Comparison of read lengths

Wolfgang Esser-Skala

2020-11-27

Load data

```
library(tidyverse)
library(Seurat)
library(future)
library(SingleR)
library(celldex)
```

These two samples differ in sequencing read length (50 and 75 bp, respectively).

```
use_samples <- c(
  "data/S_16_4503_Re_DOWN_transcriptome/filtered_feature_bc_matrix.h5",
  "data/S_16_4503_Re_transcriptome/filtered_feature_bc_matrix.h5"
)
```

Default Seurat pipeline.

```
sample_list <- map(
  use_samples,
  function(sample) {
    sample_name <- str_match(sample, "_(.*)_trans")[,2]
    sample %>%
      Read10X_h5() %>%
      CreateSeuratObject() %>%
      AddMetaData(sample_name, col.name = "sample") %>%
      NormalizeData() %>%
      FindVariableFeatures() %>%
      ScaleData() %>%
      RunPCA(verbose = FALSE) %>%
      RunUMAP(dims = 1:30) %>%
      RunTSNE(dims = 1:30) %>%
      FindNeighbors() %>%
      FindClusters()
  }
)
```

Extract metadata and cell embeddings from dimensionality reductions.

```
sample_data <- map(
  sample_list,
  function(sample) {
    list(
      sample@meta.data,
      Embeddings(sample, "tsne"),
      Embeddings(sample, "umap")
    ) %>%
      map(as_tibble, rownames = "cell") %>%
      purrr::reduce(inner_join, by = "cell")
  }
)

df <-
  full_join(
    sample_data[[1]],
    sample_data[[2]],
    by = "cell",
    suffix = c(".50", ".75")
)
```

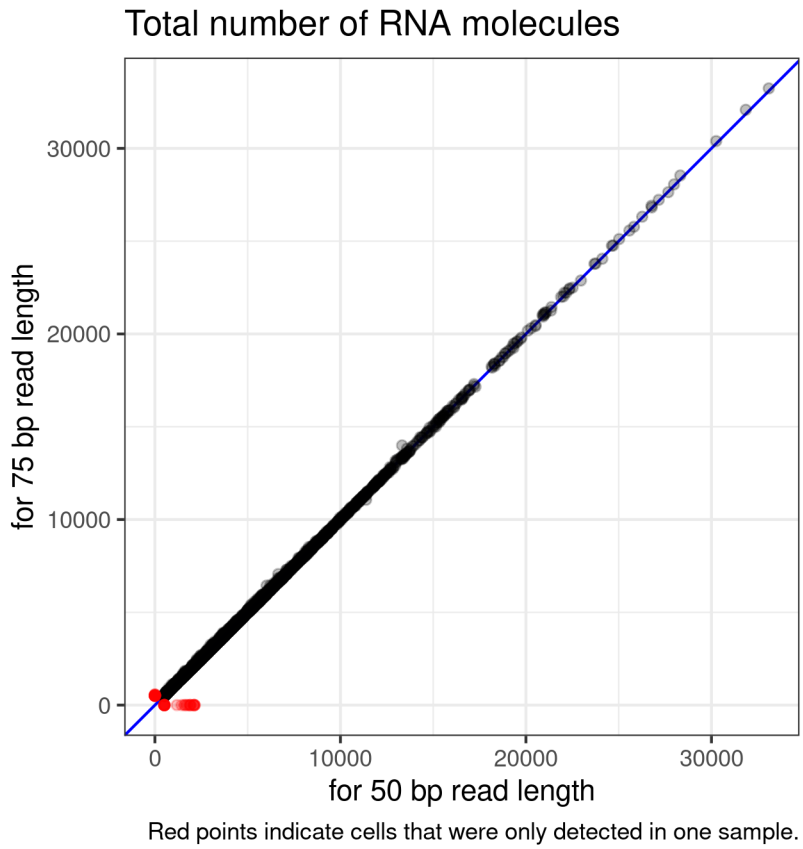
Cell types

Determine cell types via `SingleR`.

