STAR_WASP_Bechmarking_Analysis

RA

1.0 Loading Required Libraries

```
library(splitstackshape)
library(dplyr)
library(ggplot2)
library(VennDiagram)
library(RColorBrewer)
library(gridExtra)
library(cowplot)
library(knitr)
library(kableExtra)
library(tidyverse)
library(Hmisc)
library(ggridges)
library(venn)
library(ggthemes)
library(stringr)
library(splitstackshape)
library(chron)
library(magicfor)
library(ggtext)
library(reshape2)
library(plyr)
library(formattable)
library(data.table)
library(ggrepel)
```

```
#Setting working directory
setwd("/home/asiimwe/projects/run_env/alpha_star_wasp_benchmarking/dataExtractions/")
```

2.0 Data Loading, Cleaning and Preprocessing

2.1 Loading and preprocessing benchmark data from all runs - data extracts from the system-time command

```
#run_results <- read.table(pipe ('ssh atlas "cat /home/asiimwe/projects/run_env/alpha_star_wasp_compari
run_results <- read.table("benchmark_results_all_runs.txt", sep="\t", header = FALSE, fill=TRUE, quote =

run_results <- as.data.frame(run_results[-c(1),])
run_results$V4 <- NULL
```

```
colnames(run_results) <- c("Sample", "Run", "Thread", "Param","Value")

run_results$Param <- as.character(run_results$Param)
run_results$Param[run_results$Param == "Elapsed (wall clock) time (h"] <- "Wall_Clock"
run_results$Param[run_results$Param == "Maximum resident set size (kbytes)"] <- "Memory"

run_results$Value <- as.character(run_results$Value)
run_results$Value <- gsub("mm:ss or m:ss): ", "", run_results$Value)
run_results$Value <- gsub(" ", "", run_results$Value)
#run_results$Value

unique(run_results$Run)
run_results$Run <- gsub("_Runs", "", run_results$Run)
unique(run_results$Run)

run_results_STAR <- run_results %>% filter(Run == "STAR")#STAR base - no variants
run_results_STAR_WASP <- run_results %>% filter(Run == "STAR_WASP") #STAR WASP
run_results_WASP <- run_results %>% filter(Run == "WASP") #WASP - change naming globally
run_results_STAR$Run <- NULL
run_results_STAR_WASP$Run <- NULL
run_results_WASP$Run <- NULL

run_results_STAR <- spread(run_results_STAR, key = Thread, value = Value)
colnames(run_results_STAR) <- c("Sample", "Param", "16 Threads-STAR", "32 Threads-STAR", "8 Threads-STA

run_results_STAR_WASP <- spread(run_results_STAR_WASP, key = Thread, value = Value)
colnames(run_results_STAR_WASP) <- c("Sample", "Param", "16 Threads-STAR_WASP", "32 Threads-STAR_WASP",

run_results_WASP <- spread(run_results_WASP, key = Thread, value = Value)
colnames(run_results_WASP) <- c("Sample", "Param", "16 Threads-WASP", "32 Threads-WASP", "8 Threads-WAS

data <- merge(run_results_STAR, run_results_STAR_WASP, by = c("Sample", "Param"))
data <- merge(data, run_results_WASP, by = c("Sample", "Param"))
#write.csv(data, file = "/home/asiimwe/projects/run_env/alpha_star_wasp_comparison/benchmark_results_al

#Plotting speeds and memory
data_melted <- melt(data, id.vars=c("Sample", "Param"))
data_melted <- cSplit(data_melted,"variable", "-", direction="wide")

colnames(data_melted) <- c("Sample", "Param", "Value", "Threads", "Run")
data_melted$Param <- as.character(data_melted$Param)

#data_melted_subset_clock <- data_melted_subset %>% filter(Param == "Wall_Clock")
#data_melted_subset_clock$Value <- gsub("\\.", ":", data_melted_subset_clock$Value)
#data_melted_subset_clock$Value
#data_melted_subset_clock$Value  <- chron(times. = data_melted_subset_clock$Value)#rejecting format - es


############-------------------------------------------------------------------------------
# #compressed dataset - removing parameters that are not needed
# #parameters to remove:
# param_remove <- c(
#   "Average resident set size (kbytes)",
```

```
#   "Average shared text size (kbytes)",
#   "Average stack size (kbytes)",
#   "Average total size (kbytes)",
#   "Average unshared data size (kbytes)",
#   "Exit status",
#   "Major (requiring I/O) page faults",
#   "Signals delivered",
#   "Socket messages received",
#   "Socket messages sent",
#   "Swaps"
# )
#
# dataset_subset <- data %>% filter(Param %nin% param_remove)
# write.csv(dataset_subset, file = "/home/asiimwe/projects/run_env/alpha_star_wasp_comparison/run_resul
###########-------------------------------------------------------------------------------

#Benchmark plots:
#Elapsed (wall clock) time (h    mm:ss or m:ss):       Average resident set size (kbytes)
```

2.2 Loading Log.final.out results for all runs

```
# Loading Log.final.out results to extract required parameters
final.log.results <- read.table("final_log_results_all_runs.txt", sep="|", header = FALSE, fill=TRUE, q
#final.log.results <- read.table(pipe('ssh atlas "cat /home/asiimwe/projects/run_env/alpha_star_wasp_co

colnames(final.log.results) <- unlist(final.log.results[c(1),])
final.log.results <- as.data.frame(final.log.results[-c(1),])
final.log.results <- as.data.frame(final.log.results)

final.log.results$Sample <- as.character(final.log.results$Sample)
final.log.results$Run <- as.character(final.log.results$Run)
final.log.results$Thread <- as.character(final.log.results$Thread)
final.log.results$Param <- as.character(final.log.results$Param)

colnames(final.log.results)
final.log.results$Param <- str_trim(final.log.results$Param)

final.log.results_filt <- final.log.results %>% filter(Value != "")

final.log.results_filt$Run <- as.character(final.log.results_filt$Run)
unique(final.log.results_filt$Run)
final.log.results_filt$Run[final.log.results_filt$Run == "STAR_Runs"] <- "STAR"
final.log.results_filt$Run[final.log.results_filt$Run == "STAR_WASP_Runs"] <- "STAR_WASP"
final.log.results_filt$Run[final.log.results_filt$Run == "WASP_Runs"] <- "WASP"
colnames(final.log.results_filt)[3] <- "Threads"

final.log.results_filt$Threads <- as.character(final.log.results_filt$Threads)
final.log.results_filt$Threads[final.log.results_filt$Threads == "16threads"] <- "16 Threads"
final.log.results_filt$Threads[final.log.results_filt$Threads == "32threads"] <- "32 Threads"
final.log.results_filt$Threads[final.log.results_filt$Threads == "8threads"] <- "8 Threads"

#Extracting number of reads
```

```
final.log.results_filt_input_reads <- final.log.results_filt %>% filter(Param == "Number of input reads
final.log.results_filt_mapping_speed <- final.log.results_filt %>% filter(Param == "Mapping speed, Mill
final.log.results_filt_average_input_read_length <- final.log.results_filt %>% filter(Param == "Average

unique(data_melted$Sample)
unique(data_melted$Run)
unique(data_melted$Threads)
unique(final.log.results_filt_input_reads$Sample)
unique(final.log.results_filt_input_reads$Run)
unique(final.log.results_filt_input_reads$Threads)

data_melted_join_pass1 <- inner_join(data_melted, final.log.results_filt_input_reads, by = c("Sample",
data_melted_join_pass2 <- inner_join(data_melted_join_pass1, final.log.results_filt_mapping_speed, by =
data_melted_beta <- inner_join(data_melted_join_pass2, final.log.results_filt_average_input_read_length

colnames(data_melted_beta)
colnames(data_melted_beta) <- c( "Sample", "Param", "Value", "Threads", "Run", "Param.y", "Number_of_inp
data_melted_beta$Param.y <- NULL
data_melted_beta$Param.x.x <- NULL
data_melted_beta$Param.y.y <- NULL

nrow(data_melted)
nrow(final.log.results_filt_input_reads)
nrow(final.log.results_filt_mapping_speed)
nrow(final.log.results_filt_average_input_read_length)
nrow(data_melted_beta)
#Removing samples not considered for our analyses
data_melted_beta <- data_melted_beta %>% filter(Sample != "HG00514" & Sample != "NA12878_Small" & Sample
#write.csv(data_melted_beta, file = "/home/asiimwe/projects/run_env/alpha_star_wasp_benchmarking/Downst
```

