "transparent"),
```

```
        plot.title = element_text(face = "bold")) +
    theme(legend.position = "none")

p_axis_ushort_brain <- ggplot(brainq_ushort,
    aes(x = batch, y = factor(1),
        fill = batch_shortname)) +
    geom_tile(width = 1) + theme_void() +
    theme(axis.title.x = element_text()) +
    geom_text(aes(label = batch_shortname)) +
    labs(x = "Batch Annotation",
        fill = "Batch Annotation") +
    theme(legend.position = "none")
p5_qb <- p_ushort_brain/p_axis_ushort_brain +
    plot_layout(heights = c(8,
        1)) + scale_fill_igv()

pt3 <- plot_grid(p5_q, p5_qb, rel_widths = c(1.7,
    1.3))
```

```
Warning: 'position_stack()' requires non-overlapping x intervals
```

```
pt3
```



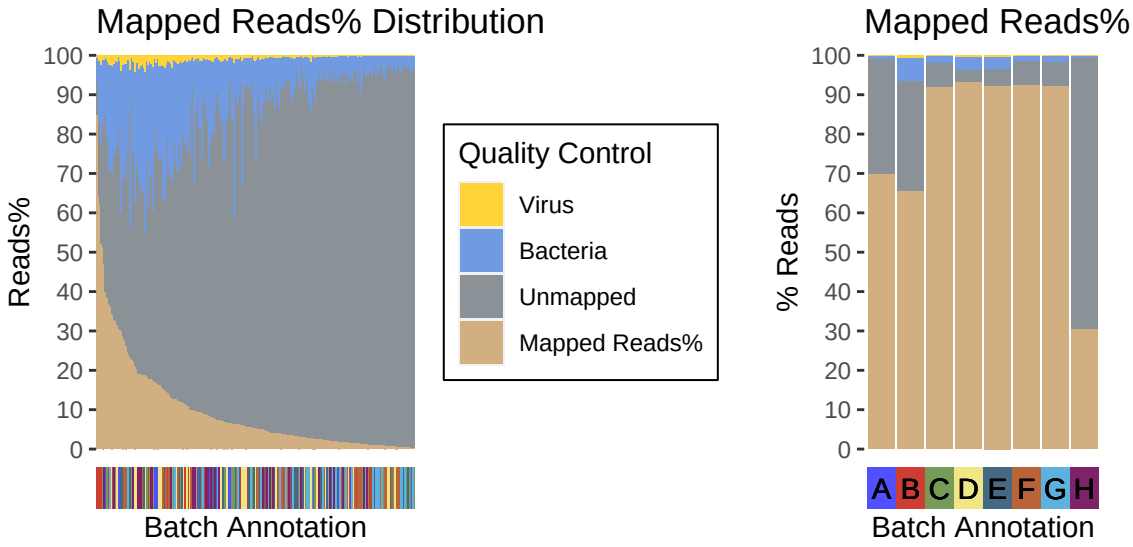
Reference genome contamination percent.

```
contaminacion$batch <- as.factor(contaminacion$batch)
contaminacion$batch_shortname <- values[contaminacion$batch]

# Referece genome
# contamination
```

```
contaminacion <- contaminacion |>
    mutate(sample1 = reorder(sample,
        -ifelse(!variable %in%
            "% ref.genome", 0,
            value), FUN = sum))
p_ref.genome <- ggplot(contaminacion,
    aes(x = sample1, y = value,
        fill = factor(variable,
            levels = c("virus",
                "bacteria", "unmapped",
                "% ref.genome")),
        width = 1.05)) + geom_bar(stat = "identity") +
    scale_y_continuous(expand = c(0,
        0), breaks = seq(0, 100,
        by = 10)) + labs(x = NULL,
    y = "Reads%", title = "Mapped Reads% Distribution",
    fill = "Quality Control") +
    theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.ticks.length.x = unit(0,
            "pt")) + theme(panel.background = element_rect(fill = "transparent"),
    plot.background = element_rect(fill = "transparent",
        color = NA), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    legend.background = element_rect(fill = "transparent"),
    legend.box.background = element_rect(fill = "transparent")) +
    scale_fill_simpsons(labels = c("Virus",
        "Bacteria", "Unmapped",
        "Mapped Reads%"))

contaminacion1 <- contaminacion |>
    distinct(sample, sample1, batch_shortname)
p_axis_ref.genome <- ggplot(contaminacion1,
    aes(x = sample1, y = factor(1),
        fill = batch_shortname)) +
    geom_tile(width = 1) + theme_void() +
    theme(axis.title.x = element_text()) +
    theme(legend.position = "none") +
    labs(x = "Batch Annotation",
        fill = "Batch")
p6_q <- p_ref.genome/p_axis_ref.genome +
    plot_layout(heights = c(8,
        1)) + scale_fill_igv()

# Brain

brainq_contaminacion$batch <- as.factor(brainq_contaminacion$batch)
brainq_contaminacion$batch_shortname <- values[brainq_contaminacion$batch]

p_ref.genome_brain <- ggplot(brainq_contaminacion,
    aes(x = batch, y = value, fill = factor(variable,
        levels = c("virus", "bacteria",
            "unmapped", "refSpecies")))) +
```

```
    geom_bar(stat = "identity") +
    scale_y_continuous(expand = c(0,
        0), breaks = seq(0, 100,
        by = 10)) + labs(x = NULL,
    y = "% Reads", title = "Mapped Reads%",
    fill = "QC") + theme(axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.ticks.length.x = unit(0,
        "pt")) + scale_fill_simpsons() +
    theme(panel.background = element_rect(fill = "transparent"),
        plot.background = element_rect(fill = "transparent",
            color = NA), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.background = element_rect(fill = "transparent"),
        legend.box.background = element_rect(fill = "transparent")) +
    theme(legend.position = "none")


p_axis_ref.genome_brain <- ggplot(brainq_contaminacion,
    aes(x = batch, y = factor(1),
        fill = batch_shortname)) +
    geom_tile(width = 1) + theme_void() +
    theme(axis.title.x = element_text()) +
    geom_text(aes(label = batch_shortname)) +
    labs(x = "Batch Annotation",
        fill = "Batch Annotation") +
    theme(legend.position = "none")
p6_qb <- p_ref.genome_brain/p_axis_ref.genome_brain +
    plot_layout(heights = c(8,
        1)) + scale_fill_igv()

plot_grid(p6_q, p6_qb, rel_widths = c(2,
    1))
```

```
Warning: ‘position_stack()‘ requires non-overlapping x intervals
```



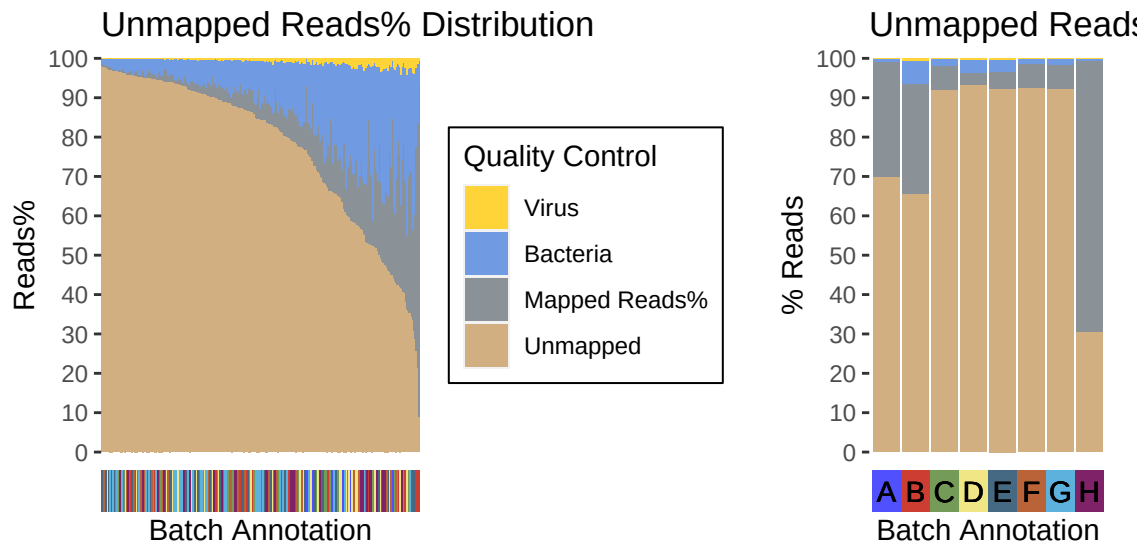
Unmapped reads percent.

