0) General QC metrics from parsebio pipeline

Kane Toh[*]

2023-12-18

Contents

1 Overview **1**

2 Define global variables **1**

3 Alignment with the Parse Biosciences split-pipe program **5**

4 QC assessments **5**

4.1 Phred quality scores (Q30) are similar between sublibraries (~85-90% high quality bases) . . 5

4.2 Alignment statistics are similar between sublibraries (~69% uniquely mapped reads) 7

4.3 The KO samples, with the exception of KO_BL, have higher sequencing depth than the WT samples 7

5 Summary and next steps **7**

1 Overview

This notebook summarises/visualises the QC results obtained from the parsebio pipeline, prior to loading the counts as seurat objects.

```
library(dplyr)
library(ggplot2)
```

2 Define global variables

```
# Loads in custom functions that I've written to assist with plotting/saving tasks
source("helper.R")
```

[*] Genomics and Data Analytics Core (GeDaC), Cancer Science Institute of Singapore, National University of Singapore; kanetoh@nus.edu.sg

```
# Path to saving the results fo the analysis
res_path <- "../results/00_generalQC/"
if (!dir.exists(res_path)){

  dir.create(res_path)
}else{
  print("Results directory exists")
}
```

```
## [1] "Results directory exists"
```

```
# Data path specifes where all the outputs from the parseBio split-pipe pipeline is stored.
data_path <- "../data/"
if (!dir.exists(data_path)){
  warning("Directory does not exist: ", data_path)
}else{
  print("Data directory exists")
}
```

```
## [1] "Data directory exists"
```

```
# Specifying the sublibrary paths
sublib1_path <- "../data/Sublibrary1/"
if (!dir.exists(sublib1_path)){
  warning("Directory does not exist: ", sublib1_path)
}else{
  print("Sublib1 directory exists")
}
```

```
## [1] "Sublib1 directory exists"
```

```
sublib2_path <- "../data/Sublibrary2/"
if (!dir.exists(sublib2_path)){
  warning("Directory does not exist: ", sublib2_path)
}else{
  print("Sublib2  directory exists")
}
```

```
## [1] "Sublib2  directory exists"
```

```
statistics_for_facet <- c("mm39_number_of_cells",
                            "mm39_median_tscp_per_cell",
                            "mm39_median_genes_per_cell",
                            "mean_reads_per_cell",
                            "number_of_reads",
                            "mm39_number_of_tscp",
                            "sequencing_saturation",
                          "mm39_fraction_reads_in_cells",
                          "mm39_fraction_tscp_in_cells",
                          "mm39_fraction_exonic")
```

```
statistics_fraction <- c("bc1_Q30", "bc2_Q30", "bc3_Q30",
                         "polyN_Q30","cDNA_Q30", "tso_fraction_in_read1",
                         "valid_barcode_fraction", "transcriptome_map_fraction")

statistics_for_replotting <- c("mm39_number_of_cells",
                            "mean_reads_per_cell",
                            "mm39_median_genes_per_cell")

#filenames <- list.files(sublib1_path,  full.names=TRUE)
#csv_filenames <- filenames[grepl("^.*/(.*.csv)", filenames)]
# Read in the agg_samp_ann_summary.csv file which aggregates all the .csv files together
agg_ana_sublib1_tidy <- readr::read_csv(paste0(sublib1_path, "agg_samp_ana_summary.csv")) %>%
  tidyr::pivot_longer(
  cols = starts_with(c("WT", "KO", "all")),
  names_to = c("sample"),
  values_to = "value") %>%
  dplyr::mutate(sublibrary = "subLib1")

agg_ana_sublib2_tidy <- readr::read_csv(paste0(sublib2_path, "agg_samp_ana_summary.csv")) %>%
  tidyr::pivot_longer(
  cols = starts_with(c("WT", "KO","all")),
  names_to = c("sample"),
  values_to = "value") %>%
  dplyr::mutate(sublibrary = "subLib2")

agg_ana_sublib_combined_facet <- rbind(agg_ana_sublib1_tidy,agg_ana_sublib2_tidy) %>%
  dplyr::filter(statistic %in% statistics_for_facet)

agg_ana_sublib_combined_fraction <- rbind(agg_ana_sublib1_tidy,agg_ana_sublib2_tidy) %>%
  dplyr::filter(statistic %in% statistics_fraction) %>%
  dplyr::select(-c('sample')) %>%
  dplyr::distinct()

agg_ana_sublib_combined_replot <- rbind(agg_ana_sublib1_tidy,agg_ana_sublib2_tidy) %>%
  dplyr::filter((statistic %in% statistics_for_replotting) &(sample != "all-well"))