Prepairing data for Manuscript summary table

```
data_melted_beta <- read.csv("/home/asiimwe/projects/run_env/alpha_star_wasp_benchmarking/Downstream_Ana
data_melted_beta$X <- NULL
## Preparing data for Latex tables

# Installing required packages
# tinytex::parse_install(
#   text = "! LaTeX Error: File `ae.sty' not found."
# )
#
#
# tinytex::parse_install(
#   text = "! LaTeX Error: File `fullpage.sty' not found."
# )
# tinytex::parse_install(
#   text = "! LaTeX Error: File `moresize.sty' not found."
# )
#
#
# tinytex::parse_install(
```

```
#   text = "! LaTeX Error: File `colortbl.sty' not found."
# )
#
#
# tinytex::parse_install(
#   text = "! LaTeX Error: File `soul.sty' not found."
# )
#
# tinytex::parse_install(
#   text = "! LaTeX Error: File `adjustbox.sty' not found."
# )
#
#
# tinytex::parse_install(
#   text = "! LaTeX Error: File `collectbox.sty' not found."
# )
# tinytex::parse_install(
#  text = "! LaTeX Error: File `pdflscape.sty' not found."
#  )
#tinytex::parse_install(
#   text = "! LaTeX Error: File `grfext.sty' not found."
# )
# tinytex::parse_install(
#    text = "! LaTeX Error: File `tabu.sty' not found."
#  )
#
# tinytex::parse_install(
#    text = "! LaTeX Error: File `threeparttable.sty' not found."
#  )
#data_melted_beta <- read.csv("/home/asiimwe/projects/run_env/alpha_star_wasp_comparison/data_melted_be

unique(data_melted_beta$Run)
data_melted_beta$Threads <- ordered(data_melted_beta$Threads , levels = c("8 Threads", "16 Threads", "3
unique(data_melted_beta$Param)
data_melted_beta_subset <- data_melted_beta %>% filter(Param == "Wall_Clock" | Param == "Memory")
data_melted_beta_subset$Average_input_read_length <- NULL
#data_melted_beta_subset$threads_run <- paste0(data_melted_beta_subset$Threads, "_", data_melted_beta_s
#data_melted_beta_subset$Param <- NULL
#data_melted_beta_subset$Threads <- NULL
#data_melted_beta_subset$Run <- NULL

#----------------------STAR----------------------
data_melted_beta_subset_STAR <- data_melted_beta_subset %>% filter(Run == "STAR")
data_melted_beta_subset_STAR_Param <- data_melted_beta_subset_STAR[,c(1:4)] #check environment differen
data_melted_beta_subset_STAR_reads <- unique(data_melted_beta_subset_STAR[,c(1,6)])
data_melted_beta_subset_STAR_speed <- unique(data_melted_beta_subset_STAR[,c(1,4, 7)])
data_melted_beta_subset_STAR_Param$comb <- paste0(data_melted_beta_subset_STAR_Param$Param, "_", data_m
data_melted_beta_subset_STAR_Param$Param <- NULL
data_melted_beta_subset_STAR_Param$Threads <- NULL
data_melted_beta_subset_STAR_Param <- as.data.frame(data_melted_beta_subset_STAR_Param)
data_melted_beta_subset_STAR_Param_spread <- spread(data_melted_beta_subset_STAR_Param, key = comb, valu

data_melted_beta_subset_STAR_Param_spread <- data_melted_beta_subset_STAR_Param_spread %>% dplyr::group_
```

```
  dplyr::summarise_all(purrr::discard, is.na)

data_melted_beta_subset_STAR_Param_spread <- as.data.frame(data_melted_beta_subset_STAR_Param_spread)

data_melted_beta_subset_STAR_speed_spread <- spread(data_melted_beta_subset_STAR_speed, key = Threads, v
colnames(data_melted_beta_subset_STAR_speed_spread)
colnames(data_melted_beta_subset_STAR_speed_spread) <- c("Sample", "Mapping_Speed_8 Threads_STAR",  "Map
STAR_tab_subset <- inner_join(data_melted_beta_subset_STAR_reads, data_melted_beta_subset_STAR_Param_spi
STAR_tab_subset <- inner_join(STAR_tab_subset, data_melted_beta_subset_STAR_speed_spread, by = "Sample")
STAR_tab_subset <- STAR_tab_subset[,c(1,2,3,4,5,9,10,11, 6,7,8)]

#----------------------STAR_WASP-----------------------
data_melted_beta_subset_STAR_WASP <- data_melted_beta_subset %>% filter(Run == "STAR_WASP")
data_melted_beta_subset_STAR_WASP_Param <- data_melted_beta_subset_STAR_WASP[,c(1:4)]
data_melted_beta_subset_STAR_WASP_reads <- unique(data_melted_beta_subset_STAR_WASP[,c(1,6)])
data_melted_beta_subset_STAR_WASP_speed <- unique(data_melted_beta_subset_STAR_WASP[,c(1,4, 7)])
data_melted_beta_subset_STAR_WASP_Param$comb <- paste0(data_melted_beta_subset_STAR_WASP_Param$Param, "_
data_melted_beta_subset_STAR_WASP_Param$Param <- NULL
data_melted_beta_subset_STAR_WASP_Param$Threads <- NULL
data_melted_beta_subset_STAR_WASP_Param <- as.data.frame(data_melted_beta_subset_STAR_WASP_Param)
#rownames(data_melted_beta_subset_STAR_WASP_Param) <- 1:nrow(data_melted_beta_subset_STAR_WASP_Param)
#data_melted_beta_subset_STAR_WASP_Param$id <- 1:nrow(data_melted_beta_subset_STAR_WASP_Param)
data_melted_beta_subset_STAR_WASP_Param_spread <- spread(data_melted_beta_subset_STAR_WASP_Param, key =
data_melted_beta_subset_STAR_WASP_Param_spread$id <- NULL

data_melted_beta_subset_STAR_WASP_Param_spread <- data_melted_beta_subset_STAR_WASP_Param_spread %>% dpl
  dplyr::summarise_all(purrr::discard, is.na)

data_melted_beta_subset_STAR_WASP_Param_spread <- as.data.frame(data_melted_beta_subset_STAR_WASP_Param

data_melted_beta_subset_STAR_WASP_speed_spread <- spread(data_melted_beta_subset_STAR_WASP_speed, key =
colnames(data_melted_beta_subset_STAR_WASP_speed_spread)
colnames(data_melted_beta_subset_STAR_WASP_speed_spread) <- c("Sample", "Mapping_Speed_8 Threads_STAR_WA
STAR_WASP_tab_subset <- inner_join(data_melted_beta_subset_STAR_WASP_Param_spread, data_melted_beta_subs
STAR_WASP_tab_subset <- STAR_WASP_tab_subset[,c(1,2,3,4,8,9,10, 5,6,7)]


#------------------WASP--------------------------------
data_melted_beta_subset_WASP <- data_melted_beta_subset %>% filter(Run == "WASP")
data_melted_beta_subset_WASP_Param <- data_melted_beta_subset_WASP[,c(1:4)]
data_melted_beta_subset_WASP_reads <- unique(data_melted_beta_subset_WASP[,c(1,6)])
data_melted_beta_subset_WASP_speed <- unique(data_melted_beta_subset_WASP[,c(1,4, 7)])
data_melted_beta_subset_WASP_Param$comb <- paste0(data_melted_beta_subset_WASP_Param$Param, "_", data_me
data_melted_beta_subset_WASP_Param$Param <- NULL
data_melted_beta_subset_WASP_Param$Threads <- NULL
data_melted_beta_subset_WASP_Param <- as.data.frame(data_melted_beta_subset_WASP_Param)
#rownames(data_melted_beta_subset_WASP_Param) <- 1:nrow(data_melted_beta_subset_WASP_Param)
#data_melted_beta_subset_WASP_Param$id <- 1:nrow(data_melted_beta_subset_WASP_Param)
data_melted_beta_subset_WASP_Param_spread <- spread(data_melted_beta_subset_WASP_Param, key = comb, valu
data_melted_beta_subset_WASP_Param_spread$id <- NULL

data_melted_beta_subset_WASP_Param_spread <- data_melted_beta_subset_WASP_Param_spread %>% dplyr::group_
  dplyr::summarise_all(purrr::discard, is.na)
```

```
data_melted_beta_subset_WASP_Param_spread <- as.data.frame(data_melted_beta_subset_WASP_Param_spread)

data_melted_beta_subset_WASP_speed_spread <- spread(data_melted_beta_subset_WASP_speed, key = Threads, v
colnames(data_melted_beta_subset_WASP_speed_spread)
colnames(data_melted_beta_subset_WASP_speed_spread) <- c("Sample", "Mapping_Speed_8 Threads_WASP",  "Map

WASP_tab_subset <- inner_join(data_melted_beta_subset_WASP_Param_spread, data_melted_beta_subset_WASP_sp
WASP_tab_subset <- WASP_tab_subset[,c(1,2,3,4,8,9,10, 5,6,7)]

STAR_tab_subset <- as.data.frame(STAR_tab_subset)
STAR_WASP_tab_subset <- as.data.frame(STAR_WASP_tab_subset)
WASP_tab_subset <- as.data.frame(WASP_tab_subset)

n_way_merger0 <- merge(STAR_tab_subset, STAR_WASP_tab_subset, by = "Sample")
n_way_merger <- merge(n_way_merger0, WASP_tab_subset, by = "Sample")

#n_way_merger$Size_holder1 <- ""
#n_way_merger$Size_holder2 <- ""

colnames(n_way_merger)

n_way_merger <- n_way_merger[, c("Sample", "Number_of_input_reads",   #"Sample", "Size_holder1","Size_h
                                 "Memory_8 Threads_STAR","Memory_16 Threads_STAR", "Memory_32 Threads_S
                                 "Mapping_Speed_8 Threads_STAR","Mapping_Speed_16 Threads_STAR", "Mappi
                                 "Wall_Clock_8 Threads_STAR","Wall_Clock_16 Threads_STAR", "Wall_Clock_
                                 "Memory_8 Threads_STAR_WASP", "Memory_16 Threads_STAR_WASP", "Memory_3
                                 "Mapping_Speed_8 Threads_STAR_WASP", "Mapping_Speed_16 Threads_STAR_WA
                                 "Wall_Clock_8 Threads_STAR_WASP", "Wall_Clock_16 Threads_STAR_WASP", "
                                 "Memory_8 Threads_WASP", "Memory_16 Threads_WASP", "Memory_32 Threads_
                                 "Mapping_Speed_8 Threads_WASP", "Mapping_Speed_16 Threads_WASP", "Mapp
                                 "Wall_Clock_8 Threads_WASP", "Wall_Clock_16 Threads_WASP", "Wall_Clock
                                 )]

#knitr::kable(n_way_merger, "latex")
```

```
# Smoothing plot function
geom_xspline <- function(mapping = NULL, data = NULL, stat = "xspline",
                         position = "identity", show.legend = NA,
                         inherit.aes = TRUE, na.rm = TRUE,
                         spline_shape=-0.25, open=TRUE, rep_ends=TRUE, ...) {
  layer(
    geom = GeomXspline,
    mapping = mapping,
    data = data,
    stat = stat,
    position = position,
    show.legend = show.legend,
    inherit.aes = inherit.aes,
    params = list(spline_shape=spline_shape,
                  open=open,
                  rep_ends=rep_ends,
                  ...)
  )
```

```
}

GeomXspline <- ggproto("GeomXspline", GeomLine,
  required_aes = c("x", "y"),
  default_aes = aes(colour = "black", size = 0.5, linetype = 1, alpha = NA)
)

stat_xspline <- function(mapping = NULL, data = NULL, geom = "line",
                     position = "identity", show.legend = NA, inherit.aes = TRUE,
                     spline_shape=-0.25, open=TRUE, rep_ends=TRUE, ...) {
  layer(
    stat = StatXspline,
    data = data,
    mapping = mapping,
    geom = geom,
    position = position,
    show.legend = show.legend,
    inherit.aes = inherit.aes,
    params = list(spline_shape=spline_shape,
                  open=open,
                  rep_ends=rep_ends,
                  ...
    )
  )
}

StatXspline <- ggproto("StatXspline", Stat,

  required_aes = c("x", "y"),

  compute_group = function(self, data, scales, params,
                           spline_shape=-0.25, open=TRUE, rep_ends=TRUE) {
    tf <- tempfile(fileext=".png")
    png(tf)
    plot.new()
    tmp <- xspline(data$x, data$y, spline_shape, open, rep_ends, draw=FALSE, NA, NA)
    invisible(dev.off())
    unlink(tf)

    data.frame(x=tmp$x, y=tmp$y)
  }
)
```

3.0 Data Visualization

3.1 Memory

```
data_melted_beta <- read.csv("/home/asiimwe/projects/run_env/alpha_star_wasp_benchmarking/Downstream_Ana
#data_melted_beta <- read.csv(pipe('ssh atlas "cat /home/asiimwe/projects/run_env/alpha_star_wasp_compa

data_melted_beta$X <- NULL
colnames(data_melted_beta)
```

```
## [1] "Sample"
## [2] "Param"
## [3] "Value"
## [4] "Threads"
## [5] "Run"
## [6] "Number_of_input_reads"
## [7] "Mapping_speed_Million_of_reads_per_hour"
## [8] "Average_input_read_length"
```

```
#"Sample"  "Param"  "Value" "Threads" "Run" "Mapping_speed_Million_of_reads_per_hour" "Average_input_re

## Cleaning up Sample names
unique(data_melted_beta$Sample)
```

```
##  [1] HG00512                     HG00513
##  [3] HG00731                     HG00732
##  [5] HG00733                     NA12878_Nucleus_nonPolyA
##  [7] NA12878_Nucleus_nonPolyA_Rep NA12878_Nucleus_PolyA
##  [9] NA12878_Nucleus_PolyA_Rep    NA12878_PolyA
## [11] NA12878_PolyA_Rep            NA12878_Total
## [13] NA12878_Total_Rep            NA19238
## [15] NA19239                      NA19240
## 16 Levels: HG00512 HG00513 HG00731 HG00732 HG00733 ... NA19240
```

```
#Removing NA12878 from all NA12878 derived samples
data_melted_beta$Sample <- gsub("NA12878_", "", data_melted_beta$Sample)
data_melted_beta$Sample <- gsub("Rep", "R2", data_melted_beta$Sample)
data_melted_beta$Sample[data_melted_beta$Sample == "Total"] <- "Total_R1"
data_melted_beta$Sample[data_melted_beta$Sample == "PolyA"] <- "PolyA_R1"
data_melted_beta$Sample[data_melted_beta$Sample == "Nucleus_PolyA"] <- "Nucleus_PolyA_R1"
data_melted_beta$Sample[data_melted_beta$Sample == "Nucleus_nonPolyA"] <- "Nucleus_nonPolyA_R1"

#Filtering samples not needed downstream:
unique(data_melted_beta$Sample)
```

```
##  [1] "HG00512"             "HG00513"             "HG00731"
##  [4] "HG00732"             "HG00733"             "Nucleus_nonPolyA_R1"
##  [7] "Nucleus_nonPolyA_R2" "Nucleus_PolyA_R1"    "Nucleus_PolyA_R2"
## [10] "PolyA_R1"            "PolyA_R2"            "Total_R1"
## [13] "Total_R2"            "NA19238"             "NA19239"
## [16] "NA19240"
```

```
data_melted_beta$Threads <- as.character(data_melted_beta$Threads )
data_melted_beta$Threads[data_melted_beta$Threads == "8threads" ] <- "8 Threads"
data_melted_beta$Threads[data_melted_beta$Threads == "16threads" ] <- "16 Threads"
data_melted_beta$Threads[data_melted_beta$Threads == "32threads" ] <- "32 Threads"
unique(data_melted_beta$Threads)
```

```
## [1] "16 Threads" "32 Threads" "8 Threads"
```

```r
#Data cleanup
str(data_melted_beta)
```

```
## 'data.frame':    3168 obs. of  8 variables:
##  $ Sample                                 : chr  "HG00512" "HG00512" "HG00512" "HG00512" ...
##  $ Param                                  : Factor w/ 22 levels "Average resident set size (kbytes)"
##  $ Value                                  : Factor w/ 1387 levels "0","1","1:01:58",..: 1 1 1 1 1 1 1
##  $ Threads                                : chr  "16 Threads" "16 Threads" "16 Threads" "16 Threads"
##  $ Run                                    : Factor w/ 3 levels "STAR","STAR_WASP",..: 1 1 1 1 1 1 1
##  $ Number_of_input_reads                  : int  66055710 66055710 66055710 66055710 66055710 660557
##  $ Mapping_speed_Million_of_reads_per_hour: num  1226 1226 1226 1226 1226 ...
##  $ Average_input_read_length              : int  102 102 102 102 102 102 102 102 102 102 ...
```

```r
data_melted_beta$Sample <- as.character(data_melted_beta$Sample)
data_melted_beta$Param <- as.character(data_melted_beta$Param)
data_melted_beta$Value <- gsub("%", "", data_melted_beta$Value)

#Reordering threads to appear in the correct order (8, 16, 32)
data_melted_beta$Threads <- as.factor(data_melted_beta$Threads)

data_melted_beta$Threads <- ordered(data_melted_beta$Threads , levels = c("8 Threads", "16 Threads", "32

data_melted_beta_memory <- as.data.frame(data_melted_beta %>% filter(Param == "Memory"))
unique(data_melted_beta_memory$Param)
```