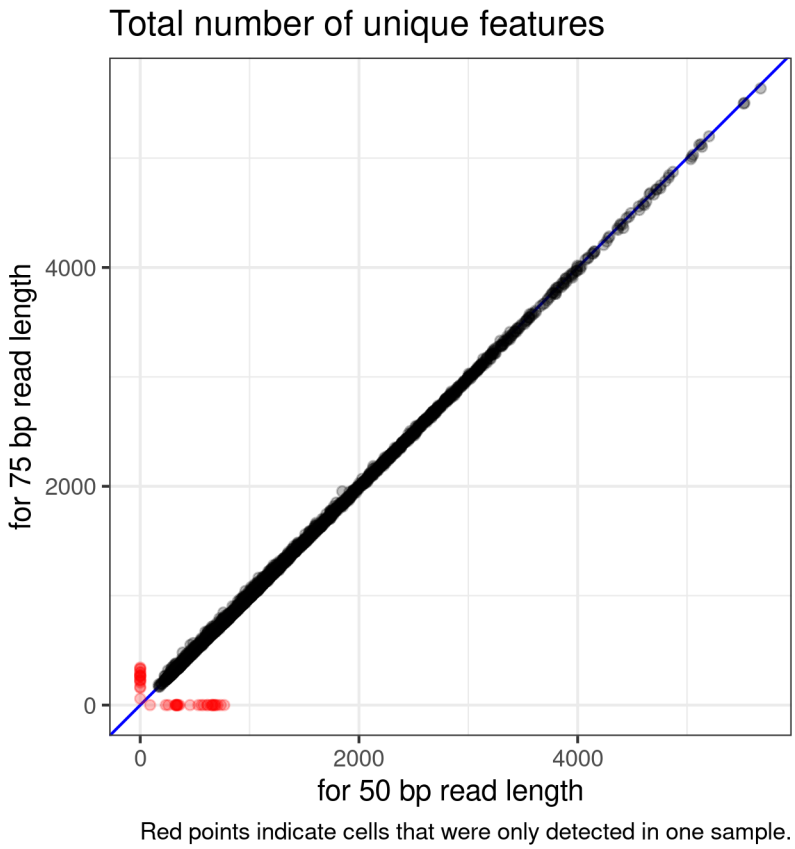
```
counts_50 <- GetAssayData(sample_list[[1]])
counts_75 <- GetAssayData(sample_list[[2]])
reference_cell_types <- HumanPrimaryCellAtlasData()

predicted_cell_types_50 <- SingleR(
  counts_50,
  reference_cell_types,
  labels = reference_cell_types$label.main
)

predicted_cell_types_75 <- SingleR(
  counts_75,
  reference_cell_types,
  labels = reference_cell_types$label.main
)

df <-
  list(
    df,
    predicted_cell_types_50 %>%
      as_tibble(rownames = "cell") %>%
      select(cell, cell_type.50 = labels),
    predicted_cell_types_75 %>%
      as_tibble(rownames = "cell") %>%
      select(cell, cell_type.75 = labels)
  ) %>%
  purrr::reduce(left_join, by = "cell")
```

Total RNA and feature counts

```
df %>%
  ggplot(aes(nCount_RNA.50, nCount_RNA.75)) +
  geom_abline(color = "blue") +
  geom_point(alpha = .25) +
  geom_point(
    data = df %>%
      filter(is.na(nCount_RNA.50) | is.na(nCount_RNA.75)) %>%
      replace_na(list(nCount_RNA.50 = 0, nCount_RNA.75 = 0)),
    color = "red",
    alpha = .25
  ) +
  coord_fixed() +
  labs(
    title = "Total number of RNA molecules",
    x = "for 50 bp read length",
    y = "for 75 bp read length",
    caption = "Red points indicate cells that were only detected in one sample."
  ) +
  theme_bw() +
  NULL
```



```r
df %>%
  ggplot(aes(nFeature_RNA.50, nFeature_RNA.75)) +
  geom_abline(color = "blue") +
  geom_point(alpha = .25) +
  geom_point(
    data = df %>%
      filter(is.na(nFeature_RNA.50) | is.na(nFeature_RNA.75)) %>%
      replace_na(list(nFeature_RNA.50 = 0, nFeature_RNA.75 = 0)),
    color = "red",
    alpha = .25
  ) +
  coord_fixed() +
  labs(
    title = "Total number of unique features",
    x = "for 50 bp read length",
    y = "for 75 bp read length",
    caption = "Red points indicate cells that were only detected in one sample."
  ) +
  theme_bw() +
  NULL
```