```
# Unmapped reads%

contaminacion <- contaminacion |>
    mutate(sample1 = reorder(sample,
        -ifelse(!variable %in%
            "unmapped", 0, value),
        FUN = sum))

p_unmapped <- ggplot(contaminacion,
    aes(x = sample1, y = value,
        fill = factor(variable,
            levels = c("virus",
                "bacteria", "% ref.genome",
                "unmapped")), width = 1.05)) +
    geom_bar(stat = "identity") +
    scale_y_continuous(expand = c(0,
        0), breaks = seq(0, 100,
        by = 10)) + labs(x = NULL,
    y = "Reads%", title = "Unmapped Reads% Distribution",
    fill = "Quality Control") +
    theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.ticks.length.x = unit(0,
            "pt")) + theme(panel.background = element_rect(fill = "transparent"),
    plot.background = element_rect(fill = "transparent",
        color = NA), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    legend.background = element_rect(fill = "transparent"),
    legend.box.background = element_rect(fill = "transparent")) +
    scale_fill_simpsons(labels = c("Virus",
        "Bacteria", "Mapped Reads%",
```

```
        "Unmapped"))


contaminacion1 <- contaminacion |>
    distinct(sample, sample1, batch_shortname)

p_axis_unmapped <- ggplot(contaminacion1,
    aes(x = sample1, y = factor(1),
        fill = batch_shortname)) +
    geom_tile(width = 1) + theme_void() +
    theme(axis.title.x = element_text()) +
    theme(legend.position = "none") +
    labs(x = "Batch Annotation",
        fill = "Batch")
p7_q <- p_unmapped/p_axis_unmapped +
    plot_layout(heights = c(8,
        1)) + scale_fill_igv()

# Brain

p_unmapped_brain <- ggplot(brainq_contaminacion,
    aes(x = batch, y = value, fill = factor(variable,
        levels = c("virus", "bacteria",
            "unmapped", "refSpecies")))) +
    geom_bar(stat = "identity") +
    scale_y_continuous(expand = c(0,
        0), breaks = seq(0, 100,
        by = 10)) + labs(x = NULL,
    y = "% Reads", title = "Unmapped Reads%",
    fill = "QC") + theme(axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.ticks.length.x = unit(0,
        "pt")) + scale_fill_simpsons() +
    theme(panel.background = element_rect(fill = "transparent"),
        plot.background = element_rect(fill = "transparent",
            color = NA), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.background = element_rect(fill = "transparent"),
        legend.box.background = element_rect(fill = "transparent")) +
    theme(legend.position = "none")

p_axis_unmapped_brain <- ggplot(brainq_contaminacion,
    aes(x = batch, y = factor(1),
        fill = batch_shortname)) +
    geom_tile(width = 1) + theme_void() +
    theme(axis.title.x = element_text()) +
    geom_text(aes(label = batch_shortname)) +
    labs(x = "Batch Annotation",
        fill = "Batch Annotation") +
    theme(legend.position = "none")
p7_qb <- p_unmapped_brain/p_axis_unmapped_brain +
    plot_layout(heights = c(8,
        1)) + scale_fill_igv()
```

```
plot_grid(p7_q, p7_qb, rel_widths = c(2,
    1))
```

```
Warning: 'position_stack()' requires non-overlapping x intervals
```



Bacteria reads percent. Contamination.

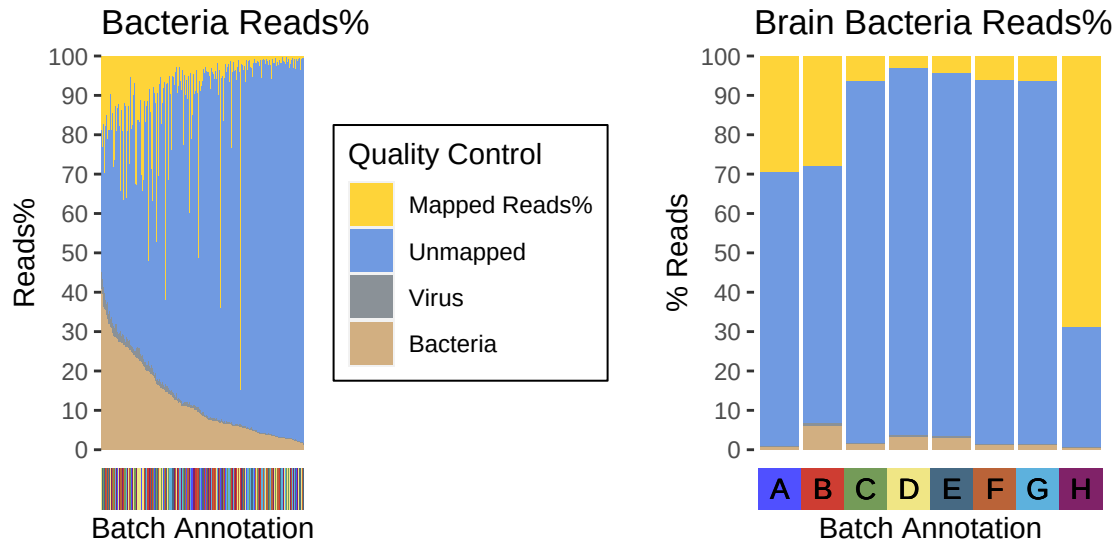
```
# Bacteria

contaminacion <- contaminacion |>
    mutate(sample1 = reorder(sample,
        -ifelse(!variable %in%
            "bacteria", 0, value),
        FUN = sum))

p_bacteria <- ggplot(contaminacion,
    aes(x = sample1, y = value,
        fill = factor(variable,
            levels = c("% ref.genome",
                "unmapped", "virus",
                "bacteria")), width = 1.05)) +
    geom_bar(stat = "identity") +
    scale_y_continuous(expand = c(0,
        0), breaks = seq(0, 100,
        by = 10)) + labs(x = NULL,
    y = "Reads%", title = "Bacteria Reads%",
    fill = "Quality Control") +
    theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.ticks.length.x = unit(0,
            "pt")) + theme(panel.background = element_rect(fill = "transparent"),
```

```
    plot.background = element_rect(fill = "transparent",
        color = NA), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    legend.background = element_rect(fill = "transparent"),
    legend.box.background = element_rect(fill = "transparent")) +
    scale_fill_simpsons(labels = c("Mapped Reads%",
        "Unmapped", "Virus", "Bacteria"))

contaminacion1 <- contaminacion |>
    distinct(sample, sample1, batch_shortname)

p_axis_bacteria <- ggplot(contaminacion1,
    aes(x = sample1, y = factor(1),
        fill = batch_shortname)) +
    geom_tile(width = 1) + theme_void() +
    theme(axis.title.x = element_text()) +
    theme(legend.position = "none") +
    labs(x = "Batch Annotation",
        fill = "Batch")
p8_q <- p_bacteria/p_axis_bacteria +
    plot_layout(heights = c(8,
        1)) + scale_fill_igv()

# Brain

p_bacteria_brain <- ggplot(brainq_contaminacion,
    aes(x = batch, y = value, fill = factor(variable,
        levels = c("unmapped",
            "refSpecies", "virus",
            "bacteria")))) + geom_bar(stat = "identity") +
    scale_y_continuous(expand = c(0,
        0), breaks = seq(0, 100,
        by = 10)) + labs(x = NULL,
    y = "% Reads", title = "Brain Bacteria Reads%",
    fill = "QC") + theme(axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.ticks.length.x = unit(0,
        "pt")) + scale_fill_simpsons() +
    theme(panel.background = element_rect(fill = "transparent"),
        plot.background = element_rect(fill = "transparent",
            color = NA), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.background = element_rect(fill = "transparent"),
        legend.box.background = element_rect(fill = "transparent")) +
    theme(legend.position = "none")

p_axis_bacteria_brain <- ggplot(brainq_contaminacion,
    aes(x = batch, y = factor(1),
        fill = batch_shortname)) +
    geom_tile(width = 1) + theme_void() +
    theme(axis.title.x = element_text()) +
    geom_text(aes(label = batch_shortname)) +
    labs(x = "Batch Annotation",
```

```
        fill = "Batch Annotation") +
    theme(legend.position = "none")

p8_qb <- p_bacteria_brain/p_axis_bacteria_brain +
    plot_layout(heights = c(8,
        1)) + scale_fill_igv()


pt8 <- plot_grid(p8_q, p8_qb, rel_widths = c(1.7,
    1.3))
```

```
Warning: 'position_stack()' requires non-overlapping x intervals
```

```
pt8
```



Virus reads percent. Contamination

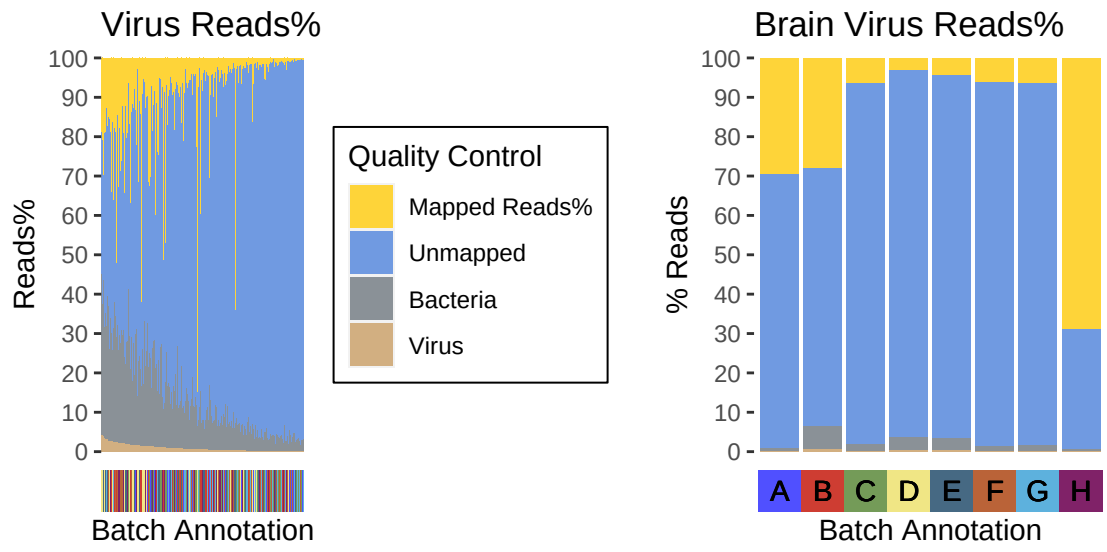
```
# Virus

contaminacion <- contaminacion |>
    mutate(sample1 = reorder(sample,
        -ifelse(!variable %in%
            "virus", 0, value),
        FUN = sum))
p_virus <- ggplot(contaminacion,
    aes(x = sample1, y = value,
        fill = factor(variable,
            levels = c("% ref.genome",
                "unmapped", "bacteria",
                "virus")), width = 1.05)) +
    geom_bar(stat = "identity") +
```