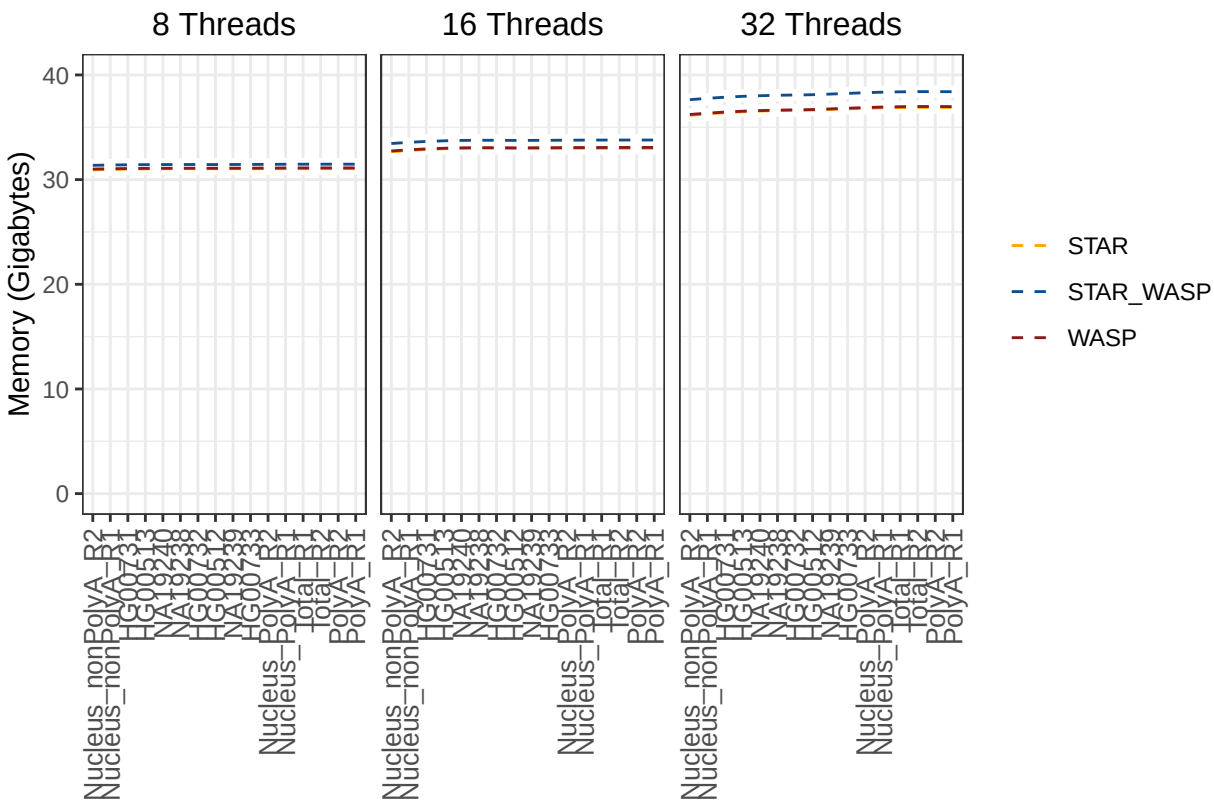
```
## [1] "Memory"
```

```r
data_melted_beta_memory$Threads <- ordered(data_melted_beta_memory$Threads , levels = c("8 Threads", "16

data_melted_beta_memory$Value <- as.numeric(data_melted_beta_memory$Value)
data_melted_beta_memory$Sample <- as.factor(data_melted_beta_memory$Sample)
data_melted_beta_memory$Run <- as.factor(data_melted_beta_memory$Run)

data_melted_beta_memory <- data_melted_beta_memory[order(data_melted_beta_memory$Sample, data_melted_be

#range(data_melted_beta_memory$Value) #30649908 38445368

data_melted_beta_memory <- data_melted_beta_memory %>%
  group_by(Sample, Threads) %>%
  dplyr::mutate(mem_order = sum(Value)) %>%
  arrange(mem_order)
#class(data_melted_beta_memory) - "grouped_df" "tbl_df"     "tbl"        "data.frame"
data_melted_beta_memory <- as.data.frame(data_melted_beta_memory)


#Memory
# data_melted_beta_memory %>% filter(Sample != "NA12878_Small") %>%
#   #mutate(Sample = fct_reorder2(Sample, Threads, mem_order,  .desc = FALSE)) %>%
#   ggplot(aes(x = reorder(Sample, (Value)/1000000), y = (Value)/1000000, group=Run)) + #Dividing by 10
#   geom_line(aes(color = Run, linetype = Run)) +
#   scale_color_manual(values = c("black", "gray10", "gray50")) +
#   facet_wrap(~Threads) +
```

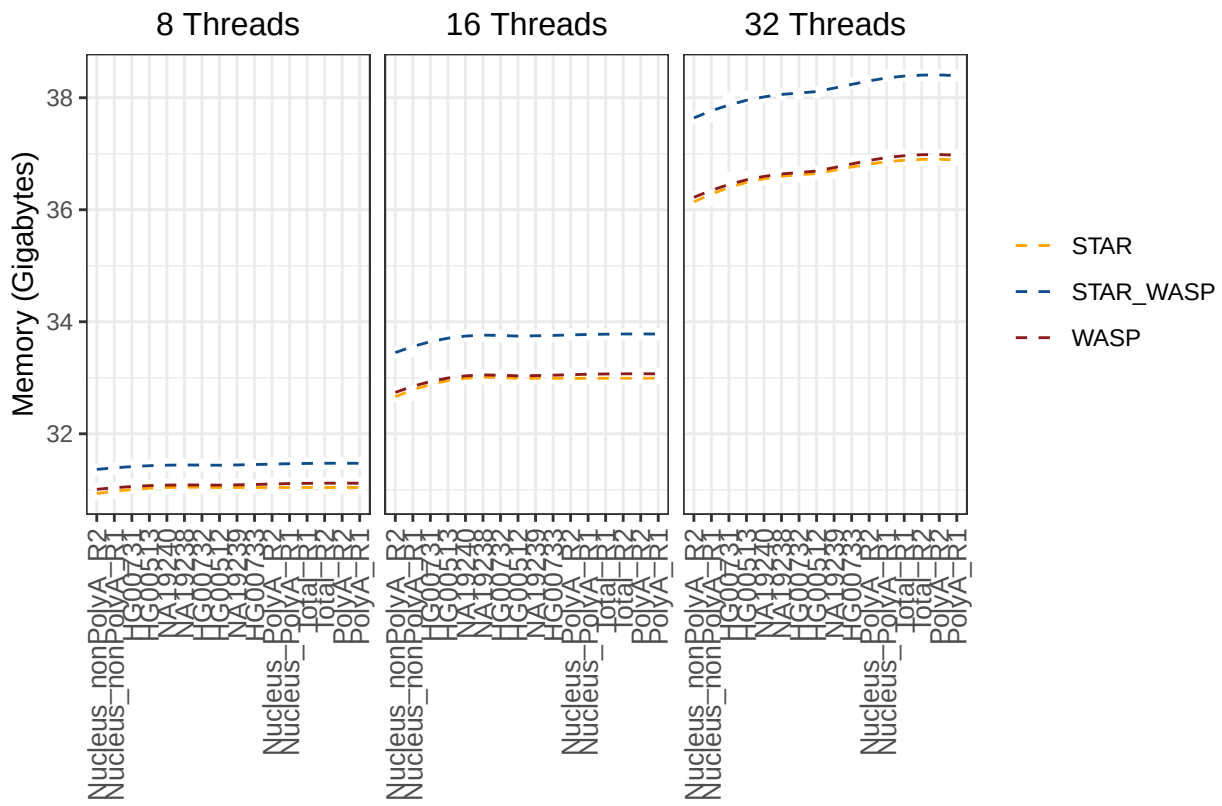
```
#   labs(y = "Memory (Gigabytes)", x="") + scale_y_continuous() +
#   theme_bw() + theme(legend.title = element_blank()) + #Base, bw, excel_new(), few, light, lindraw
#   theme(strip.background =element_rect(fill="white", colour = "white"))+
#   theme(strip.text = element_text(colour = 'black'), strip.text.x = element_markdown(hjust = 0)) +
#   #theme(strip.text.x = element_blank()) +
#   theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10)) #+
#   #geom_text(x = 6, y = max(data_melted_beta_memory$Value), aes(label = label),
#     #    data = dat_text, check_overlap = TRUE, inherit.aes = FALSE)

global_colors <-  c("orange", "dodgerblue4", "firebrick4")

#Overal we shall the same sample order across the board based on Number of Input Reads and Sample pairs
unique(data_melted_beta %>% filter(Run == "STAR") %>% select("Sample", "Number_of_input_reads"))
```

```
##                  Sample Number_of_input_reads
## 1               HG00512              66055710
## 23              HG00513              58601893
## 45              HG00731              56254714
## 67              HG00732              70029452
## 89              HG00733              92075712
## 111 Nucleus_nonPolyA_R1             110469791
## 133 Nucleus_nonPolyA_R2             106919251
## 155    Nucleus_PolyA_R1             128402941
## 177    Nucleus_PolyA_R2             116517502
## 199            PolyA_R1              97548052
## 221            PolyA_R2              93555584
## 243            Total_R1             105089150
## 265            Total_R2              92494632
## 287             NA19238              63949386
## 309             NA19239              72457249
## 331             NA19240              59219085
```

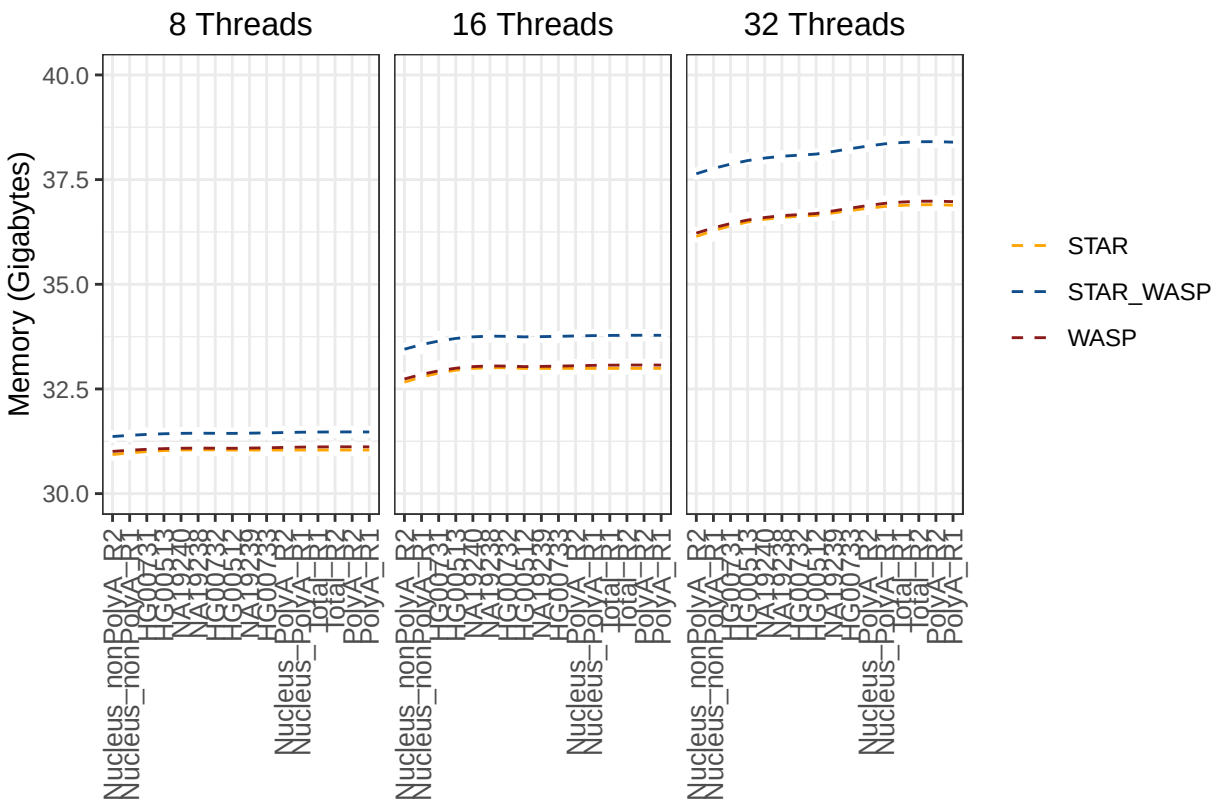
```
order_dfs <- c(
  "Nucleus_PolyA_R1", "Nucleus_PolyA_R2",
  "Nucleus_nonPolyA_R1", "Nucleus_nonPolyA_R2",
  "Total_R1", "Total_R2",
  "PolyA_R1", "PolyA_R2",
  "HG00733", "NA19239", "HG00732", "HG00512","NA19238","NA19240","HG00513", "HG00731")

data_melted_beta_memory %>%
 # mutate(Sample = fct_reorder2(Sample, Threads, mem_order,  .desc = FALSE)) %>%
  ggplot(aes(x = reorder(Sample, (Value)/1000000), y = (Value)/1000000, group=Run,color=factor(Run))) +
  geom_point(color="white") +
  geom_smooth(se=FALSE, linetype="dashed", size=0.5) +
  #geom_xspline(size=0.5)
  scale_color_manual(values = global_colors) +
  facet_wrap(~Threads) +
  labs(y = "Memory (Gigabytes)", x="") + scale_y_continuous(limits = c(0,40)) + #rescaling to range fro
  theme_bw() + theme(legend.title = element_blank()) +
  theme(strip.background =element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x = element_markdown(hjust = 0.5, size=
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))
```




```
# Without re-scaling
data_melted_beta_memory %>%
 # mutate(Sample = fct_reorder2(Sample, Threads, mem_order,  .desc = FALSE)) %>%
  ggplot(aes(x = reorder(Sample, (Value)/1000000), y = (Value)/1000000, group=Run,color=factor(Run))) +
  geom_point(color="white") +
  geom_smooth(se=FALSE, linetype="dashed", size=0.5) +
  #geom_xspline(size=0.5)
  scale_color_manual(values = global_colors) +
  facet_wrap(~Threads) +
  labs(y = "Memory (Gigabytes)", x="") + scale_y_continuous() + #rescaling to range from 0-40 GB
  theme_bw() + theme(legend.title = element_blank()) +
  theme(strip.background =element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x = element_markdown(hjust = 0.5, size=
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))
```