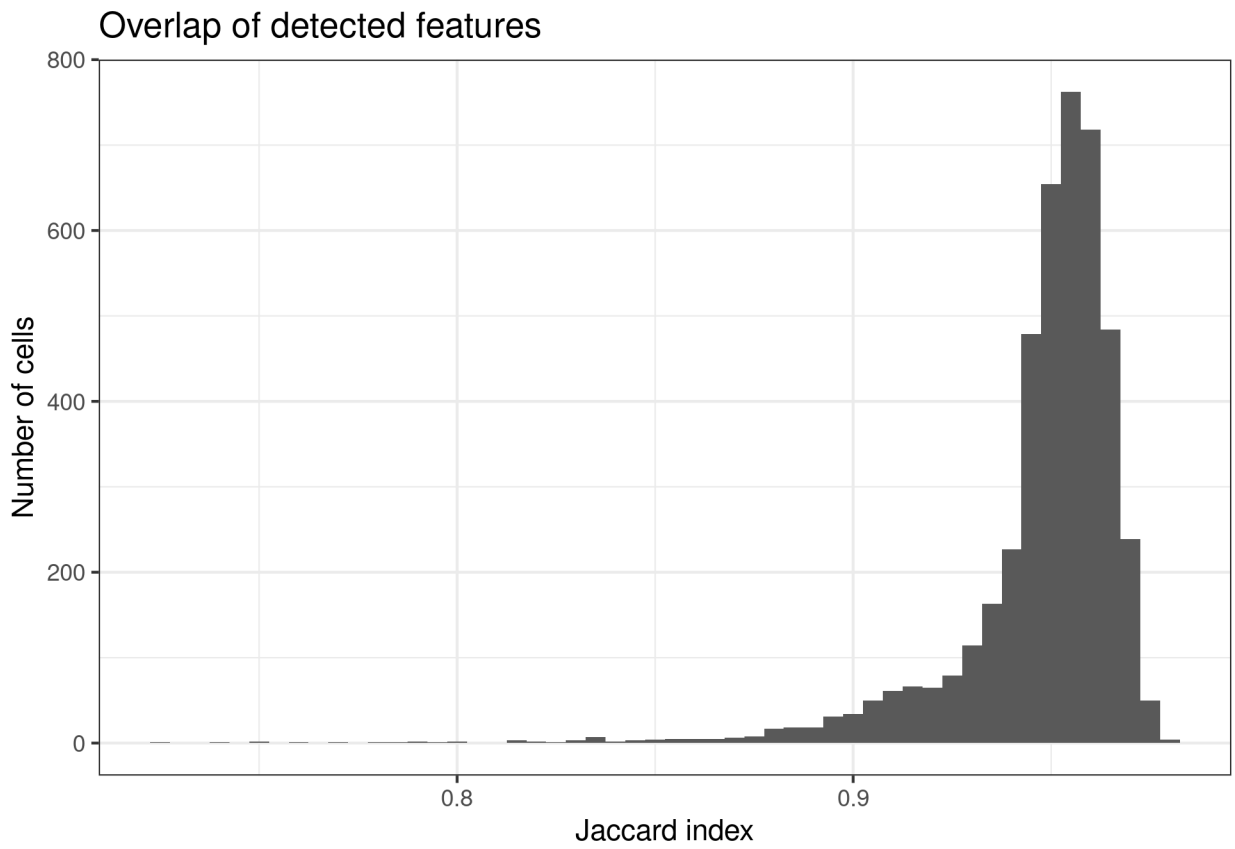


Jaccard index

Determine if the same features were detected in a cell irrespective of read length. Quantify this overlap via the Jaccard index (number of features in intersection divided by number of features in union.)

```
calc_jaccard_index <- function(c1, c2) {
  length(intersect(c1, c2)) / length(union(c1, c2))
}

common_cells <- intersect(colnames(counts_50), colnames(counts_75))

jaccard_index <-
  map_dbl(
    common_cells,
    function(cell) {
      calc_jaccard_index(
        names(which(counts_50[, cell] > 0)),
        names(which(counts_75[, cell] > 0))
      )
    }
  ) %>%
  set_names(common_cells)
```

```
jaccard_index %>%
  enframe("cell", "jaccard_index") %>%
  ggplot(aes(jaccard_index)) +
  geom_histogram(binwidth = .005) +
  labs(
    title = "Overlap of detected features",
    x = "Jaccard index",
    y = "Number of cells"
  ) +
  theme_bw() +
  NULL
```