### Read in pipelines.csv file
```

```
statistics_facet_plot<- ggplot(agg_ana_sublib_combined_facet, aes(x= sublibrary, y = value))+
  geom_boxplot(fill='gray',outlier.shape = NA)+
  geom_point(position=position_dodge(width=0.75), aes(color=sample), size = 4)+
  facet_wrap(~statistic, scales="free", ncol=4 ) +
  scale_color_manual(values=c("#540FBB", "#3c4e4b", "#00bfa0", "#76c8c8", "#98d1d1",
                              "#df979e", "#d7658b", "#c80064"))+
  theme_bw(base_family = "Roboto",
              base_size = 16)
#svg(paste0(res_path,"/alignment-stats-facet.svg"), width=12, height=5)
#png(paste0(res_path,"/alignment-stats-facet-full.png"), units = "in", res = 300, width = 15, height =
SaveFigure(res_path = res_path,
           plots = statistics_facet_plot,
           name = "statistics_facet_plot",
           width = 20,
           height = 12)
```

```
#dev.off()
```

```
statistics_fraction_plot <- ggplot(agg_ana_sublib_combined_fraction,
                                   aes(x=statistic, y=value))+
  geom_col(position = position_dodge2(preserve = "single"),
           aes(fill=sublibrary))+
    geom_text(aes(group = agg_ana_sublib_combined_fraction$sublibrary, y = agg_ana_sublib_combined_frac
  lims(y=c(0,1.2))+
  coord_flip()+
  theme_minimal(base_family = "Roboto",
                base_size = 16)
#svg(paste0(res_path,"alignment-fractions.svg"), width=12, height=8)
SaveFigure(res_path = res_path,
           plots = statistics_fraction_plot,
           name = "statistics_fraction_plot",
           width = 12,
           height = 6)
```

```
pipeline_stats_sublib1 <-  readr::read_csv(paste0(sublib1_path,"process/pipeline_stats.csv")) %>%
  dplyr::mutate(sublibrary="subLib1")

pipeline_stats_sublib2 <-  readr::read_csv(paste0(sublib2_path,"process/pipeline_stats.csv")) %>%
  dplyr::mutate(sublibrary="subLib2")

pipeline_stats_combined <- rbind(pipeline_stats_sublib1,pipeline_stats_sublib2)
pipeline_stats_combined
readAlignStats <- c("reads_align_input", "reads_align_unique", "reads_align_multimap",
                    "reads_too_many_loci", "reads_map_transcriptome")
pipeline_stats_readAlign <- pipeline_stats_combined %>%
  dplyr::filter(statistic %in%  readAlignStats)


align_input_total <- pipeline_stats_readAlign %>%
  dplyr::filter(statistic == "reads_align_input") %>%
  dplyr::select(value) %>%
  pull()

pipeline_stats_readAlign_spread <- pipeline_stats_readAlign %>%
  tidyr::spread(sublibrary, value)
pipeline_stats_readAlign_spread_matrix <- pipeline_stats_readAlign_spread %>%
  dplyr::select(-statistic)
pipeline_stats_readAlign_update <- sweep(as.matrix(pipeline_stats_readAlign_spread_matrix),2,align_inpu
  as.data.frame() %>%
  dplyr::mutate(statistic = pipeline_stats_readAlign_spread$statistic) %>%
  dplyr::filter(statistic != "reads_align_input") %>%
  tidyr::pivot_longer(cols = c("subLib1", "subLib2"), names_to = "sublibrary", values_to = "value")

pipeline_stats_readAlign_plot <- ggplot(pipeline_stats_readAlign_update, aes(x=statistic, y=value)) +
  geom_col(position = position_dodge2(preserve = "single"),
           aes(fill=sublibrary))+
  geom_text(aes(group = pipeline_stats_readAlign_update$sublibrary, y = pipeline_stats_readAlign_update$
  lims(y=c(0,1))+
  scale_x_discrete(limits = c("reads_too_many_loci","reads_align_multimap", "reads_map_transcriptome","
  coord_flip()+
  theme_minimal(base_family = "Roboto",
                base_size = 16)

SaveFigure(res_path = res_path,
           plots = pipeline_stats_readAlign_plot,
           name = "pipeline_stats_readAlign_plot",
           width = 12,
           height = 6)
```

```
top_row <- cowplot::plot_grid(statistics_fraction_plot, pipeline_stats_readAlign_plot, nrow = 1, labels
bottom_row <- cowplot::plot_grid(statistics_facet_plot,  nrow = 1, labels=c("C"), label_size = 18)

combined_stats <- cowplot::plot_grid(top_row, bottom_row, nrow=2)

SaveFigure(res_path = res_path,
           plots = combined_stats,
           name = "all_general_stats_combined",
           width = 20,
           height = 17)
```

```
knitr::include_graphics(paste0(res_path, "all_general_stats_combined.png"))
```