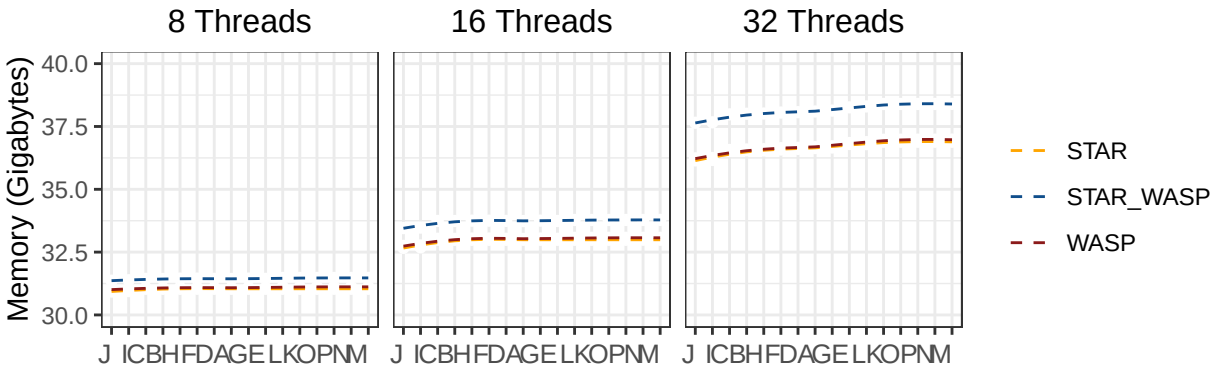



```
# 30-40 scale
data_melted_beta_memory %>%
 # mutate(Sample = fct_reorder2(Sample, Threads, mem_order,  .desc = FALSE)) %>%
  ggplot(aes(x = reorder(Sample, (Value)/1000000), y = (Value)/1000000, group=Run,color=factor(Run))) +
  geom_point(color="white") +
  geom_smooth(se=FALSE, linetype="dashed", size=0.5) +
  #geom_xspline(size=0.5)
  scale_color_manual(values =global_colors) +
  facet_wrap(~Threads) +
  labs(y = "Memory (Gigabytes)", x="") + scale_y_continuous(limits = c(30,40)) + #rescaling to range fr
  theme_bw() + theme(legend.title = element_blank()) +
  theme(strip.background =element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x = element_markdown(hjust = 0.5, size=
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))
```




```
# Applying shorter sample labels:
leg.txt <- levels(data_melted_beta_memory$Sample)
x.labels <- structure(LETTERS[seq_along(leg.txt)],
                      .Names = leg.txt) #Our sample names are too long - using letters that will be mapp

p1 <- data_melted_beta_memory %>%
 # mutate(Sample = fct_reorder2(Sample, Threads, mem_order,  .desc = FALSE)) %>%
  ggplot(aes(x = reorder(Sample, (Value)/1000000), y = (Value)/1000000, group=Run,color=factor(Run))) +
  geom_point(color="white") +
  geom_smooth(se=FALSE, linetype="dashed", size=0.5) +
  #geom_xspline(size=0.5)
  scale_color_manual(values =global_colors) +
  facet_wrap(~Threads) +
  labs(y = "Memory (Gigabytes)", x="") + scale_y_continuous(limits = c(30,40)) + #rescaling to range fr
  theme_bw() + theme(legend.title = element_blank()) +
  theme(strip.background =element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x = element_markdown(hjust = 0.5, size=
  theme(axis.text.x = element_text(angle = 0,hjust = 1,vjust = 0.5, size=10)) +scale_x_discrete(name = "

legend <-  get_legend(data_melted_beta_memory %>%
    mutate(Sample = fct_reorder2(Sample, Threads, mem_order,  .desc = FALSE)) %>%
  ggplot(aes(x = reorder(Sample, (Value)/1000000), y = (Value)/1000000, group=Run,color=factor(Run))) +
     geom_line() +
     scale_color_manual(values = global_colors) +
     facet_wrap(~Threads) +
    labs(y = "Memory (Gigabytes)", x="") + scale_y_continuous(limits = c(30,40)) +
```

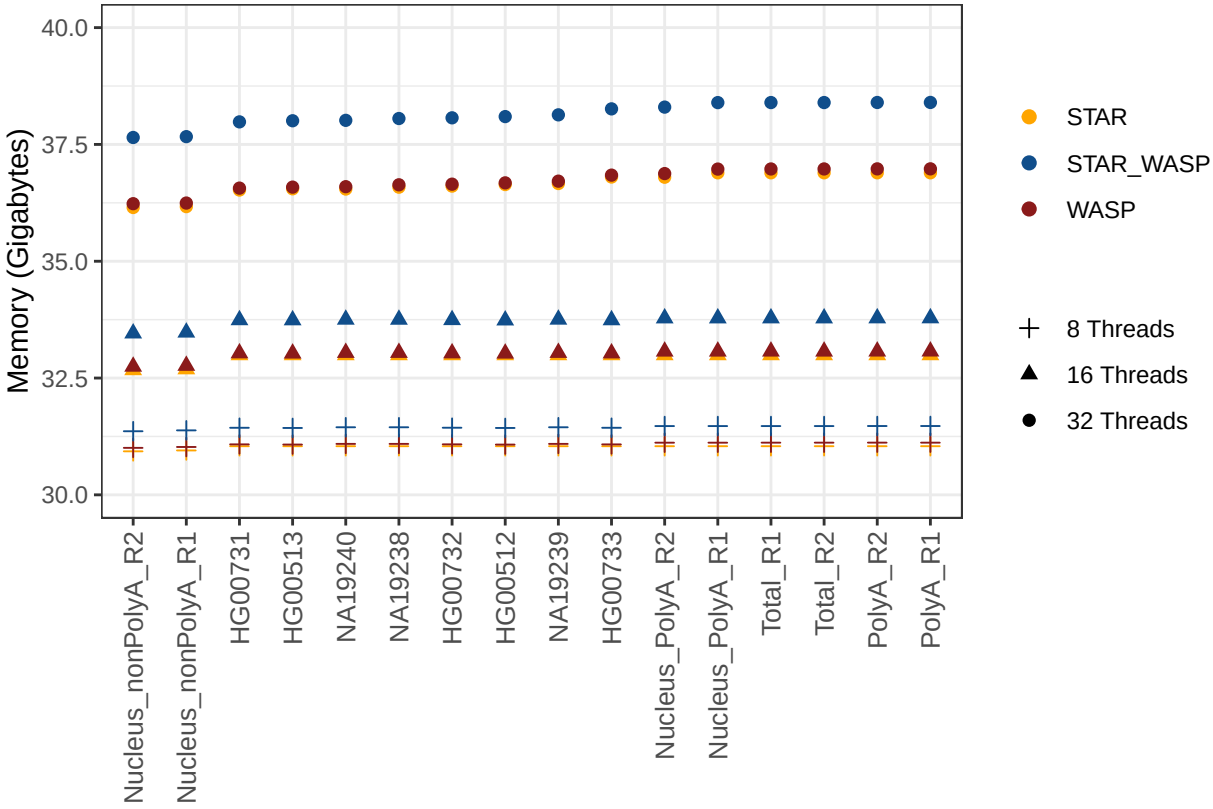
```
    theme_bw() + theme(legend.title = element_blank(), legend.position = "bottom") + #Base, bw, excel_
    theme(strip.background =element_rect(fill="white", colour = "white"))+
    theme(strip.text = element_text(colour = 'black'), strip.text.x = element_markdown(hjust = 0)) +
        geom_point(aes(shape = Sample), alpha = 0) +
  scale_shape_manual(name = "Samples", values = x.labels) +
    guides(shape = guide_legend(override.aes = list(size = 2.5, alpha = 1))) +
    scale_x_discrete(name = "Samples", labels = x.labels))

grid.arrange(p1, legend, nrow=2, ncol=1)
```



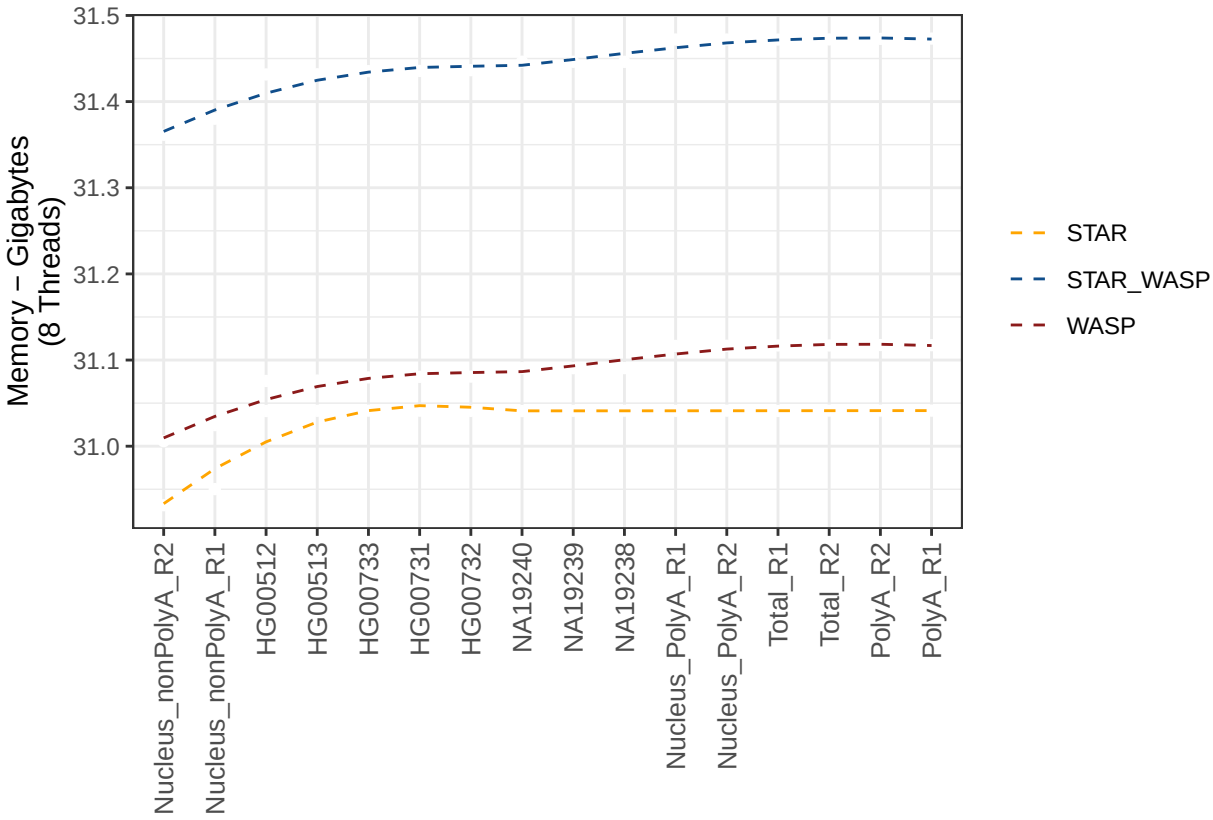
HG00512	E	HG00733	I	Nucleus_nonPolyA_R1	M	PolyA_R1	STAR STAR_WASP
HG00513	F	NA19238	J	Nucleus_nonPolyA_R2	N	PolyA_R2	
HG00731	G	NA19239	K	Nucleus_PolyA_R1	O	Total_R1	
HG00732	H	NA19240	L	Nucleus_PolyA_R2	P	Total_R2	

```
# Representation 2
data_melted_beta_memory %>%
  #mutate(Sample = fct_reorder2(Sample, Threads, mem_order,  .desc = FALSE)) %>%
    ggplot(aes(x = reorder(Sample, (Value)/1000000), y = (Value)/1000000, group=Run)) +
  geom_point(aes(color=factor(Run),shape=factor(Threads),fill=factor(Threads)), size=2)+
  #geom_line()+
  scale_shape_manual(values=c(3, 17, 16))+
  #scale_color_manual(values=c('#999999','#E69F00', '#56B4E9'))+
  scale_color_manual(values = global_colors) +
 # facet_wrap(~Threads) +
  #labs(y = "Mapping Speed (Million of reads/hour", x="") +
  labs(y = "Memory (Gigabytes)", x="")+ scale_y_continuous(limits = c(30,40)) +
  theme_bw() + theme(legend.title = element_blank()) + #Base, bw, excel_new(), few, light, lindraw
  theme(strip.background =element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x = element_markdown(hjust = 0)) +
```