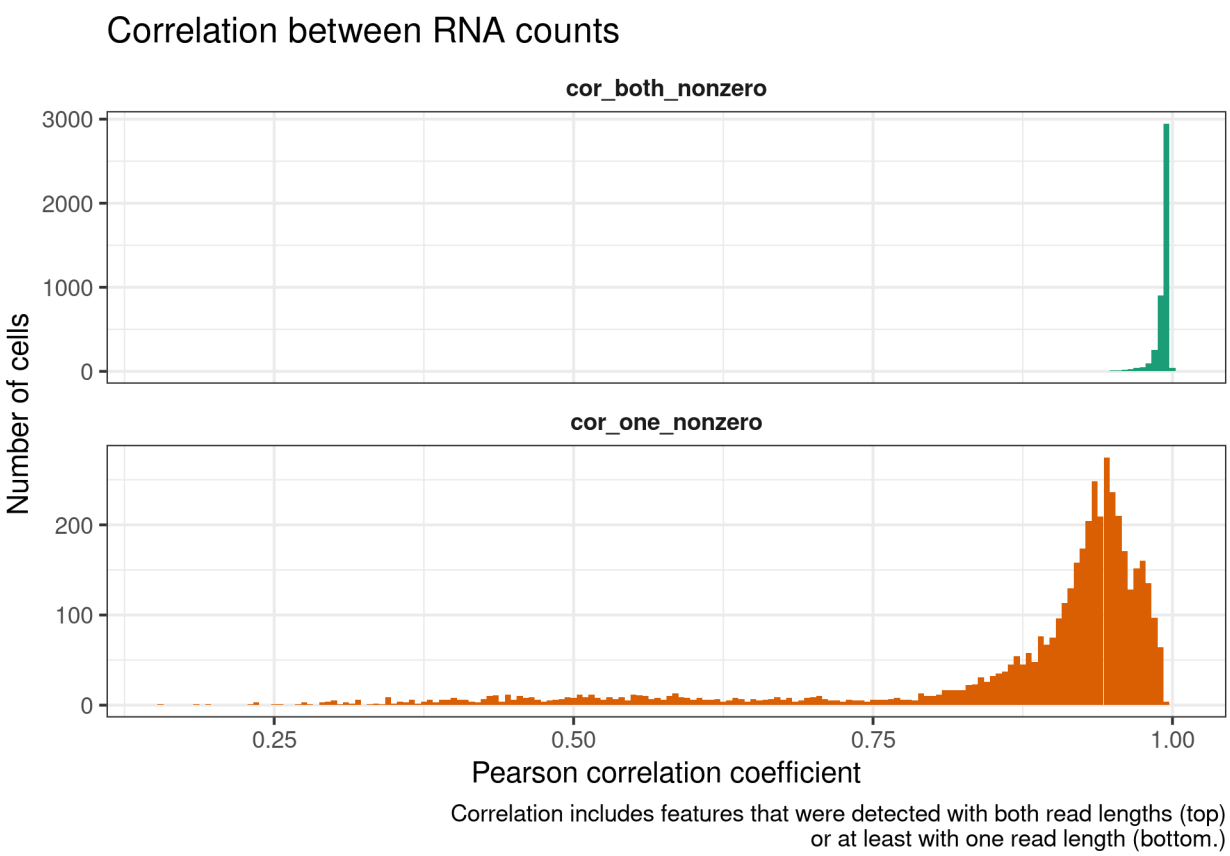
RNA count correlation

Calculate the correlation between RNA counts in each cell depending on read length.

```
cor_50_75 <-
  map_dfr(
    common_cells,
    function(cell) {
      df <-
        tibble(
          counts_50 = counts_50[, cell],
          counts_75 = counts_75[, cell]
        )
      cor_one_nonzero <-
        df %>%
        filter(counts_50 > 0 | counts_75 > 0) %>%
        cor() %>%
        magrittr::extract(2, 1)
      cor_both_nonzero <-
        df %>%
        filter(counts_50 > 0 & counts_75 > 0) %>%
        cor() %>%
        magrittr::extract(2, 1)
      tibble(
```