3 Alignment with the Parse Biosciences split-pipe program

Reads were aligned to the mouse genome (GRCm39) with the split-pipe (v0.9.6p) program from Parse Biosciences with the release 107 Ensembl GTF annotations. Sublibraries were first generated separately and then combined into a single dataset using `split-pipe --mode combine` command, passing in the `comb_unfilt_dge True` argument to generate the unfiltered count matrix.

Here, we reassess the quality of both sublibraries and verify the results of the alignment procedure by examining into the statistics derived from the `agg_ana_summary.csv` and `process/pipeline_info.csv` files.

4 QC assessments

4.1 Phred quality scores (Q30) are similar between sublibraries (~85-90% high quality bases)

From Figure 1A, we find that both sublibraries have very similar fraction of reads with Phred quality scores of 30 or above for barcodes 1-3 (bc1_Q30, bc2_Q30 and bc3_Q30) and polyN bases (polyN_Q30) on read2, as well as the bases from the transcript sequence on read1 (DNA_Q30). A phred score of 30 for a base means that the base call accuracy (i.e., the probability of a correct base call) is 99.9%, and therefore the greater the value of these fractions, the greater the fraction of high quality reads. From the data, we note that the range of values lies between around 0.85 - 0.90 for the reads from both sublibraries. These values for the barcode reads (bc1_Q30, bc2_Q30 and bc3_Q30) could explain why the `valid_barcode_fraction` for both sublibraries are around 60%, which implies that around 40% of the total reads from both sublibraries are discarded as the barcodes do not pass the quality filter.