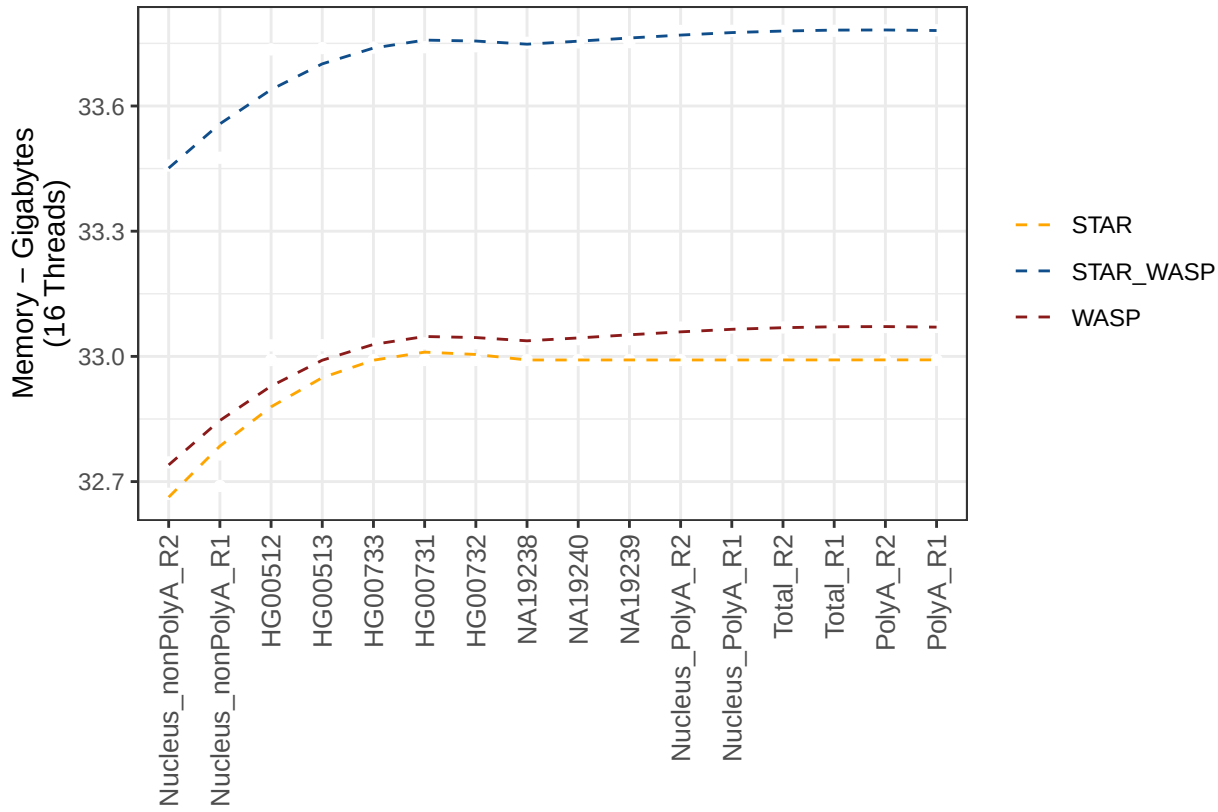
```
  #theme(strip.text.x = element_blank()) +
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10)) #+
```



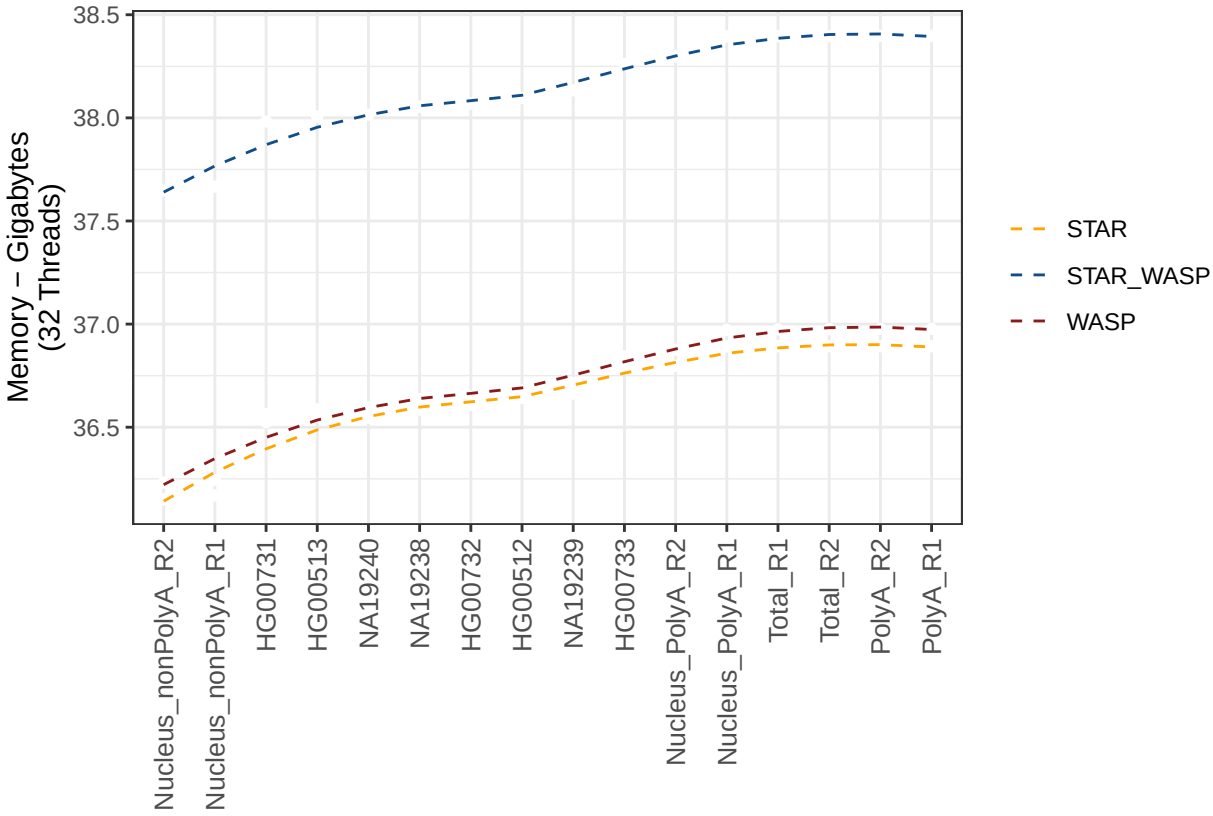

```
  #geom_text(x = 6, y = max(data_melted_beta_memory$Value), aes(label = label),
    #    data = dat_text, check_overlap = TRUE, inherit.aes = FALSE)

#Plot 1 - Splitting up plots by thread
# 8 Threads
data_melted_beta_memory %>% filter(Threads == "8 Threads") %>%
  #mutate(Sample = fct_reorder2(Sample, Threads, mem_order,  .desc = FALSE)) %>%
    ggplot(aes(x = reorder(Sample, (Value)/1000000), y = (Value)/1000000, group=Run,color=factor(Run)))
  #geom_line(aes(color = Run, linetype = Run)) + #in
    geom_point(color="white") +   geom_smooth(se=FALSE, linetype="dashed", size=0.5) +
  #scale_color_manual(values = c("black", "gray40", "gray80")) +
    scale_color_manual(values = global_colors) +
  #facet_wrap(~Threads) +
  labs(y = paste0("Memory - Gigabytes", "\n", "(8 Threads)"), x="") + scale_y_continuous(limits = c()) +
  theme_bw() + theme(legend.title = element_blank()) +
  theme(strip.background =element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x = element_markdown(hjust = 0)) +
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))
```



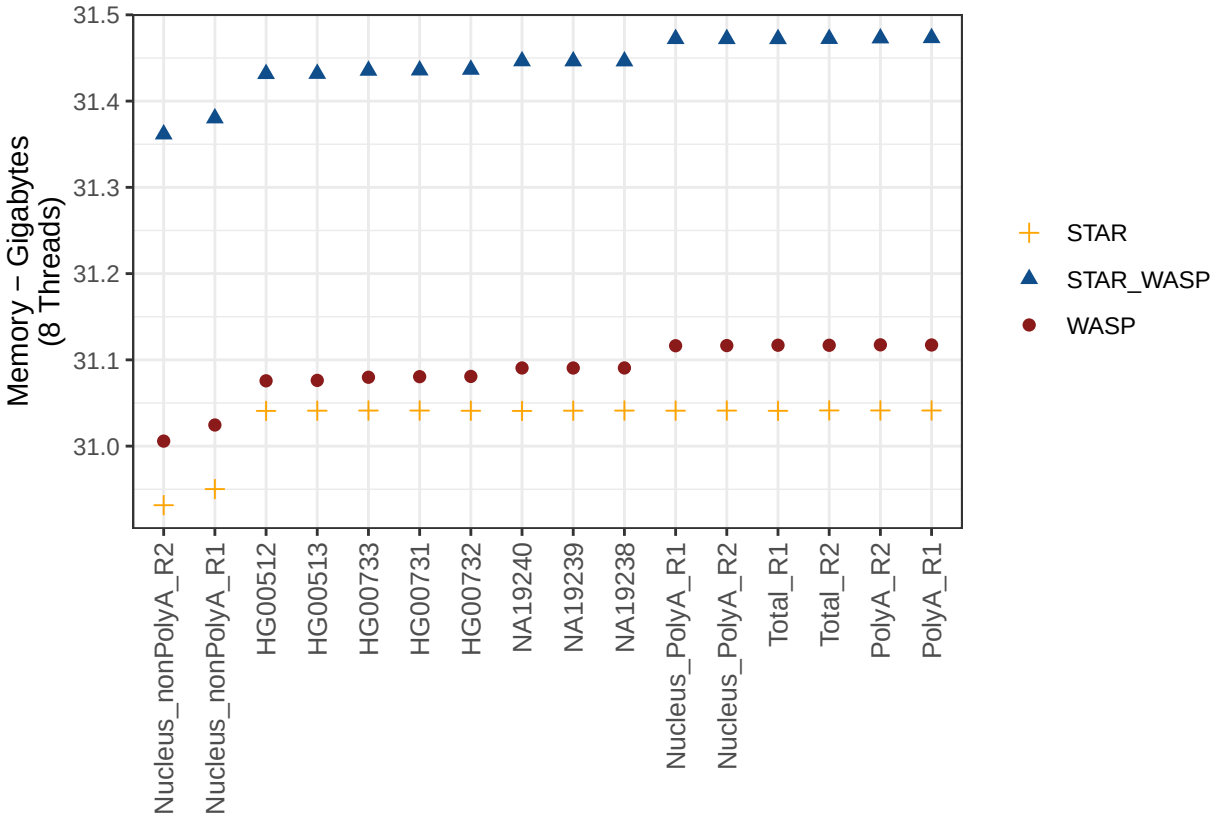

```
# 16 Threads
data_melted_beta_memory %>% filter(Threads == "16 Threads") %>%
  #mutate(Sample = fct_reorder2(Sample, Threads, mem_order,  .desc = FALSE)) %>%
    ggplot(aes(x = reorder(Sample, (Value)/1000000), y = (Value)/1000000, group=Run,color=factor(Run)))
  #geom_line(aes(color = Run, linetype = Run)) +
  geom_point(color="white") +   geom_smooth(se=FALSE, linetype="dashed", size=0.5) +
    scale_color_manual(values =global_colors) +
  #facet_wrap(~Threads) +
  labs(y = paste0("Memory - Gigabytes", "\n", "(16 Threads)"), x="") + scale_y_continuous(limits = c())
  theme_bw() + theme(legend.title = element_blank()) +
  theme(strip.background =element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x = element_markdown(hjust = 0)) +
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))
```



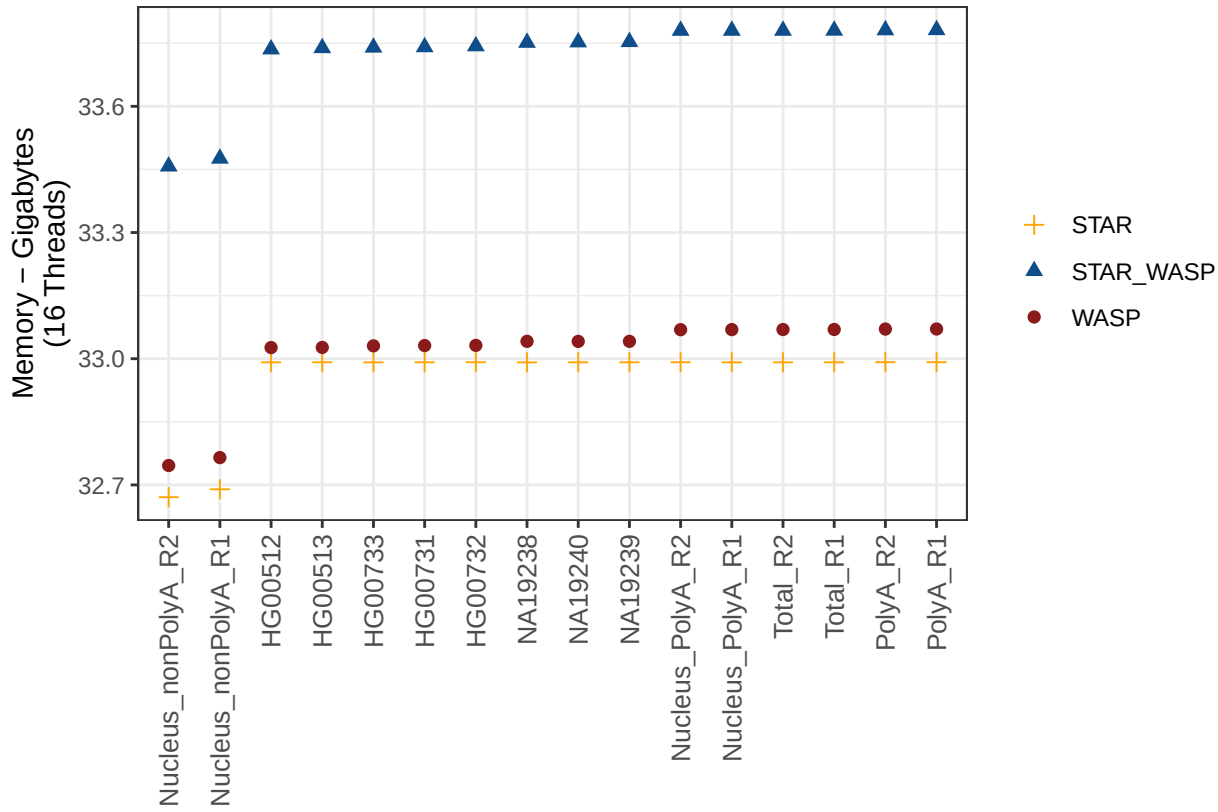

```
# 32 threads
data_melted_beta_memory %>% filter(Threads == "32 Threads") %>%
  #mutate(Sample = fct_reorder2(Sample, Threads, mem_order,  .desc = FALSE)) %>%
    ggplot(aes(x = reorder(Sample, (Value)/1000000), y = (Value)/1000000, group=Run,color=factor(Run)))
  #geom_line(aes(color = Run, linetype = Run)) +
  geom_point(color="white") +   geom_smooth(se=FALSE, linetype="dashed", size=0.5) +
    scale_color_manual(values =global_colors) +
  labs(y = paste0("Memory - Gigabytes", "\n", "(32 Threads)"), x="") + scale_y_continuous(limits = c())
  theme_bw() + theme(legend.title = element_blank()) +
  theme(strip.background =element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x = element_markdown(hjust = 0)) +
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))
```