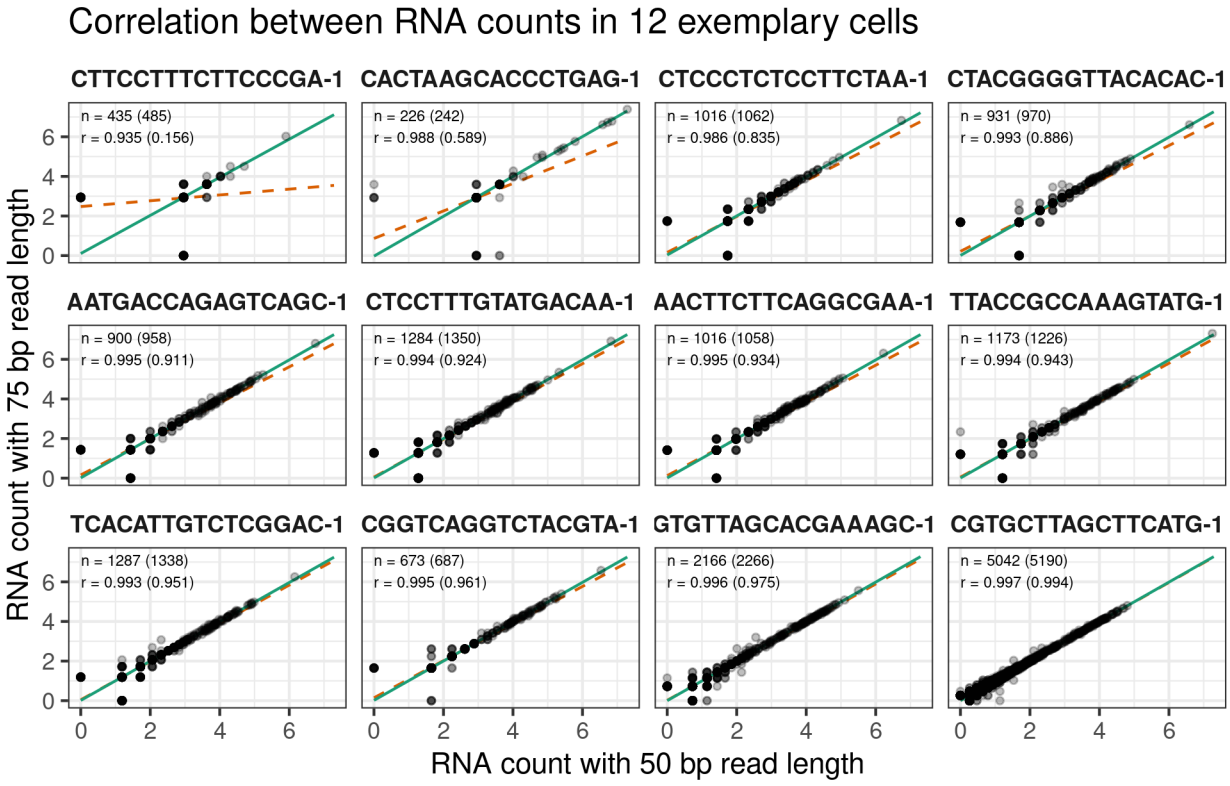
```
        cell = cell,
        cor_one_nonzero = cor_one_nonzero,
        cor_both_nonzero = cor_both_nonzero
      )
    }
  )
```

```
cor_50_75 %>%
  pivot_longer(!cell, names_to = "method", values_to = "cor") %>%
  ggplot(aes(cor, fill = method)) +
  geom_histogram(binwidth = .005, show.legend = FALSE) +
  scale_fill_manual(values = c("#1b9e77", "#d95f02")) +
  facet_wrap(vars(method), scales = "free_y", ncol = 1) +
  labs(
    title = "Correlation between RNA counts",
    x = "Pearson correlation coefficient",
    y = "Number of cells",
    caption = "Correlation includes features that were detected with both read lengths (top)\nor at leas
  ) +
  theme_bw() +
  theme(
      strip.background = element_blank(),
      strip.text = element_text(face = "bold")
    ) +
  NULL
```



```r
plot_regression <- function(cells, ncol = NULL) {
  counts <- map_dfr(
    cells,
    ~tibble(
      cell = .x,
      counts_50 = counts_50[, .x],
      counts_75 = counts_75[, .x]
    )
  )

  df_one_nonzero <-
    counts %>%
    filter(counts_50 > 0 | counts_75 > 0)
  df_both_nonzero <-
    counts %>%
    filter(counts_50 > 0 & counts_75 > 0)
  df_cor <-
    cor_50_75 %>%
    filter(cell %in% cells) %>%
    mutate(
      label = sprintf(
        "n = %i (%i)\nr = %.3f (%.3f)",
        df_both_nonzero %>% dplyr::count(cell) %>% pull(n),
        df_one_nonzero %>% dplyr::count(cell) %>% pull(n),
        cor_both_nonzero,
        cor_one_nonzero
      )
    )

  ggplot(df_one_nonzero, aes(counts_50, counts_75)) +
    geom_smooth(
      color = "#d95f02", linetype = "dashed",
      method = "lm", size = .5, fullrange = TRUE, se = FALSE
    ) +
    geom_smooth(
      data = df_both_nonzero, color = "#1b9e77",
      method = "lm", size = .5, fullrange = TRUE, se = FALSE
    ) +
    geom_point(size = 1, alpha = .25) +
    geom_text(
      data = df_cor,
      aes(label = label),
      x = 0,
      y = max(df_both_nonzero$counts_75),
      size = 2,
      hjust = 0,
      vjust = 1
    ) +
    facet_wrap(vars(as_factor(cell) %>% fct_inorder()), ncol = ncol) +
    labs(
      title = "Correlation between RNA counts in 12 exemplary cells",
      x = "RNA count with 50 bp read length",
      y = "RNA count with 75 bp read length",
```

```
      caption = "n, number of features; r, Pearson correlation;\nvalues in parentheses include features
    ) +
    theme_bw() +
    theme(
      strip.background = element_blank(),
      strip.text = element_text(face = "bold")
    ) +
    NULL
}

selected_cor_cells <-
  cor_50_75 %>%
  arrange(cor_one_nonzero) %>%
  dplyr::slice(
    seq(from = 1L, to = nrow(cor_50_75), length.out = 12) %>% as.integer()
  ) %>%
  pull(cell)

plot_regression(selected_cor_cells, ncol = 4)
```