```
    scale_y_continuous(expand = c(0,
        0), breaks = seq(0, 100,
        by = 10)) + labs(x = NULL,
    y = "Reads%", title = "Virus Reads%",
    fill = "Quality Control") +
    theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.ticks.length.x = unit(0,
            "pt")) + theme(panel.background = element_rect(fill = "transparent"),
    plot.background = element_rect(fill = "transparent",
        color = NA), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    legend.background = element_rect(fill = "transparent"),
    legend.box.background = element_rect(fill = "transparent")) +
    scale_fill_simpsons(labels = c("Mapped Reads%",
        "Unmapped", "Bacteria",
        "Virus"))

contaminacion1 <- contaminacion |>
    distinct(sample, sample1, batch_shortname)

p_axis_virus <- ggplot(contaminacion1,
    aes(x = sample1, y = factor(1),
        fill = batch_shortname)) +
    geom_tile(width = 1) + theme_void() +
    theme(axis.title.x = element_text()) +
    theme(legend.position = "none") +
    labs(x = "Batch Annotation",
        fill = "Batch")

p9_q <- p_virus/p_axis_virus +
    plot_layout(heights = c(8,
        1)) + scale_fill_igv()

# Brain

p_virus_brain <- ggplot(brainq_contaminacion,
    aes(x = batch, y = value, fill = factor(variable,
        levels = c("unmapped",
            "refSpecies", "bacteria",
            "virus")))) + geom_bar(stat = "identity") +
    scale_y_continuous(expand = c(0,
        0), breaks = seq(0, 100,
        by = 10)) + labs(x = NULL,
    y = "% Reads", title = "Brain Virus Reads%",
    fill = "QC") + theme(axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.ticks.length.x = unit(0,
        "pt")) + scale_fill_simpsons() +
    theme(panel.background = element_rect(fill = "transparent"),
        plot.background = element_rect(fill = "transparent",
            color = NA), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
```

```
        legend.background = element_rect(fill = "transparent"),
        legend.box.background = element_rect(fill = "transparent")) +
    theme(legend.position = "none")

p_axis_virus_brain <- ggplot(brainq_contaminacion,
    aes(x = batch, y = factor(1),
        fill = batch_shortname)) +
    geom_tile(width = 1) + theme_void() +
    theme(axis.title.x = element_text()) +
    geom_text(aes(label = batch_shortname)) +
    labs(x = "Batch Annotation",
        fill = "Batch Annotation") +
    theme(legend.position = "none")
p9_qb <- p_virus_brain/p_axis_virus_brain +
    plot_layout(heights = c(8,
        1)) + scale_fill_igv()

pt9 <- plot_grid(p9_q, p9_qb, rel_widths = c(1.7,
    1.3))
```

```
Warning: 'position_stack()' requires non-overlapping x intervals
```

```
pt9
```



Total reads in absolute value.

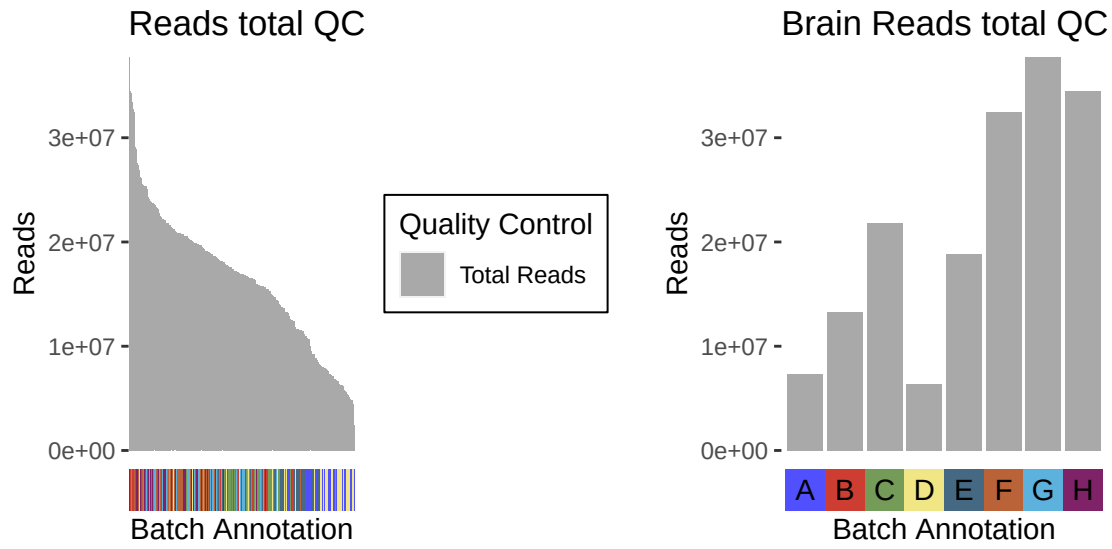
```
reads_total_brain <- reads_total[c(13,
    42, 77, 106, 141, 194, 222,
    256), ]

p_reads_t <- ggplot(reads_total,
```

```
    aes(x = reorder(sample, -reads),
        y = reads, fill = variable,
        width = 1.05)) + geom_bar(stat = "identity") +
    scale_y_continuous(expand = c(0,
        0)) + labs(x = NULL, y = "Reads",
    title = "Reads total QC", fill = "Quality Control") +
    theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.ticks.length.x = unit(0,
            "pt")) + theme(panel.background = element_rect(fill = "transparent"),
    plot.background = element_rect(fill = "transparent",
        color = NA), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    legend.background = element_rect(fill = "transparent"),
    legend.box.background = element_rect(fill = "transparent")) +
    scale_fill_manual(values = c("darkgrey"),
        labels = c("Total Reads"))

p_axis_reads_t <- ggplot(reads_total,
    aes(x = reorder(sample, -reads),
        y = factor(1), fill = batch_shortname)) +
    geom_tile(width = 1) + theme_void() +
    theme(axis.title.x = element_text(),
        legend.position = c(1.09,
            2.2)) + labs(x = "Batch Annotation",
    fill = "Batch Annotation") +
    theme(legend.position = "none") +
    scale_fill_igv()

p10_q <- p_reads_t/p_axis_reads_t +
    plot_layout(heights = c(8,
        1))

# Brain

p_reads_total_brain <- ggplot(reads_total_brain,
    aes(x = sample, y = reads,
        fill = "none")) + geom_bar(stat = "identity") +
    scale_y_continuous(expand = c(0,
        0)) + labs(x = NULL, y = "Reads",
    title = "Brain Reads total QC") +
    theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.ticks.length.x = unit(0,
            "pt")) + theme(panel.background = element_rect(fill = "transparent"),
    plot.background = element_rect(fill = "transparent",
        color = NA), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    legend.background = element_rect(fill = "transparent"),
    legend.box.background = element_rect(fill = "transparent")) +
    theme(legend.position = "none") +
    scale_fill_manual(values = c("darkgrey"))
```

```
p_axis_total_brain <- ggplot(reads_total_brain,
    aes(x = sample, y = factor(1),
        fill = batch_shortname)) +
    geom_tile(width = 1) + geom_text(aes(label = batch_shortname)) +
    theme_void() + theme(axis.title.x = element_text()) +
    labs(x = "Batch Annotation") +
    theme(legend.position = "none") +
    scale_fill_igv()

p10_qb <- p_reads_total_brain/p_axis_total_brain +
    plot_layout(heights = c(8,
        1))
pt10 <- plot_grid(p10_q, p10_qb,
    rel_widths = c(1.7, 1.3))
```

```
Warning: 'position_stack()' requires non-overlapping x intervals
```

```
pt10
```



Expression Matrix

Builiding a expression matrix for microRNA data

We builded a EM from files obtained from srnaToolBox. We did it for each batch, and joined them in a complete expression matrix.