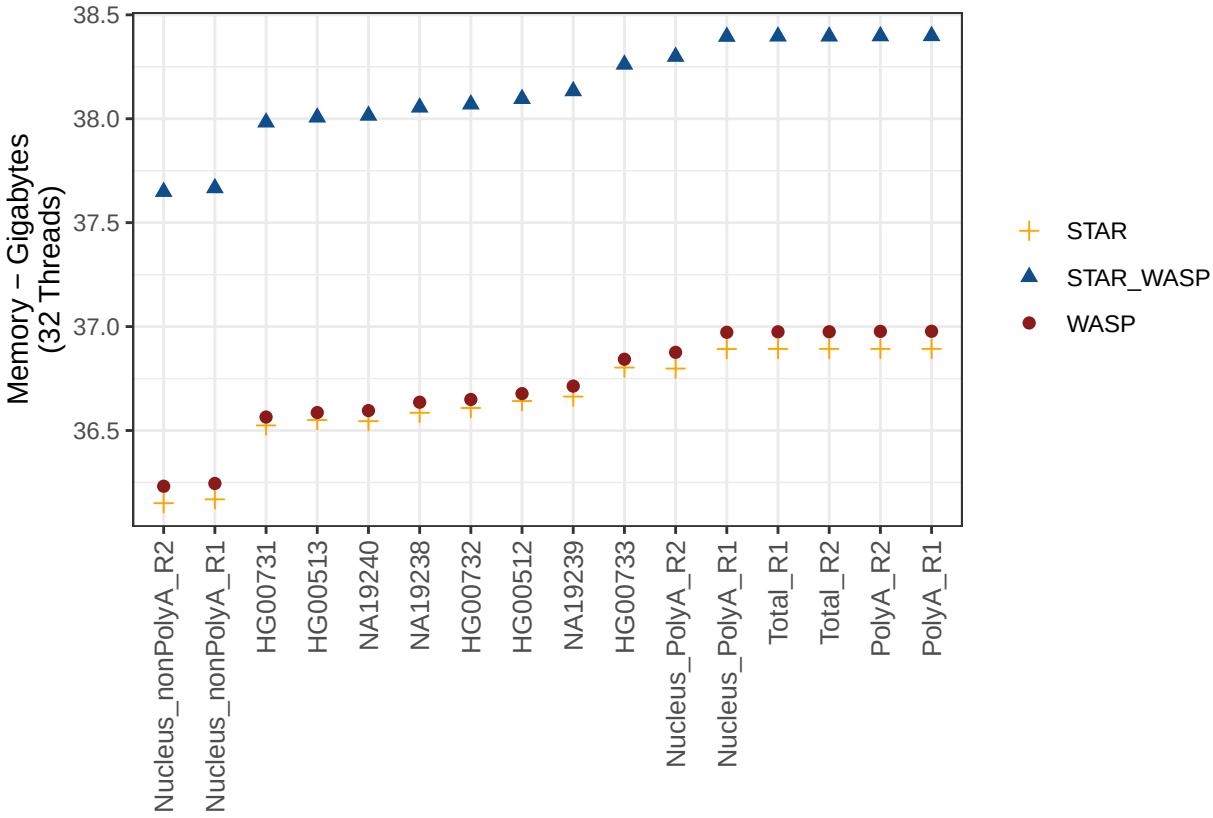

```
#Plot 2
# 8 Threads
data_melted_beta_memory %>% filter(Threads == "8 Threads") %>%
 # mutate(Sample = fct_reorder2(Sample, Threads, mem_order,  .desc = FALSE)) %>%
    ggplot(aes(x = reorder(Sample, (Value)/1000000), y = (Value)/1000000, group=Run,color=factor(Run)))
  geom_point(aes(color=factor(Run),shape=factor(Run),fill=factor(Run)), size=2)+
  scale_shape_manual(values=c(3, 17, 16))+
  scale_color_manual(values=global_colors)+
  labs(y = paste0("Memory - Gigabytes", "\n", "(8 Threads)"), x="") + scale_y_continuous(limits = c()) +
  theme_bw() + theme(legend.title = element_blank()) +
  theme(strip.background =element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x = element_markdown(hjust = 0)) +
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))
```




```
# 16 Threads
data_melted_beta_memory %>% filter(Threads == "16 Threads") %>%
 # mutate(Sample = fct_reorder2(Sample, Threads, mem_order,  .desc = FALSE)) %>%
    ggplot(aes(x = reorder(Sample, (Value)/1000000), y = (Value)/1000000, group=Run,color=factor(Run)))
  geom_point(aes(color=factor(Run),shape=factor(Run),fill=factor(Run)), size=2)+
  scale_shape_manual(values=c(3, 17, 16))+
  scale_color_manual(values=global_colors)+
  labs(y = paste0("Memory - Gigabytes", "\n", "(16 Threads)"), x="") + scale_y_continuous(limits = c())
  theme_bw() + theme(legend.title = element_blank()) +
  theme(strip.background =element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x = element_markdown(hjust = 0)) +
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))
```




```
# 32 threads
data_melted_beta_memory %>% filter(Threads == "32 Threads") %>%
 # mutate(Sample = fct_reorder2(Sample, Threads, mem_order,  .desc = FALSE)) %>%
    ggplot(aes(x = reorder(Sample, (Value)/1000000), y = (Value)/1000000, group=Run,color=factor(Run)))
  geom_point(aes(color=factor(Run),shape=factor(Run),fill=factor(Run)), size=2)+
  scale_shape_manual(values=c(3, 17, 16))+
  scale_color_manual(values=global_colors)+
  labs(y = paste0("Memory - Gigabytes", "\n", "(32 Threads)"), x="") + scale_y_continuous(limits = c())
  theme_bw() + theme(legend.title = element_blank()) +
  theme(strip.background =element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x = element_markdown(hjust = 0)) +
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))
```




```
#extracting max memory used per run and thread for qsub mem allocation guage
data_melted_beta_memory %>% filter(Run == "STAR" & Threads == "8 Threads") %>% select(Value) %>% max() 
```

```
## [1] 31041356
```

```
data_melted_beta_memory %>% filter(Run == "STAR" & Threads == "16 Threads") %>% select(Value) %>% max()
```

```
## [1] 32991760
```

```
data_melted_beta_memory %>% filter(Run == "STAR" & Threads == "32 Threads") %>% select(Value) %>% max()
```

```
## [1] 36892688
```

```
data_melted_beta_memory %>% filter(Run == "WASP" & Threads == "8 Threads") %>% select(Value) %>% max() 
```

```
## [1] 31117428
```

```
data_melted_beta_memory %>% filter(Run == "WASP" & Threads == "16 Threads") %>% select(Value) %>% max()
```

```
## [1] 33070648
```

```
data_melted_beta_memory %>% filter(Run == "WASP" & Threads == "32 Threads") %>% select(Value) %>% max()
```

```
## [1] 36977224
```

```
data_melted_beta_memory %>% filter(Run == "STAR_WASP" & Threads == "8 Threads") %>% select(Value) %>% ma
```

```
## [1] 31473172
```

```
data_melted_beta_memory %>% filter(Run == "STAR_WASP" & Threads == "16 Threads") %>% select(Value) %>% m
```

```
## [1] 33781520
```

```
data_melted_beta_memory %>% filter(Run == "STAR_WASP" & Threads == "32 Threads") %>% select(Value) %>% m
```