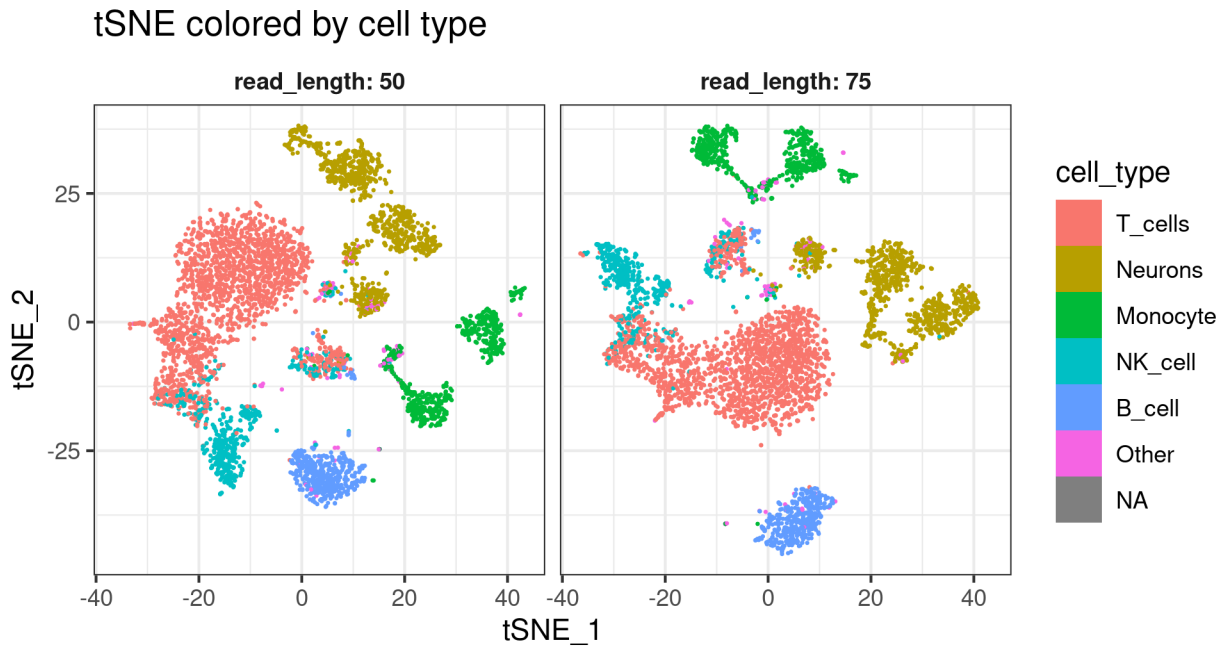


Dimensionality reductions

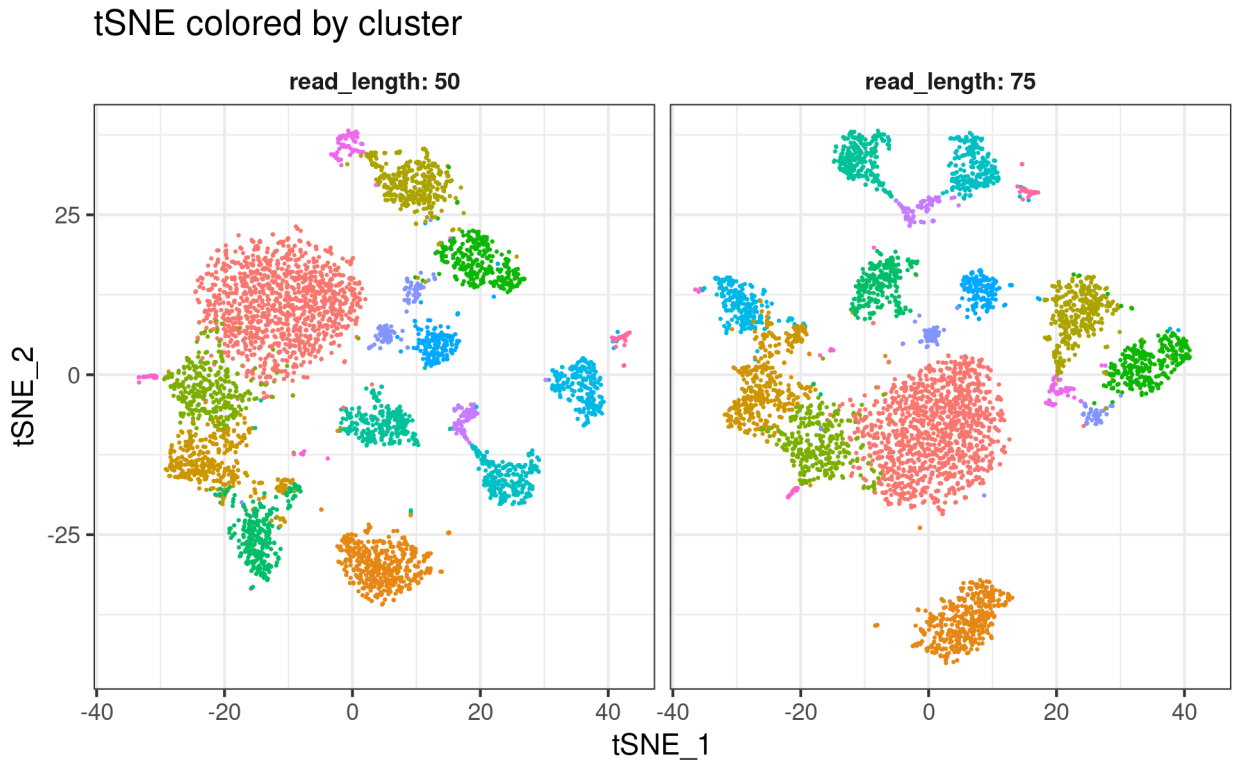
```
df_dim <-
  df %>%
  select(
    cell,
    starts_with("tSNE"),
    starts_with("UMAP"),
    starts_with("seurat"),
    starts_with("cell_type")
  ) %>%
  pivot_longer(!cell, names_to = c(".value", "read_length"), names_sep = "\\.") %>%
  mutate(cell_type = as_factor(cell_type) %>% fct_infreq() %>% fct_lump()) %>%
  {.}
```

```
dim_plot <- function(dim1, dim2, color, show_legend = TRUE, title = NULL) {
  ggplot(df_dim, aes({{dim1}}, {{dim2}})) +