```
setwd("~/Exosomas/QC/Expresion/Expresion_180322-190603/")
filenames <- list.files(pattern = "*.tsv")  # Read data
all_files <- lapply(filenames,
    function(x) {
        # All files in a list
```

```
        read.table(file = x, sep = "\t",
            header = TRUE)
    })
list3 <- lapply(all_files, "[",
    c(1, 3))  # I need these 2 columns
list4 <- rbindlist(list3, fill = TRUE)  # Binding all DF from the list
list4 <- lapply(list3, function(x) x[!duplicated(x$name),
    ])  # Removing duplicates
matrixA <- rbindlist(list4, fill = TRUE)
matrixA[is.na(matrixA)] = 0
matrixA <- ddply(matrixA, .(name),
    numcolwise(sum))

write.table(matrixA, file = "~/Exosomas/QC/Expresion/Matrices de Expresion/matrixA.csv",
    quote = TRUE, sep = ",")

setwd("~/Exosomas/QC/Expresion/Expresion_180626-180628-190531/")
filenames <- list.files(pattern = "*.tsv")
all_files <- lapply(filenames,
    function(x) {
        read.table(file = x, sep = "\t",
            header = TRUE)
    })
list3 <- lapply(all_files, "[",
    c(1, 3))
list4 <- rbindlist(list3, fill = TRUE)
list4 <- lapply(list3, function(x) x[!duplicated(x$name),
    ])
matrixB <- rbindlist(list4, fill = TRUE)
matrixB[is.na(matrixB)] = 0
matrixB <- ddply(matrixB, .(name),
    numcolwise(sum))

write.table(matrixB, file = "~/Exosomas/QC/Expresion/Matrices de Expresion/matrixB.csv",
    quote = TRUE, sep = ",")

setwd("~/Exosomas/QC/Expresion/Expresion_180726-190530/")
filenames <- list.files(pattern = "*.tsv")
all_files <- lapply(filenames,
    function(x) {
        read.table(file = x, sep = "\t",
            header = TRUE)
    })
list3 <- lapply(all_files, "[",
    c(1, 3))
list4 <- rbindlist(list3, fill = TRUE)
list4 <- lapply(list3, function(x) x[!duplicated(x$name),
    ])
matrixC <- rbindlist(list4, fill = TRUE)
matrixC[is.na(matrixC)] = 0
matrixC <- ddply(matrixC, .(name),
    numcolwise(sum))
```

```
write.table(matrixC, file = "~/Exosomas/QC/Expresion/Matrices de Expresion/matrixC.csv",
    quote = TRUE, sep = ",")

setwd("~/Exosomas/QC/Expresion/Expresion_180913-190529/")
filenames <- list.files(pattern = "*.tsv")
all_files <- lapply(filenames,
    function(x) {
        read.table(file = x, sep = "\t",
            header = TRUE)
    })
list3 <- lapply(all_files, "[",
    c(1, 3))
list4 <- rbindlist(list3, fill = TRUE)
list4 <- lapply(list3, function(x) x[!duplicated(x$name),
    ])
matrixD <- rbindlist(list4, fill = TRUE)
matrixD[is.na(matrixD)] = 0
matrixD <- ddply(matrixD, .(name),
    numcolwise(sum))

write.table(matrixD, file = "~/Exosomas/QC/Expresion/Matrices de Expresion/matrixD.csv",
    quote = TRUE, sep = ",")

setwd("~/Exosomas/QC/Expresion/Expresion_180919-190527/")
filenames <- list.files(pattern = "*.tsv")
all_files <- lapply(filenames,
    function(x) {
        read.table(file = x, sep = "\t",
            header = TRUE)
    })
list3 <- lapply(all_files, "[",
    c(1, 3))
list4 <- rbindlist(list3, fill = TRUE)
list4 <- lapply(list3, function(x) x[!duplicated(x$name),
    ])
matrixE <- rbindlist(list4, fill = TRUE)
matrixE[is.na(matrixE)] = 0
matrixE <- ddply(matrixE, .(name),
    numcolwise(sum))
write.table(matrixE, file = "~/Exosomas/QC/Expresion/Matrices de Expresion/matrixE.csv",
    quote = TRUE, sep = ",")


setwd("~/Exosomas/QC/Expresion/Expresion_181003-190524/")
filenames <- list.files(pattern = "*.tsv")
all_files <- lapply(filenames,
    function(x) {
        read.table(file = x, sep = "\t",
            header = TRUE)
    })
list3 <- lapply(all_files, "[",
    c(1, 3))
list4 <- rbindlist(list3, fill = TRUE)
```

```
list4 <- lapply(list3, function(x) x[!duplicated(x$name),
    ])
matrixF <- rbindlist(list4, fill = TRUE)
matrixF[is.na(matrixF)] = 0
matrixF <- ddply(matrixF, .(name),
    numcolwise(sum))
write.table(matrixF, file = "~/Exosomas/QC/Expresion/Matrices de Expresion/matrixF.csv",
    quote = TRUE, sep = ",")


setwd("~/Exosomas/QC/Expresion/Expresion_181004-190521/")
filenames <- list.files(pattern = "*.tsv")
all_files <- lapply(filenames,
    function(x) {
        read.table(file = x, sep = "\t",
            header = TRUE)
    })
list3 <- lapply(all_files, "[",
    c(1, 3))
list4 <- rbindlist(list3, fill = TRUE)
list4 <- lapply(list3, function(x) x[!duplicated(x$name),
    ])
matrixG <- rbindlist(list4, fill = TRUE)
matrixG[is.na(matrixG)] = 0
matrixG <- ddply(matrixG, .(name),
    numcolwise(sum))
write.table(matrixG, file = "~/Exosomas/QC/Expresion/Matrices de Expresion/matrixG.csv",
    quote = TRUE, sep = ",")


setwd("~/Exosomas/QC/Expresion/Expresion_181023-190319/")
filenames <- list.files(pattern = "*.tsv")
all_files <- lapply(filenames,
    function(x) {
        read.table(file = x, sep = "\t",
            header = TRUE)
    })
list3 <- lapply(all_files, "[",
    c(1, 3))
list4 <- rbindlist(list3, fill = TRUE)
list4 <- lapply(list3, function(x) x[!duplicated(x$name),
    ])
matrixH <- rbindlist(list4, fill = TRUE)
matrixH[is.na(matrixH)] = 0
matrixH <- ddply(matrixH, .(name),
    numcolwise(sum))
write.table(matrixH, file = "~/Exosomas/QC/Expresion/Matrices de Expresion/matrixH.csv",
    quote = TRUE, sep = ",")


setwd("~/Exosomas/QC/Expresion/Matrices de Expresion/")
filenames <- list.files(pattern = "*.csv")
all_files <- lapply(filenames,
```

```
    function(x) {
        read.delim(file = x, sep = ",",
            header = TRUE, stringsAsFactors = FALSE)
    })
matrixX <- rbindlist(all_files,
    fill = TRUE)
matrixX <- matrixX %>%
    mutate_if(is.integer, as.numeric)
matrixX[is.na(matrixX)] = 0
matrixX <- ddply(matrixX, .(name),
    numcolwise(sum))
matrixX <- matrixX[apply(matrixX[,
    -1], 1, function(x) !all(x ==
    0)), ]

p <- "~/Exosomas/QC/Expresion/Matrices de Expresion/Expression matrix/matrixX.csv"
write.table(matrixX, file = p,
    quote = TRUE, sep = ",")

file.remove("matrixA.csv", "matrixB.csv",
    "matrixC.csv", "matrixD.csv",
    "matrixE.csv", "matrixF.csv",
    "matrixG.csv", "matrixH.csv")
```

```
[1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
t <- "~/Exosomas/QC/Expresion/Matrices de Expresion/Expression matrix/matrixX.csv"
Matrix <- read.table(file = t,
    header = TRUE, sep = ",")