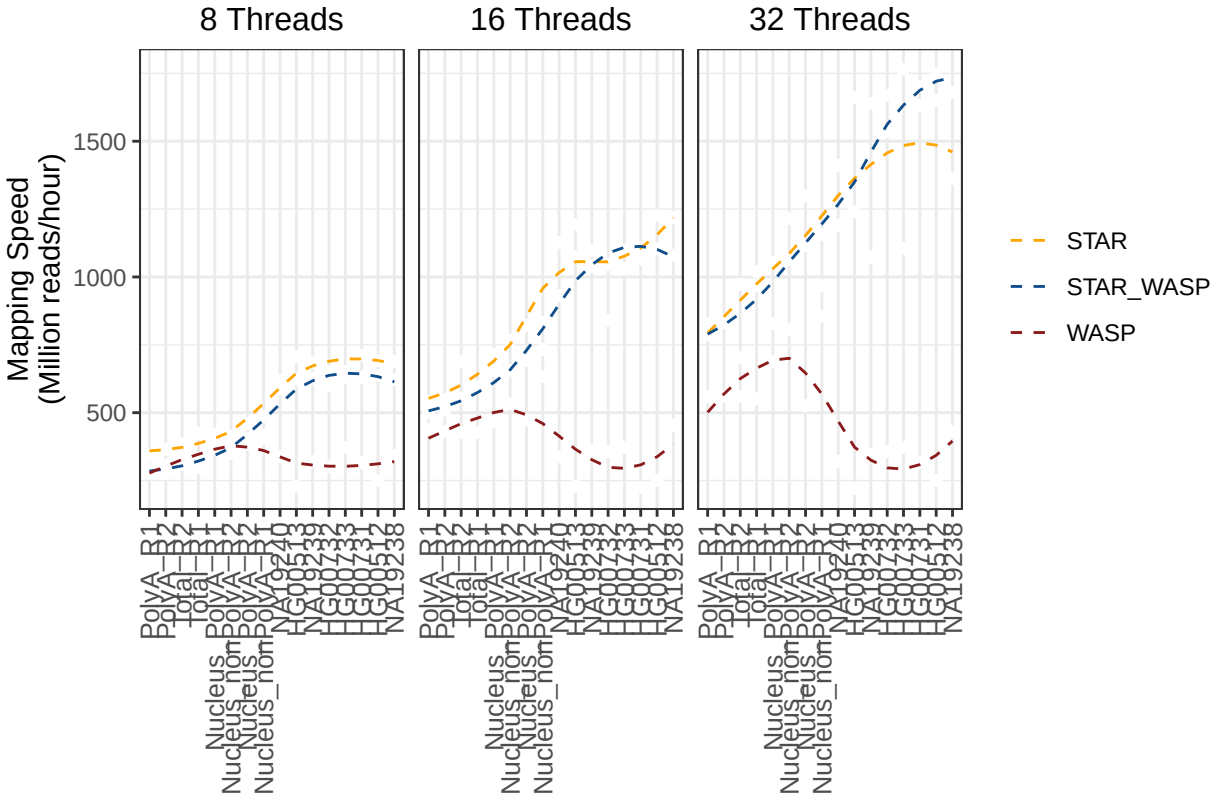
```
## [1] 38398332
```

3.2 Mapping Speed (Mapping_speed_Million_of_reads_per_hour)

** Not considered as WASP's speed is based off filtered reads, considering wall clock instead (next section)

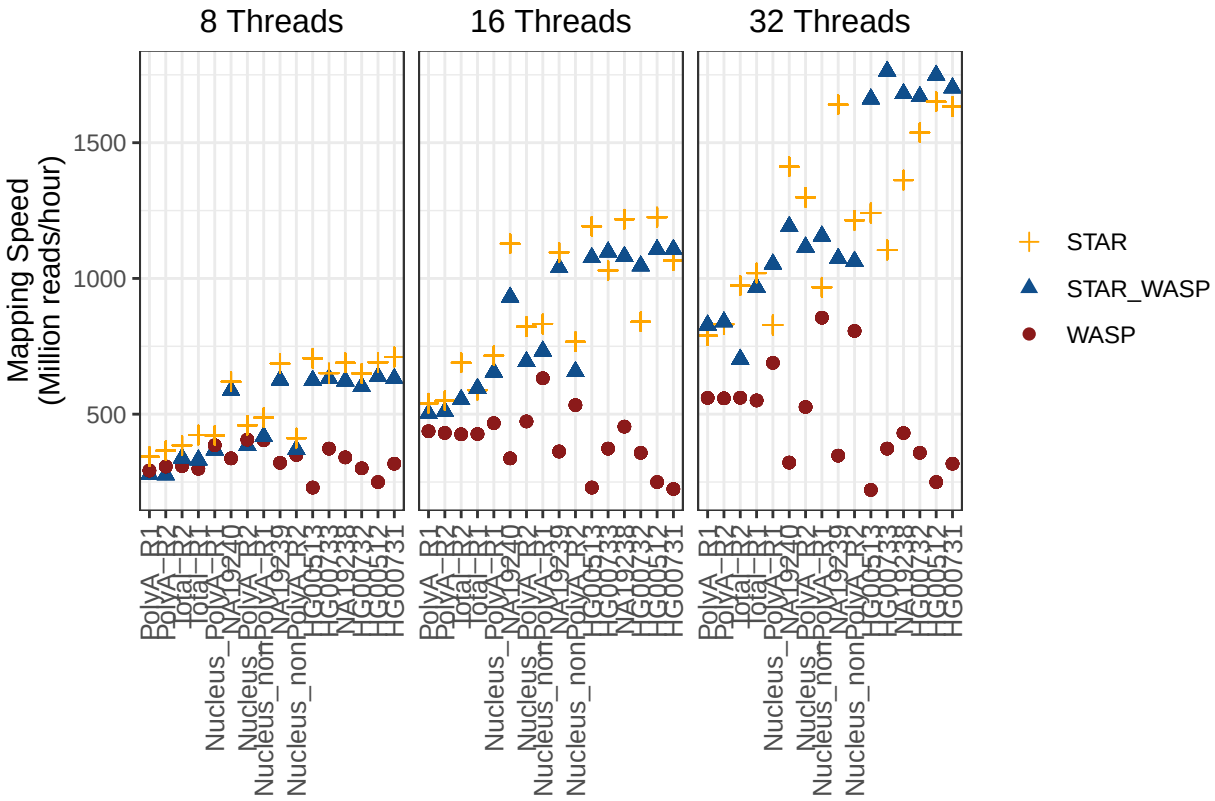
```
data_melted_beta_speed <- data_melted_beta[order(data_melted_beta$Sample, data_melted_beta$Mapping_speed

data_melted_beta_speed$Threads <- ordered(data_melted_beta_speed$Threads , levels = c("8 Threads", "16 T

data_melted_beta_speed$Value <- as.numeric(data_melted_beta_speed$Mapping_speed_Million_of_reads_per_hou
data_melted_beta_speed$Sample <- as.factor(data_melted_beta_speed$Sample)
data_melted_beta_speed$Run <- as.factor(data_melted_beta_speed$Run)

data_melted_beta_speed <- data_melted_beta_speed[order(data_melted_beta_speed$Sample, data_melted_beta_s


data_melted_beta_speed <- data_melted_beta_speed %>%
  group_by(Sample, Threads) %>%
  dplyr::mutate(speed_order = sum(Mapping_speed_Million_of_reads_per_hour)) %>%
  arrange(speed_order)

data_melted_beta_speed <- as.data.frame(data_melted_beta_speed)

# data_melted_beta_speed %>%
#   ggplot(aes(x = Sample, y = Mapping_speed_Million_of_reads_per_hour, group=Run)) +
#   geom_line(aes(color = Run, linetype = Run)) +
#   scale_color_manual(values = c("darkred", "steelblue", "cyan3")) +
#   facet_wrap(~Threads) + labs(y = "Mapping_Speed", x="") +
#   theme_light() +  theme(legend.title = element_blank()) + #Base, bw, excel_new(), few, light, lindra
#   #theme(strip.background =element_rect(fill="white"))+
#   # theme(strip.text = element_text(colour = 'black')) +
#   theme(strip.text.x = element_blank()) + theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust
# #geom_text(x = 3, y = max(data_melted_beta_memory$Value), aes(label = label), data = dat_text, check_
```

```
data_melted_beta_speed %>%
  #mutate(Sample = fct_reorder2(Sample, Threads, speed_order,  .desc = FALSE)) %>%
  ggplot(aes(x = reorder(Sample, Mapping_speed_Million_of_reads_per_hour), y = Mapping_speed_Million_of_
 # geom_line(aes(color = Run, linetype = Run)) + #stat_smooth()
  geom_point(color="white") +
  geom_smooth(se=FALSE, linetype="dashed", size=0.5) +
  #scale_color_manual(values = c("black", "gray50", "gray80" )) + #"darkred", "dodgerblue4", "orange"
    scale_color_manual(values=global_colors)+
  facet_wrap(~Threads) +
  #labs(y = "Mapping Speed (Million of reads/hour", x="") +
  labs(y = paste0("Mapping Speed", "\n", "(Million reads/hour)"), x = "")+
  theme_bw() + theme(legend.title = element_blank()) + #Base, bw, excel_new(), few, light, lindraw
  theme(strip.background =element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x = element_markdown(hjust = 0.5, size=
  #theme(strip.text.x = element_blank()) +
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))  #+
```



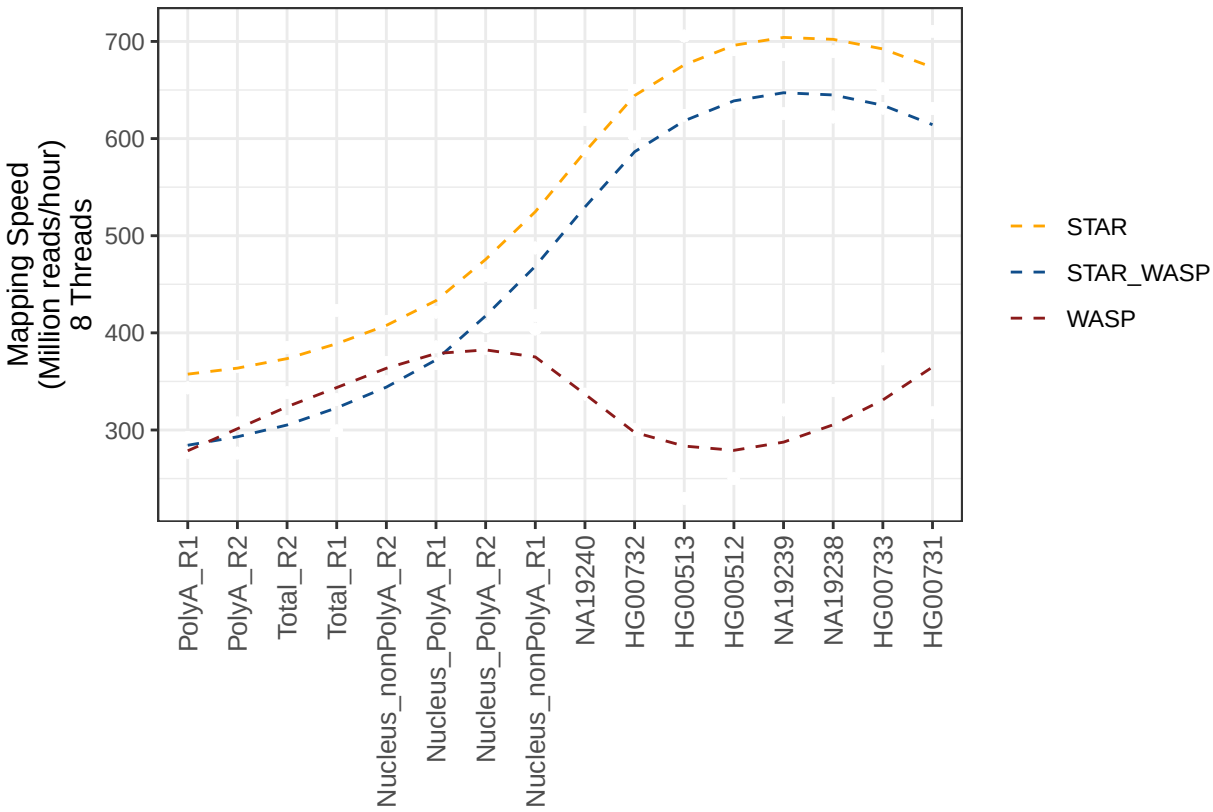

```
  #geom_text(x = 6, y = max(data_melted_beta_memory$Value), aes(label = label),
    #    data = dat_text, check_overlap = TRUE, inherit.aes = FALSE)


data_melted_beta_speed %>%
  mutate(Sample = fct_reorder2(Sample, Threads, speed_order,  .desc = FALSE)) %>%
  ggplot(aes(x = Sample, y = Mapping_speed_Million_of_reads_per_hour, group=Run)) +
  geom_point(aes(color=factor(Run),shape=factor(Run),fill=factor(Run)), size=2)+
```

```
  #geom_line()+
  scale_shape_manual(values=c(3, 17, 16))+
  scale_color_manual(values=global_colors)+
  #scale_color_manual(values = c("darkred", "dodgerblue4", "orange")) +
 facet_wrap(~Threads) +
  #labs(y = "Mapping Speed (Million of reads/hour", x="") +
  labs(y = paste0("Mapping Speed", "\n", "(Million reads/hour)"), x="")+
  theme_bw() + theme(legend.title = element_blank()) + #Base, bw, excel_new(), few, light, lindraw
  theme(strip.background =element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x = element_markdown(hjust = 0.5, size=
  #theme(strip.text.x = element_blank()) +
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10)) #+
```

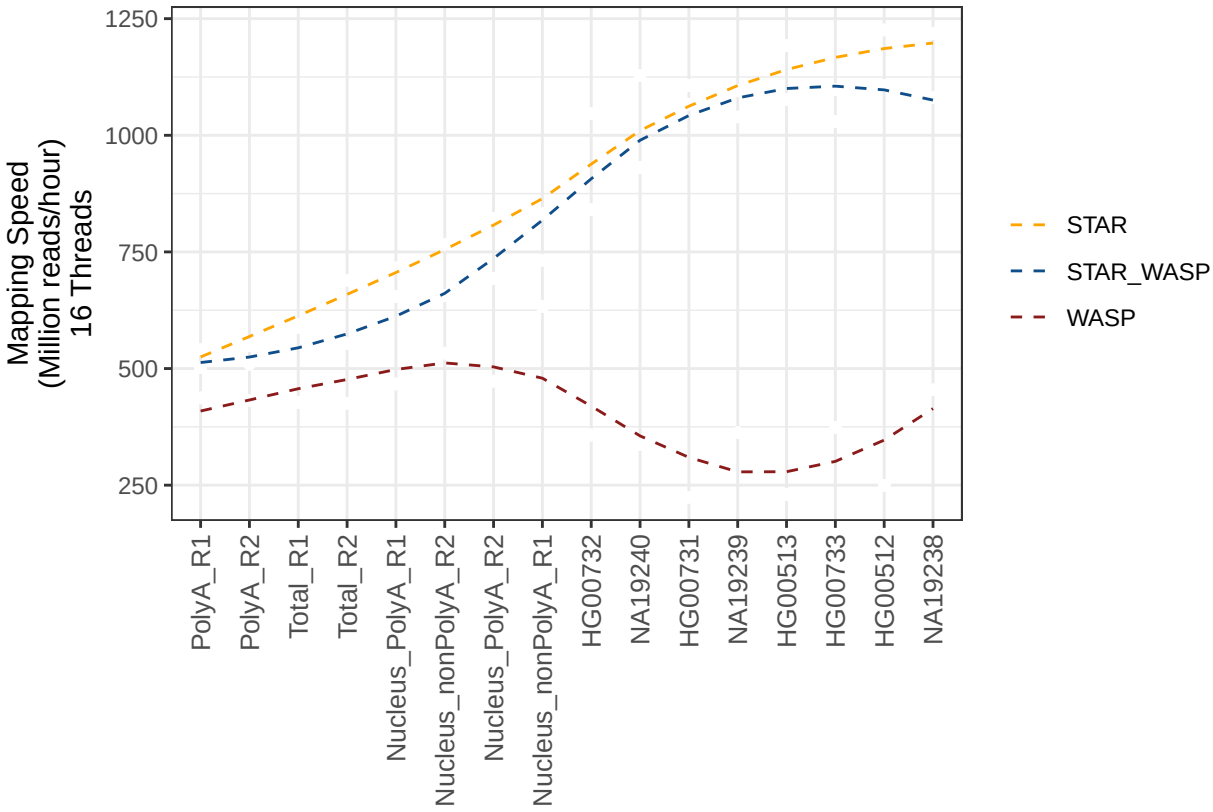



```
  #geom_text(x = 6, y = max(data_melted_beta_memory$Value), aes(label = label),
    #     data = dat_text, check_overlap = TRUE, inherit.aes = FALSE)


#Plot 1
#8 Threads
data_melted_beta_speed %>% filter (Threads == "8 Threads") %>%
  #mutate(Sample = fct_reorder2(Sample, Threads, speed_order,  .desc = FALSE)) %>%
  ggplot(aes(x = reorder(Sample, Value), y = Mapping_speed_Million_of_reads_per_hour, group=Run, color=
  #geom_line(aes(color = Run, linetype = Run)) +
  geom_point(color="white") +
  geom_smooth(se=FALSE, linetype="dashed", size=0.5) +
```