    geom_point(
      aes(color = {{color}}, fill = {{color}}),
      size = .1,
      key_glyph = "rect",
      na.rm = TRUE,
      show.legend = show_legend
    ) +
    coord_fixed() +
    facet_wrap(vars(read_length), labeller = "label_both") +
    theme_bw() +
    theme(
      strip.background = element_blank(),
      strip.text = element_text(face = "bold")
    ) +
    labs(title = title) +
    NULL
}
```

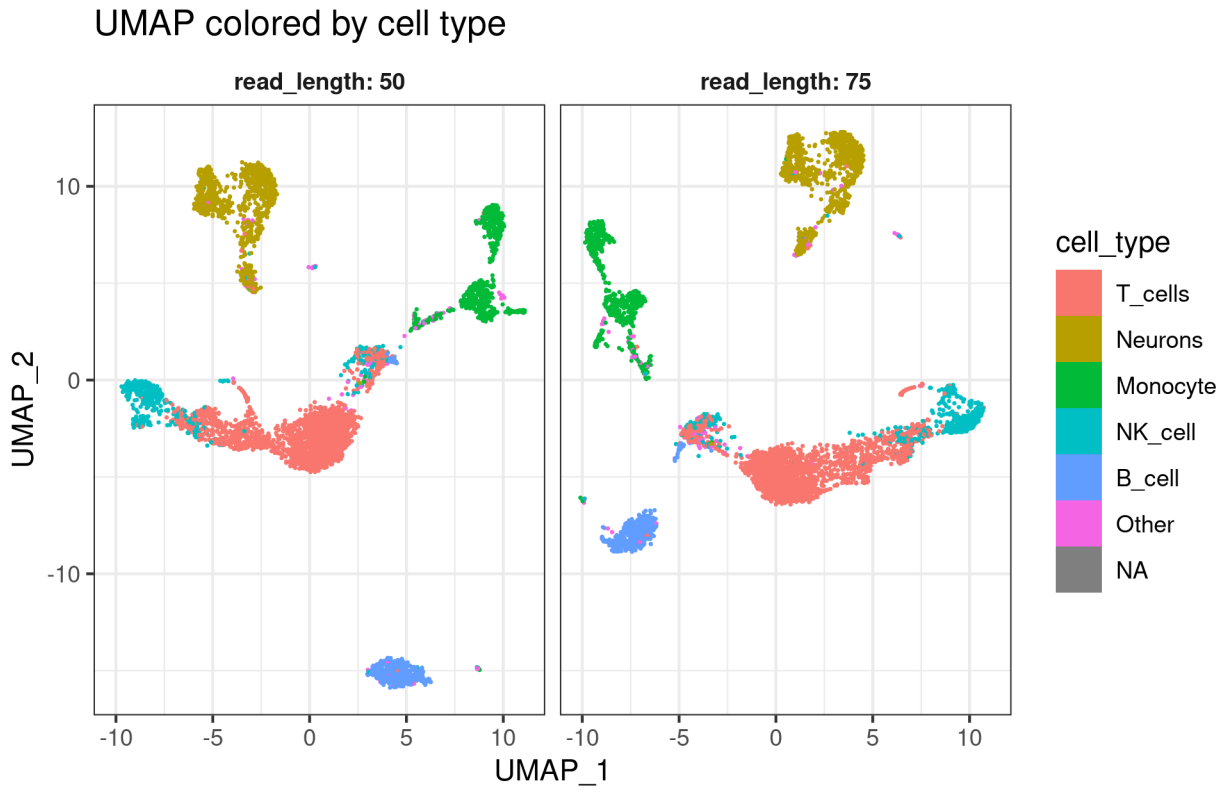
```
dim_plot(tSNE_1, tSNE_2, cell_type,
         title = "tSNE colored by cell type")
```