tail(Matrix[, 1:4], n = 5)
```

```
              name X180322.190603_Brain_S36 X32140243_EXO_S2 X32141753_EXO_S10
1922   hsa-miR-9985                     269              30                6
1923 hsa-miR-99a-3p                       5               0                0
1924 hsa-miR-99a-5p                   31837           13099              792
1925 hsa-miR-99b-3p                    1387             388               25
1926 hsa-miR-99b-5p                   55305           49736             3282
```

Genome Distribution

Building a expression matrix for Genome distribution

We builded a new expression matrix to check the average lenght of the microRNA reads.

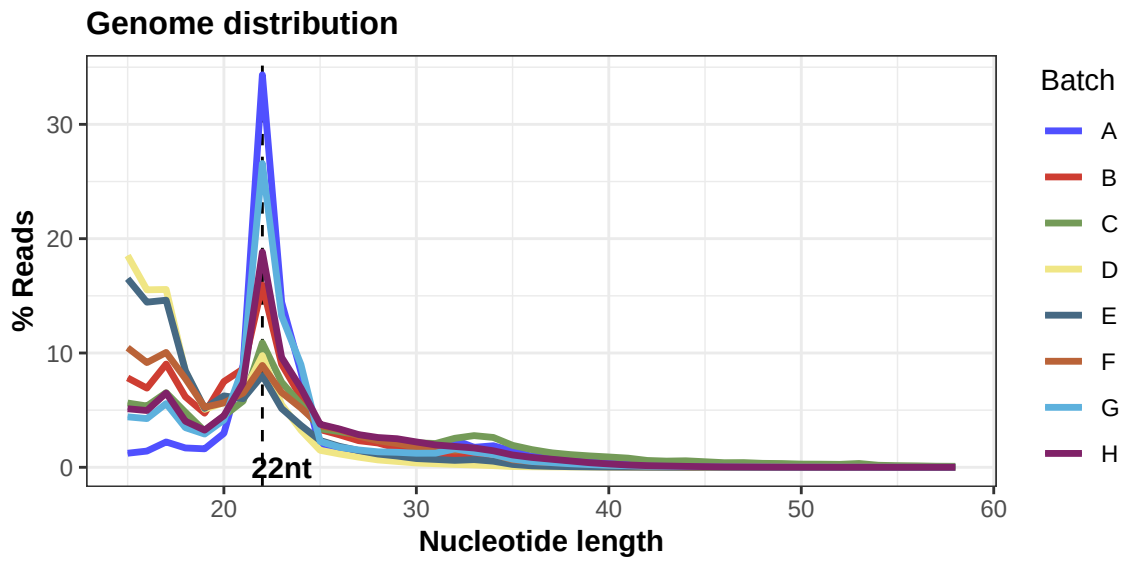
We are looking for 21-23 nucleotides reads, which is the cannonical lenght for them.

Genome Distribution

Plotting the lenght distribution of reads by batch

```
readLen_RC_A$A = rowMeans(readLen_RC_A[,
    c(2, 36)])
readLen_RC_B$B = rowMeans(readLen_RC_B[,
    c(2, 32)])
readLen_RC_C$C = rowMeans(readLen_RC_C[,
    c(2, 36)])
readLen_RC_D$D = rowMeans(readLen_RC_D[,
    c(2, 36)])
readLen_RC_E$E = rowMeans(readLen_RC_E[,
    c(2, 35)])
readLen_RC_F$F = rowMeans(readLen_RC_F[,
    c(2, 34)])
readLen_RC_G$G = rowMeans(readLen_RC_G[,
    c(2, 35)])
readLen_RC_H$H = rowMeans(readLen_RC_H[,
    c(2, 29)])

readLen <- data.frame(readLen_RC_A$Read_Length_nt,
    readLen_RC_A$A, readLen_RC_B$B,
    readLen_RC_C$C, readLen_RC_D$D,
    readLen_RC_E$E, readLen_RC_F$F,
    readLen_RC_G$G, readLen_RC_H$H)
names(readLen) <- c("Length", "A",
    "B", "C", "D", "E", "F", "G",
    "H")

readLen_long <- data.frame(reshape::melt(readLen,
    id.vars = "Length"))

colnames(readLen_long) <- c("Length",
    "Batch", "Value")

ggplot(readLen_long, aes(x = Length,
    y = Value, color = Batch)) +
    geom_line() + scale_color_igv() +
    geom_vline(xintercept = 22,
        linetype = "dashed") +
    annotate("text", x = 23, y = 0,
        label = "22nt", angle = 0,
        fontface = "bold") + theme_bw() +
    labs(x = "Nucleotide length",
        y = "% Reads", title = "Genome distribution") +
    geom_line(size = 1.2) + theme(axis.title = element_text(face = "bold"),
    plot.title = element_text(size = 12,
        face = "bold"))
```

```
Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use ‘linewidth‘ instead.
```



Multi dimensional Scaling

We observed a possible batch effect in the reads distribution, so we performed a MDS using DESeq2 package.

```
[1] TRUE
```

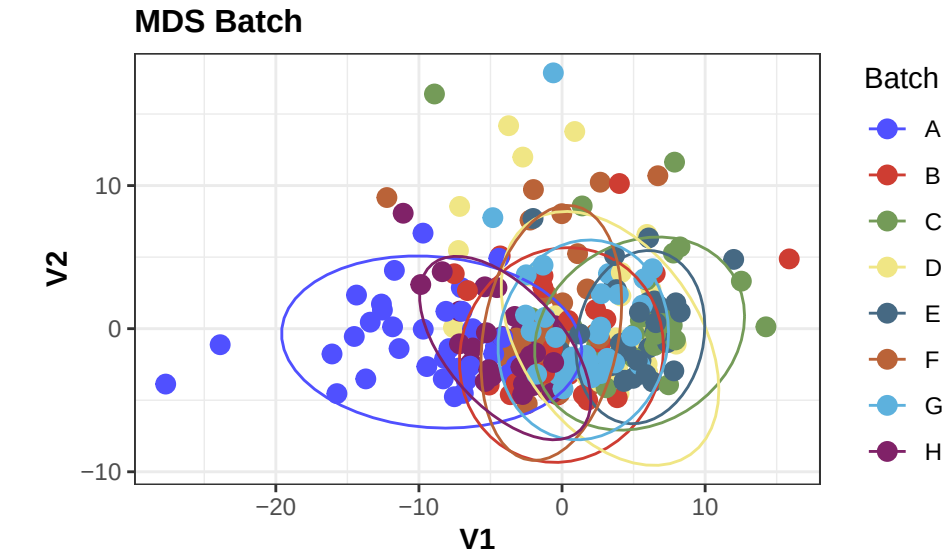
DESeq2 object

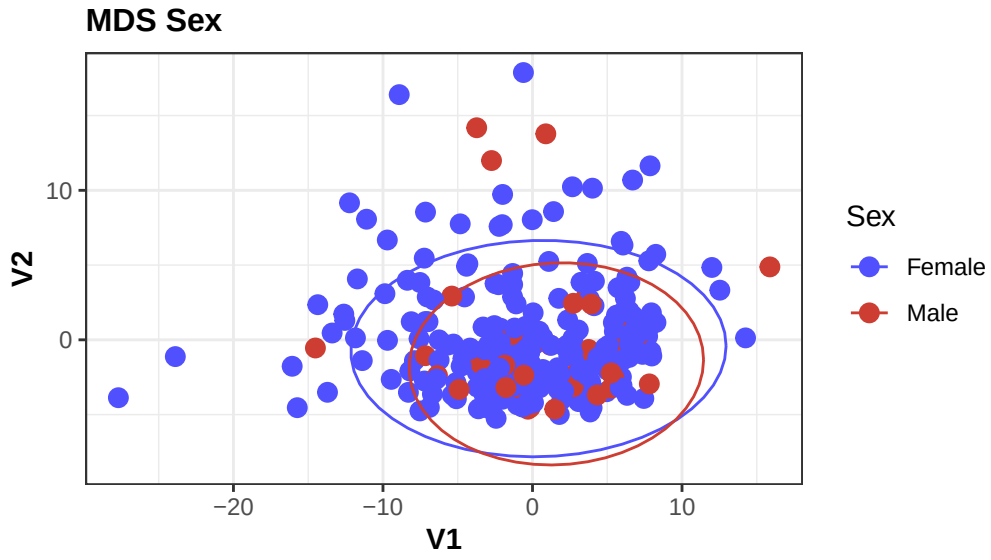
```
# Filtering by 50 counts per
# million and 20 samples

counts.CPM <- cpm(matriz2)
thresh <- counts.CPM > 50
keep <- rowSums(thresh) >= 20
counts.keep <- matriz2[keep, ]

# Design matrix

design <- model.matrix(~0 + cross2$Diagnosis)