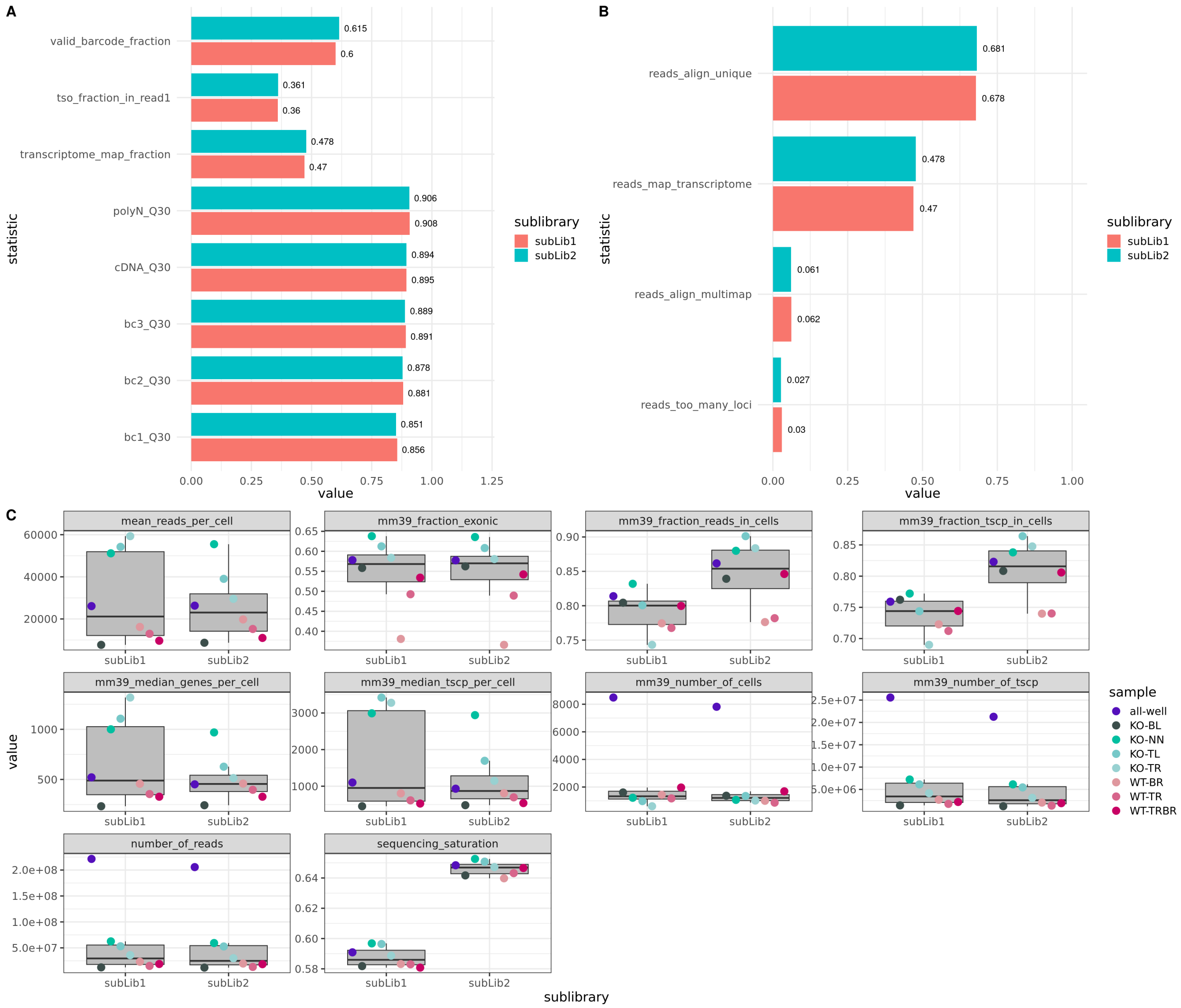

Figure 1: General QC statistics for both sublibraries

4.2 Alignment statistics are similar between sublibraries (~69% uniquely mapped reads)

Turning to the STAR read alignment statistics Figure 1B, we observe that the proportion of reads that align uniquely to the genome is around 67-68% for both sublibraries. This is slightly towards the lower end as we would expect at least 70% or more uniquely mapped reads against a well-studied reference genome. Fewer reads align to multiple regions of the genome, and even fewer reads are discarded as they map to too many loci. Thus, it appears that around 30% of the reads are unmapped for both sublibraries due to other unspecified reasons, presumably due to their lower base quality. About less than 50% of the reads map to the transcriptome for both sublibraries.

4.3 The KO samples, with the exception of KO_BL, have higher sequencing depth than the WT samples

With the exception of the KO_BL sample, the KO samples are sequenced to a greater depth (greater number of reads) than the WT samples (Figure 1C, mean_reads_per_cell panel). In addition, alongside the lower number of cells detected (`mm39_number_of_cells`), these samples have a greater median number of transcripts and genes detected per cell (`mm39_median_tscp_per_cell`, `mm39_median_genes_per_cell`) and number of unique transcripts detected per sample (`number_of_tscp`).

In general, for both sublibraries, the median genes identified per cell is quite low (usually, we expect around 1-2k genes identified per cell but here it is less than 1.5k). In addition, for the WT samples in particular, the mean reads per cell is also low (we expect at least 20-50k reads per cell on average, but it is less than this for the WT samples and KO_BL). Taken together, these features mean that we may not have enough power to properly cluster and separate cells as an insufficient number of transcripts are identified to distinguish between the cell populations. However, upon examining the sequencing saturation levels for the samples in both sublibraries, we find that they are around 58-65%, which is reasonably high. At this sequencing saturation level, we will need around 2.5 more reads on average for the discovery of an additional unique transcript. Despite the relatively high sequencing saturation levels, considering that the number of detected genes are low for all samples, it will be highly beneficial to re-sequence the samples (with emphasis on the WT samples) to greater depth if both time and cost permit.

5 Summary and next steps

- Phred QC scores appears lower than optimal, which may explain why around 40% of the barcode reads from both sublibraries were discarded. This may also explain the lower proportion of uniquely mapped reads (around 68%)- a feature which may warrant further investigation.
- There appears to be an imbalance in the sequencing depth for the KO vs WT samples, with the WT samples generally having lower median number of genes and transcripts detected, as well as lower mean number of reads and reads per cell. In addition, the median number of genes detected for all samples is less than 2000, which may result in difficulties in stable clustering and resolving the different cell types between the samples. It may therefore be useful to sequence these samples to greater depth if the downstream analysis does not show clear separation of cell types.
- In our analysis, we will not consider the all-well (combined) samples. Instead, we will reprocess each sample separately and only merge them prior to downstream analysis.