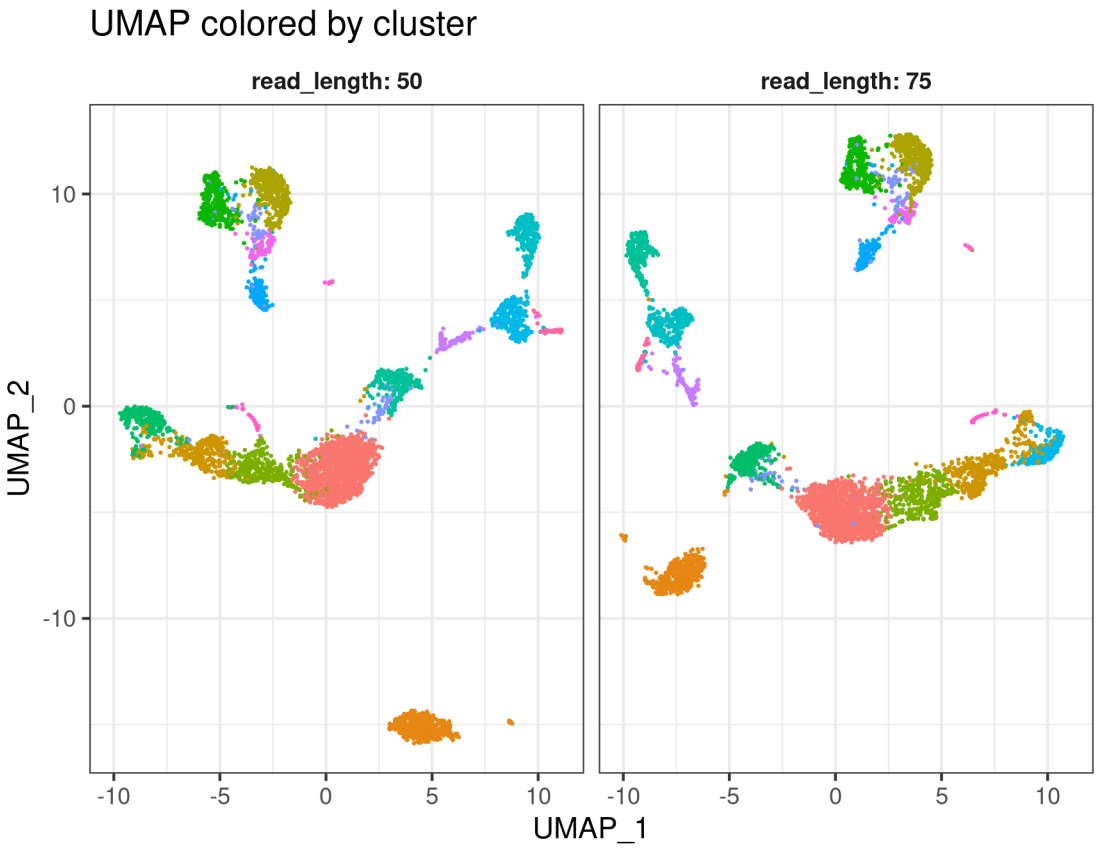

```
dim_plot(tSNE_1, tSNE_2, seurat_clusters,
         title = "tSNE colored by cluster", show_legend = FALSE)
```




```
dim_plot(UMAP_1, UMAP_2, cell_type,
         title = "UMAP colored by cell type")
```




```
dim_plot(UMAP_1, UMAP_2, seurat_clusters,
         title = "UMAP colored by cluster", show_legend = FALSE)
```


UMAP colored by cluster
read_length: 50
read_length: 75
UMAP_2
UMAP_1
10
0
-10
-10
-5
0
5
10