# DEseq2 object

dds <- DESeqDataSetFromMatrix(counts.keep,
    colData = cross2, design = design)
```

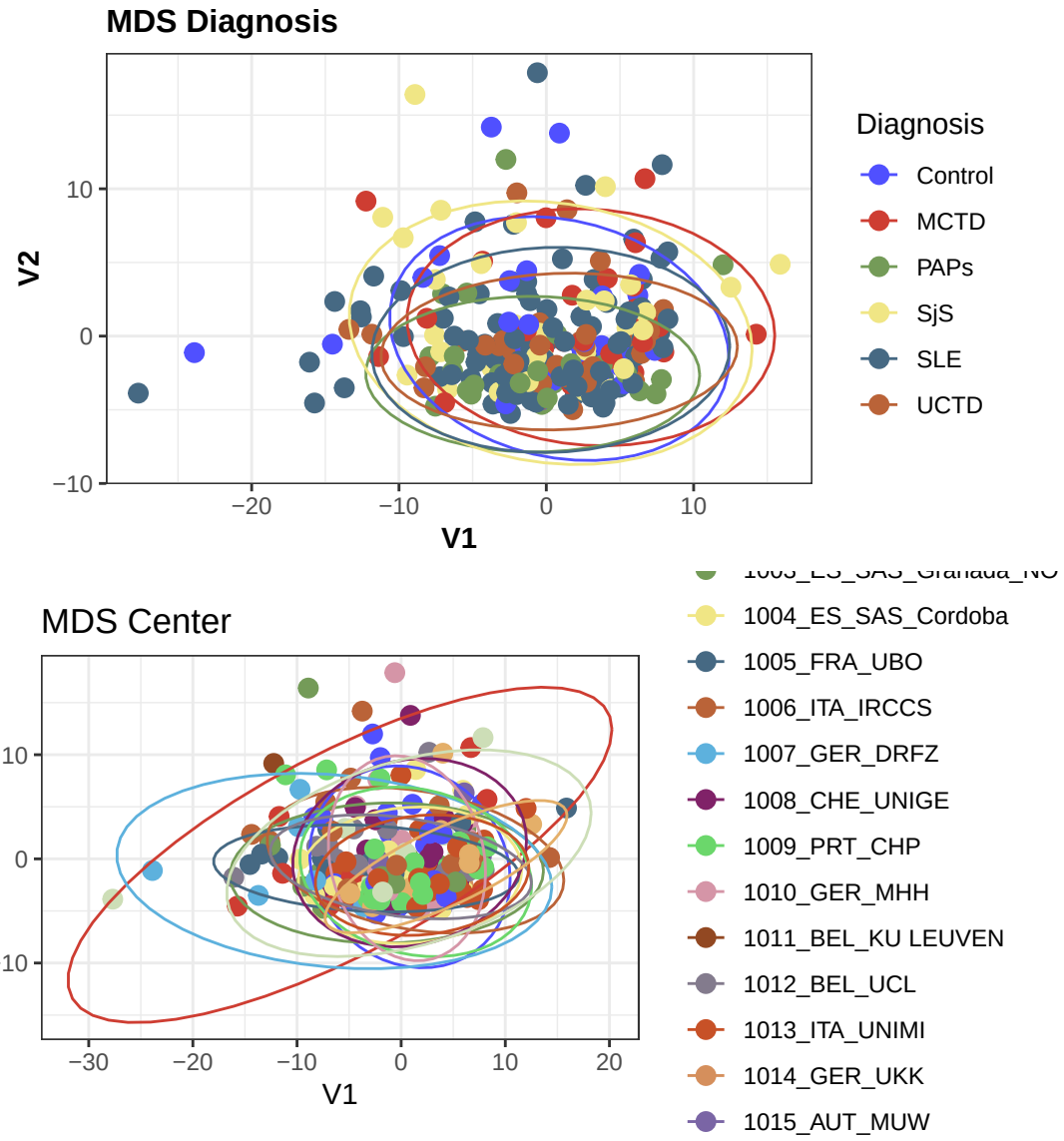
```
converting counts to integer mode
```

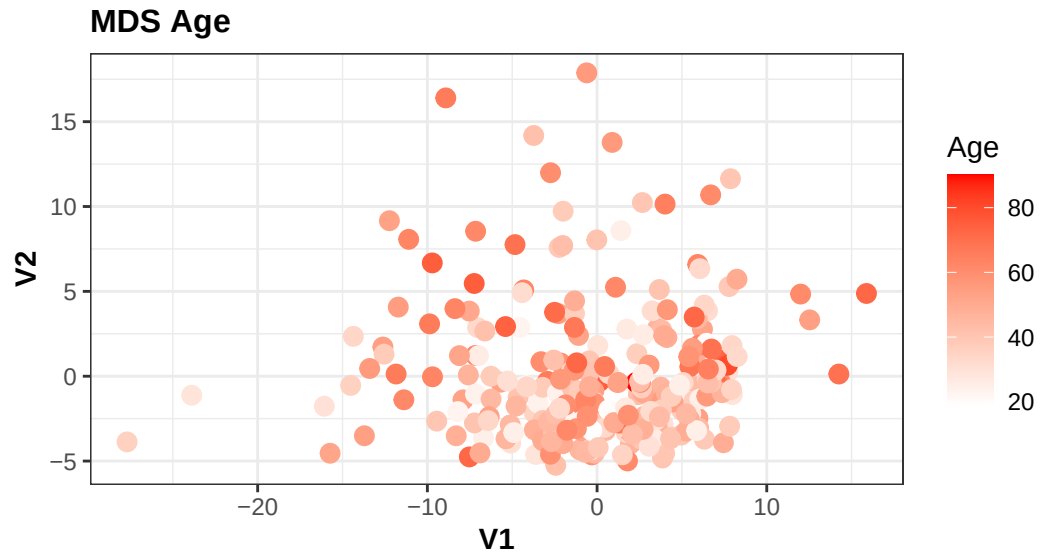
Multi dimenstional scaling plots

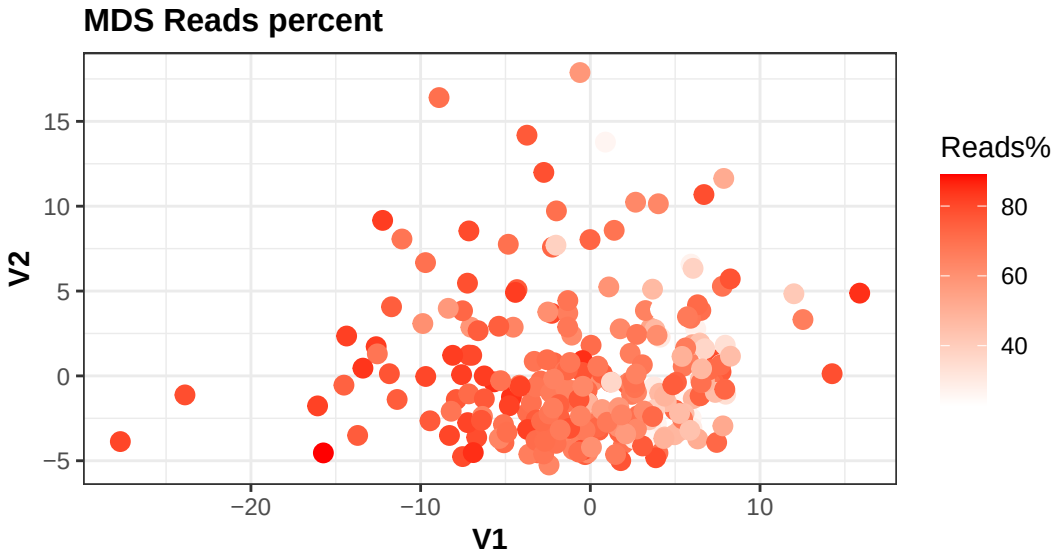
Plot of all the quality features to search for possible batch effect.

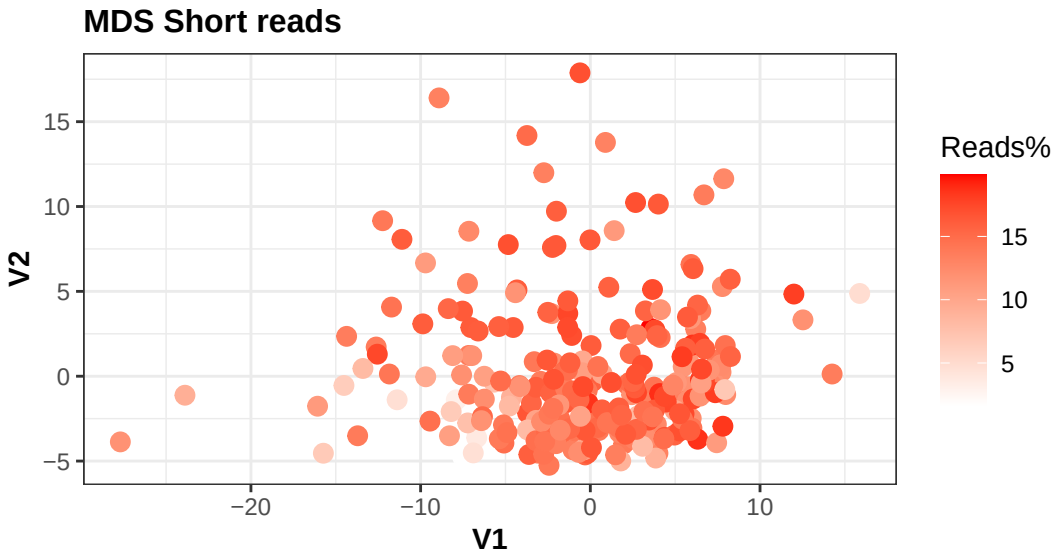

MDS Batch
Batch
A
B
C
D
E
F
G
H
V1
V2
10
0
−10
−20
−10
0
10

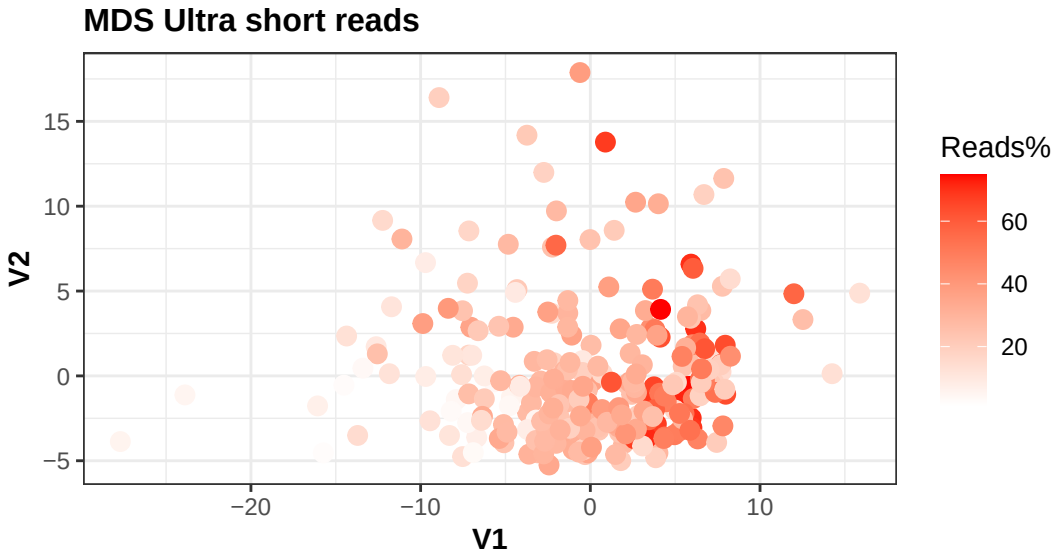

MDS Sex
Sex
Female
Male
V1
V2
10
0
−20
−10
0
10

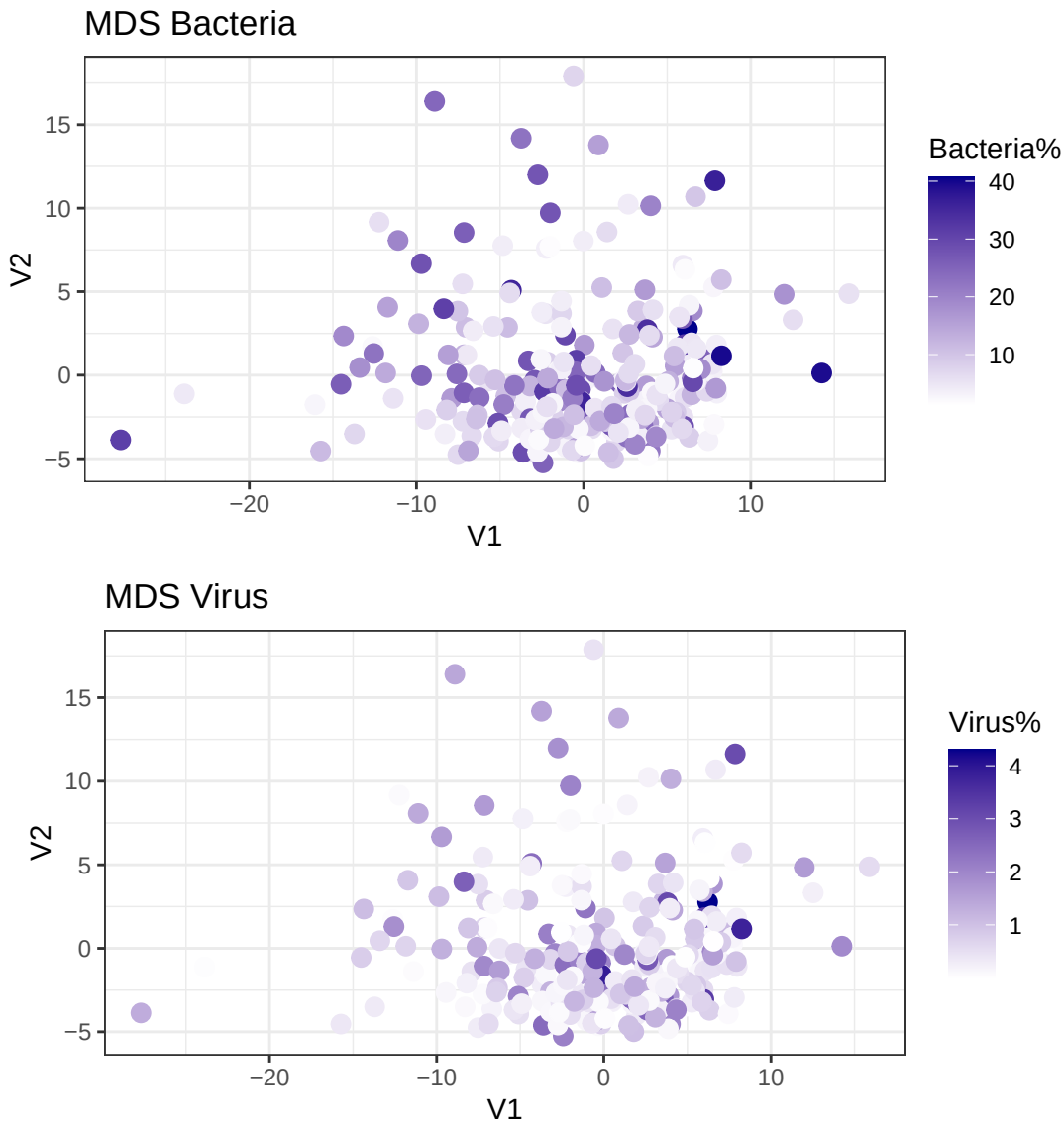

MDS Diagnosis
V2
V1
10
0
−10
−20
−10
0
10
Diagnosis
Control
MCTD
PAPs
SjS
SLE
UCTD
MDS Center
V2
V1
10
0
−10
−30
−20
−10
0
10
20
1004_ES_SAS_Cordoba
1005_FRA_UBO
1006_ITA_IRCCS
1007_GER_DRFZ
1008_CHE_UNIGE
1009_PRT_CHP
1010_GER_MHH
1011_BEL_KU LEUVEN
1012_BEL_UCL
1013_ITA_UNIMI
1014_GER_UKK
1015_AUT_MUW


MDS Age
15
10
5
0
−5
V2
−20
−10
0
10
V1
Age
80
60
40
20


MDS Reads percent
15
10
5
0
−5
V2
−20
−10
0
10
V1
Reads%
80
60
40


MDS Short reads
V2
V1
15
10
5
0
−5
−20
−10
0
10
Reads%
15
10
5


MDS Ultra short reads
V2
V1
15
10
5
0
−5
−20
−10
0
10
Reads%
60
40
20



Principal components correlation

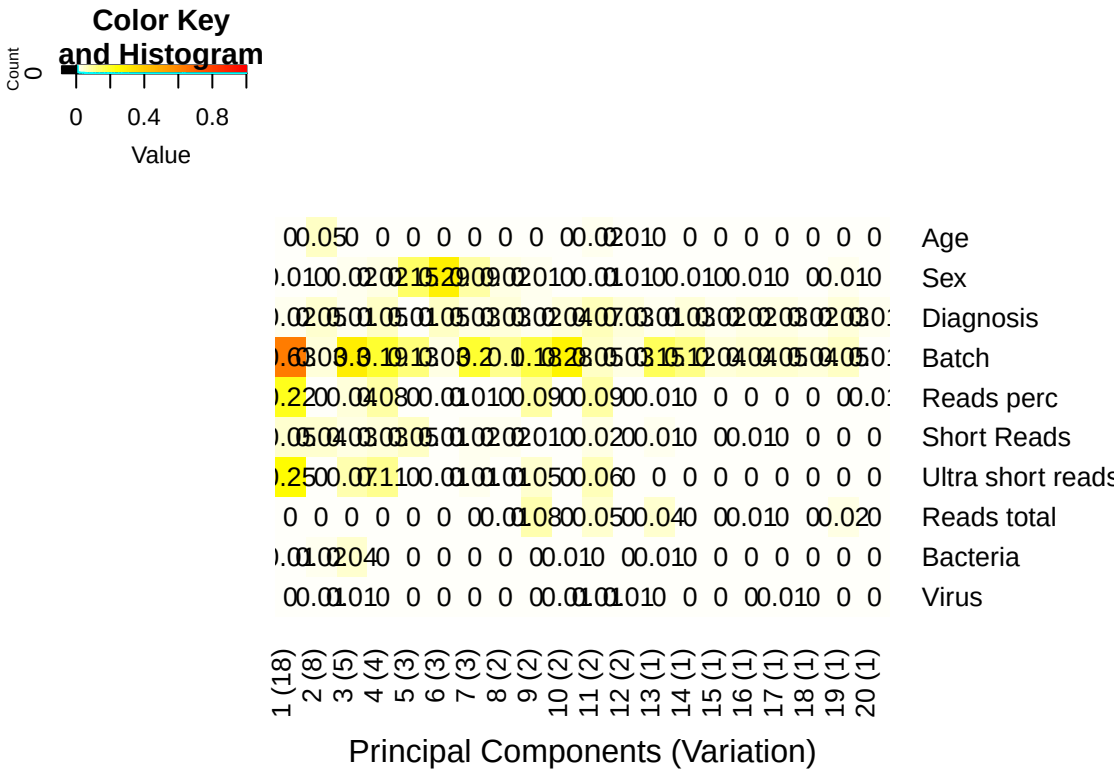
Finally we performed a test to check the possible correlation between variables and find the real origin of the batch effect.

```
cross3 <- data.frame(cross2$Age,
    cross2$Gender, cross2$Diagnosis,
    cross2$batch_shortname, cross2$Reads,
    cross2$Short_reads, cross2$Ultra_short_reads,
    cross2$reads_total, cross2$Bacteria,
    cross2$Virus)
rownames(cross3) <- cross2$sample
colnames(cross3) <- c("Age", "Gender",
    "Diagnosis", "Batch", "Reads perc",
    "Short Reads", "Ultra short reads",
    "Reads total", "Bacteria",
```

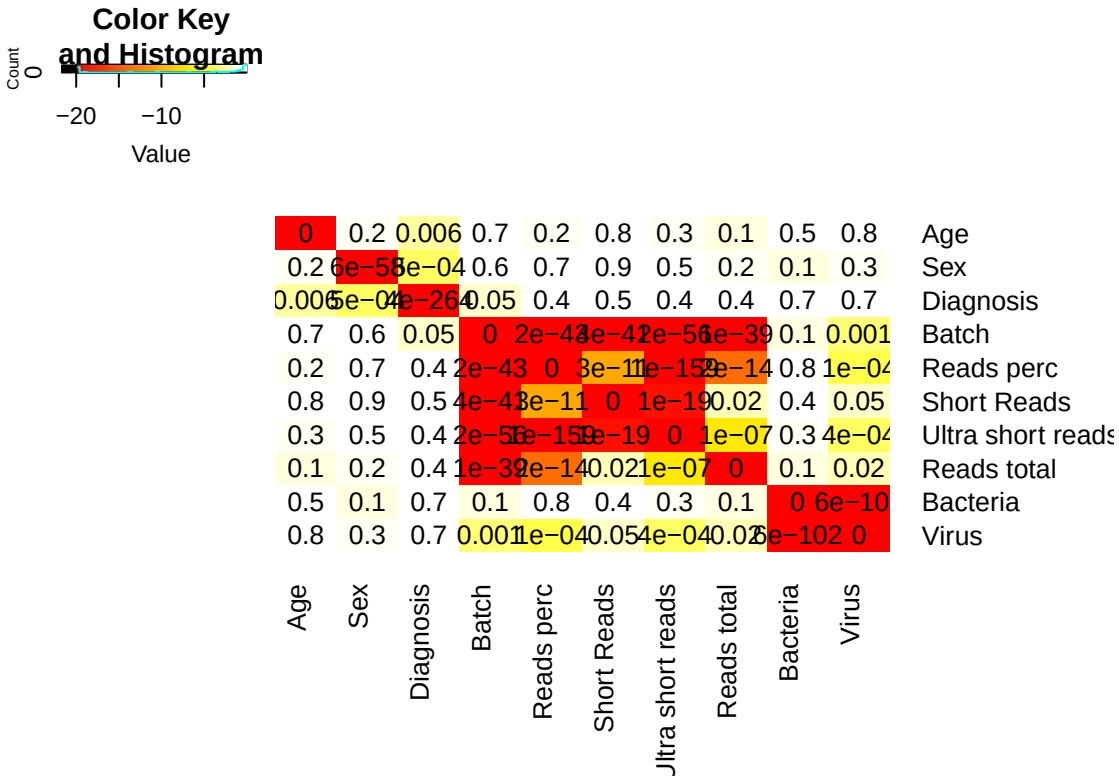
```
    "Virus")

all.equal(colnames(assay(vsd)),
    rownames(cross3))
```

```
[1] TRUE
```

```
cross3$Gender <- as.factor(cross3$Gender)
cross3$Age <- as.numeric(cross3$Age)
cross3$Diagnosis <- as.factor(cross3$Diagnosis)
names(cross3)[names(cross3) ==
    "Gender"] <- "Sex"

pr_out <- prince(assay(vsd), cross3,
    top = 20, center = TRUE)
prince.plot(pr_out, note = TRUE,
    Rsquared = TRUE)
```




```
confounding(cross3, method = "chisq")
```




```
$p.values
                          Age          Sex     Diagnosis        Batch
Age               0.000000000 1.590022e-01  5.580716e-03 7.321726e-01
Sex               0.159002167 6.300377e-58  4.948630e-04 5.618898e-01
Diagnosis         0.005580716 4.948630e-04 3.934711e-264 4.976318e-02
Batch             0.732172600 5.618898e-01  4.976318e-02 0.000000e+00
Reads perc        0.223722023 7.275303e-01  3.731276e-01 1.695845e-43
Short Reads       0.756531872 9.130635e-01  4.588309e-01 3.936504e-41
Ultra short reads 0.277141330 5.044514e-01  3.693463e-01 2.418471e-56
Reads total       0.102979142 1.717422e-01  3.746660e-01 1.274002e-39
Bacteria          0.481711511 1.287641e-01  6.975253e-01 1.127403e-01
Virus             0.776720238 3.126284e-01  6.599840e-01 9.708320e-04
                     Reads perc  Short Reads Ultra short reads  Reads total
Age                2.237220e-01 7.565319e-01      2.771413e-01 1.029791e-01
Sex                7.275303e-01 9.130635e-01      5.044514e-01 1.717422e-01
Diagnosis          3.731276e-01 4.588309e-01      3.693463e-01 3.746660e-01
Batch              1.695845e-43 3.936504e-41      2.418471e-56 1.274002e-39
Reads perc         0.000000e+00 2.798007e-11     1.016647e-159 2.110187e-14
Short Reads        2.798007e-11 0.000000e+00      1.276887e-19 1.716175e-02
Ultra short reads 1.016647e-159 1.276887e-19      0.000000e+00 1.324265e-07
Reads total        2.110187e-14 1.716175e-02      1.324265e-07 0.000000e+00
Bacteria           7.758546e-01 4.099047e-01      3.171362e-01 1.353835e-01
Virus              1.215356e-04 5.238477e-02      4.014704e-04 1.649828e-02
                       Bacteria         Virus
Age                4.817115e-01  7.767202e-01
Sex                1.287641e-01  3.126284e-01
Diagnosis          6.975253e-01  6.599840e-01
Batch              1.127403e-01  9.708320e-04
Reads perc         7.758546e-01  1.215356e-04
```

```
Short Reads        4.099047e-01  5.238477e-02
Ultra short reads  3.171362e-01  4.014704e-04
Reads total        1.353835e-01  1.649828e-02
Bacteria           0.000000e+00 5.921406e-102
Virus             5.921406e-102  0.000000e+00

$n
                  Age Sex Diagnosis Batch Reads perc Short Reads
Age               265 265       265   265        265         265
Sex               265 265       265   265        265         265
Diagnosis         265 265       265   265        265         265
Batch             265 265       265   265        265         265
Reads perc        265 265       265   265        265         265
Short Reads       265 265       265   265        265         265
Ultra short reads 265 265       265   265        265         265
Reads total       265 265       265   265        265         265
Bacteria          265 265       265   265        265         265
Virus             265 265       265   265        265         265
                  Ultra short reads Reads total Bacteria Virus
Age                             265         265      265   265
Sex                             265         265      265   265
Diagnosis                       265         265      265   265
Batch                           265         265      265   265
Reads perc                      265         265      265   265
Short Reads                     265         265      265   265
Ultra short reads               265         265      265   265
Reads total                     265         265      265   265
Bacteria                        265         265      265   265
Virus                           265         265      265   265

$test.function
                  Age  Sex           Diagnosis    Batch         Reads perc
Age               "lm" "lm"          "lm"         "lm"          "lm"
Sex               "lm" "chisq.test" "chisq.test" "chisq.test" "lm"
Diagnosis         "lm" "chisq.test" "chisq.test" "chisq.test" "lm"
Batch             "lm" "chisq.test" "chisq.test" "chisq.test" "lm"
Reads perc        "lm" "lm"          "lm"         "lm"          "lm"
Short Reads       "lm" "lm"          "lm"         "lm"          "lm"
Ultra short reads "lm" "lm"          "lm"         "lm"          "lm"
Reads total       "lm" "lm"          "lm"         "lm"          "lm"
Bacteria          "lm" "lm"          "lm"         "lm"          "lm"
Virus             "lm" "lm"          "lm"         "lm"          "lm"
                  Short Reads Ultra short reads Reads total Bacteria Virus
Age               "lm"        "lm"              "lm"        "lm"     "lm"
Sex               "lm"        "lm"              "lm"        "lm"     "lm"
Diagnosis         "lm"        "lm"              "lm"        "lm"     "lm"
Batch             "lm"        "lm"              "lm"        "lm"     "lm"
Reads perc        "lm"        "lm"              "lm"        "lm"     "lm"
Short Reads       "lm"        "lm"              "lm"        "lm"     "lm"
Ultra short reads "lm"        "lm"              "lm"        "lm"     "lm"
Reads total       "lm"        "lm"              "lm"        "lm"     "lm"
Bacteria          "lm"        "lm"              "lm"        "lm"     "lm"
Virus             "lm"        "lm"              "lm"        "lm"     "lm"
```

```
$classes
              Age               Sex         Diagnosis             Batch
        "numeric"          "factor"          "factor"          "factor"
       Reads perc       Short Reads Ultra short reads       Reads total
        "numeric"         "numeric"         "numeric"         "numeric"
         Bacteria             Virus
        "numeric"         "numeric"
```