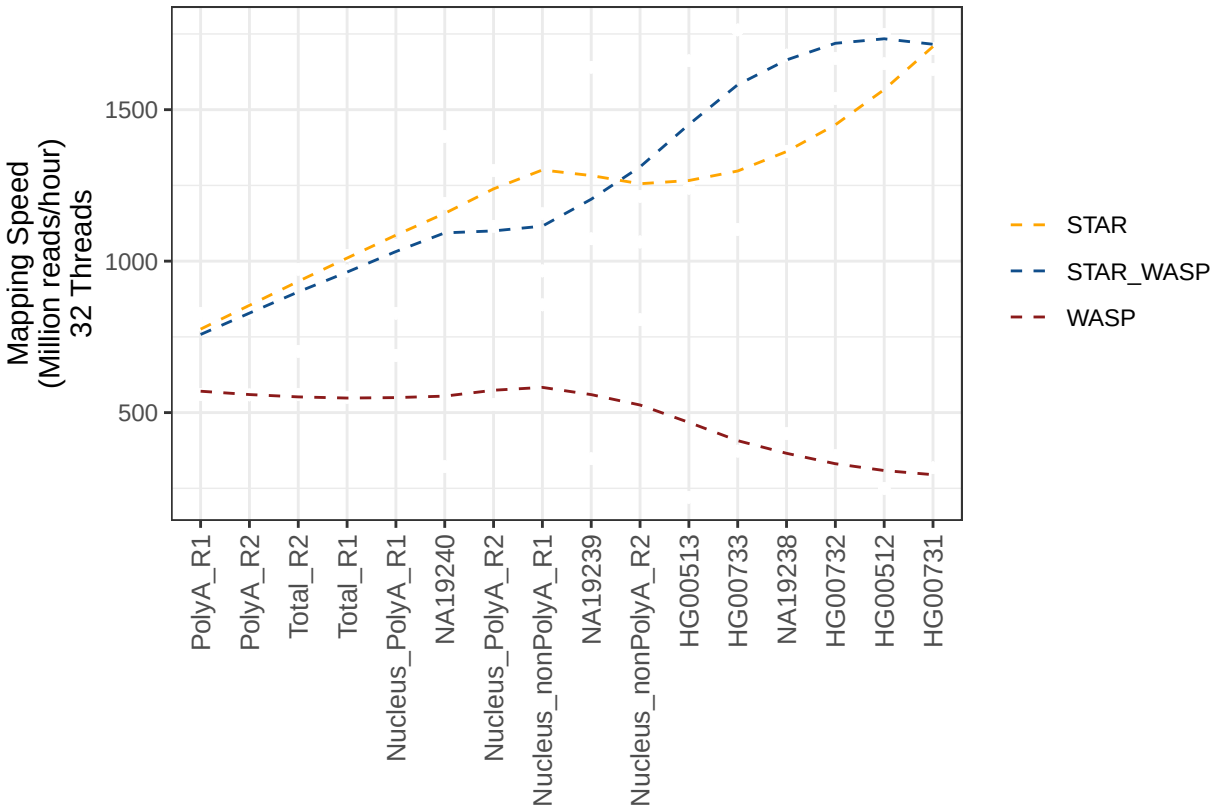
```
  scale_color_manual(values=global_colors)+
  labs(y = paste0("Mapping Speed", "\n", "(Million reads/hour)", "\n", "8 Threads"), x = "")+
  theme_bw() + theme(legend.title = element_blank()) +
  theme(strip.background =element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x = element_markdown(hjust = 0, size=12)
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))
```




```
#16 Threads
data_melted_beta_speed %>% filter(Threads == "16 Threads") %>%
  #mutate(Sample = fct_reorder2(Sample, Threads, speed_order,  .desc = FALSE)) %>%
  ggplot(aes(x = reorder(Sample, Value), y = Mapping_speed_Million_of_reads_per_hour, group=Run, color=
  #geom_line(aes(color = Run, linetype = Run)) +
  geom_point(color="white") +
  geom_smooth(se=FALSE, linetype="dashed", size=0.5) +
  scale_color_manual(values=global_colors)+
  labs(y = paste0("Mapping Speed", "\n", "(Million reads/hour)", "\n", "16 Threads"), x = "")+
  theme_bw() + theme(legend.title = element_blank()) +
  theme(strip.background =element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x = element_markdown(hjust = 0, size=12)
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))
```




```
#32 Threads
data_melted_beta_speed %>% filter(Threads == "32 Threads") %>%
 # mutate(Sample = fct_reorder2(Sample, Threads, speed_order,  .desc = FALSE)) %>%
  ggplot(aes(x = reorder(Sample, Value), y = Mapping_speed_Million_of_reads_per_hour, group=Run, color=
  #geom_line(aes(color = Run, linetype = Run)) +
   geom_point(color="white") +
  geom_smooth(se=FALSE, linetype="dashed", size=0.5) +
  scale_color_manual(values=global_colors)+
  labs(y = paste0("Mapping Speed", "\n", "(Million reads/hour)", "\n", "32 Threads"), x = "")+
  theme_bw() + theme(legend.title = element_blank()) +
  theme(strip.background =element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x = element_markdown(hjust = 0, size=12
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))
```



3.3 Speed Based on Wall Clock

Note that each sample should have the same number of input reads at the start for each run, however, for WASP, the Log.final.out file reports input reads passed or already filtered reads which are less than the initial input reads (also affects mapping speed, see example below for HG00512). We therefore need to consider speed based on the wall clock, i.e overall time taken for the run to complete

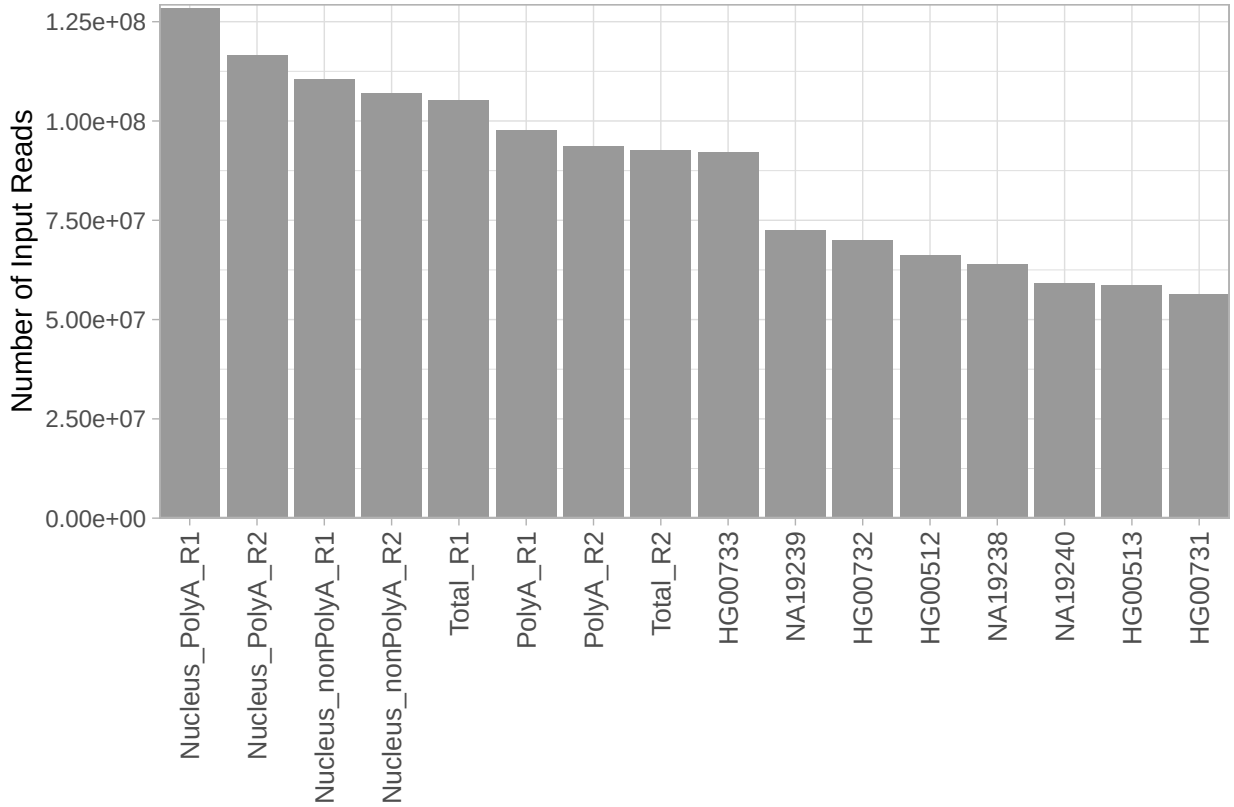
```
unique(data_melted_beta_speed[,c(1, 4:7)] %>% filter (Sample == "HG00512"))
```

```
##      Sample    Threads       Run Number_of_input_reads
## 1   HG00512  8 Threads      WASP               1110628
## 23  HG00512  8 Threads STAR_WASP              66055710
## 45  HG00512  8 Threads      STAR              66055710
## 67  HG00512 16 Threads      WASP               1110628
## 89  HG00512 16 Threads STAR_WASP              66055710
## 111 HG00512 16 Threads      STAR              66055710
## 133 HG00512 32 Threads      WASP               1110628
## 155 HG00512 32 Threads      STAR              66055710
## 177 HG00512 32 Threads STAR_WASP              66055710
##     Mapping_speed_Million_of_reads_per_hour
## 1                                    249.89
## 23                                   637.54
## 45                                   691.28
## 67                                   249.89
## 89                                  1106.05
```

```
## 111                                1225.78
## 133                                 249.89
## 155                                1651.39
## 177                                1748.53
```

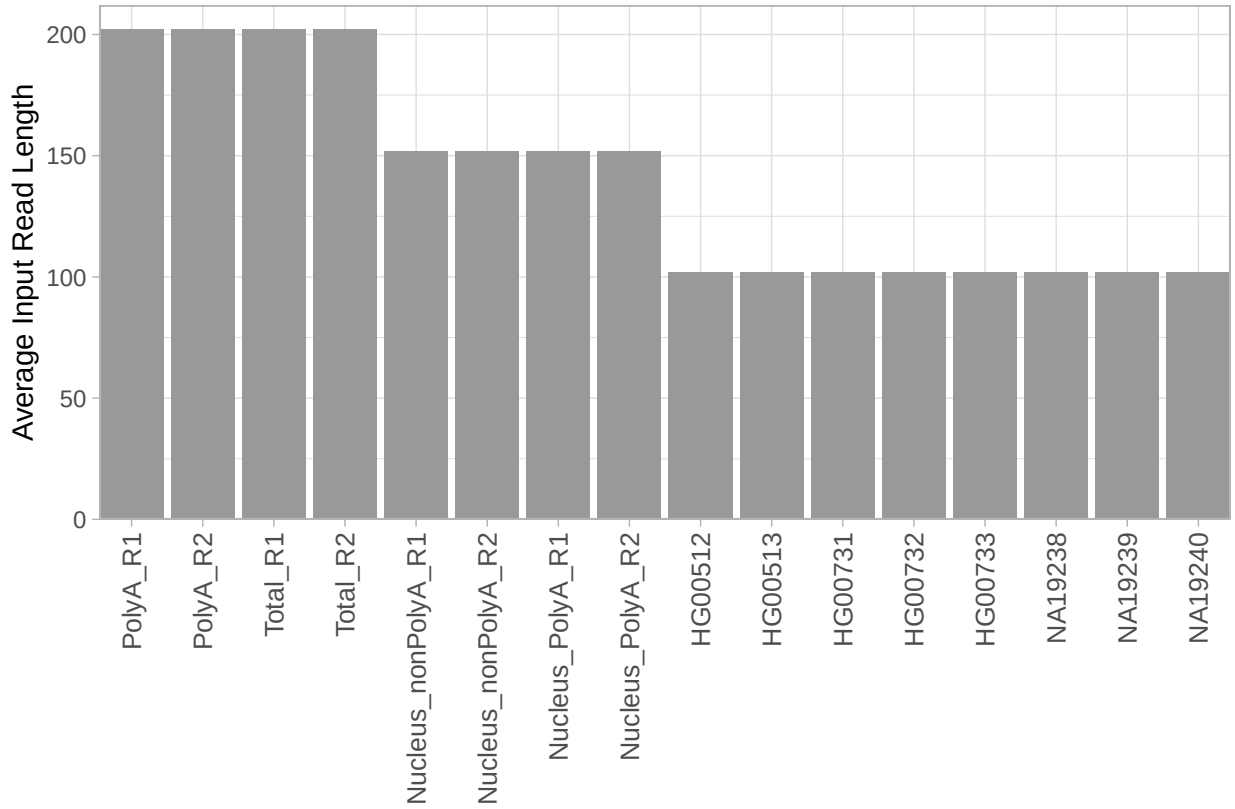
```
data_melted_beta_wall_clock <- data_melted_beta %>% filter(Param == "Wall_Clock")

# Extracting number of input reads per sample based on STAR runs - should be same input reads for each
num_input_reads_per_sample <- unique(data_melted_beta %>% filter(Run == "STAR") %>% select("Sample", "Nu
colnames(num_input_reads_per_sample)[2] <- "Number_of_input_reads_initial"
data_melted_beta_wall_clock <- inner_join(data_melted_beta_wall_clock, num_input_reads_per_sample, by =

### Number of input reads per sample:
num_input_reads_per_sample %>%
  ggplot(aes(reorder(x=Sample, -Number_of_input_reads_initial), y=Number_of_input_reads_initial)) +
  geom_bar(stat = "identity", fill="gray60") +
  theme_light() +
  scale_y_continuous(expand=c(0,0), limits=c(0, max(num_input_reads_per_sample$Number_of_input_reads_ini
  scale_x_discrete(expand=c(0,0)) +
  labs(y="Number of Input Reads", x = "") +
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))
```

Note that for this analysis, we are taking the number of input reads per sample prior to each



run

```
# Plotting by average input read length
input_read_len_per_sample <- unique(data_melted_beta %>% filter(Run == "STAR") %>% select("Sample", "Av
input_read_len_per_sample %>%
  ggplot(aes(reorder(x=Sample, -Average_input_read_length), y=Average_input_read_length)) +
  geom_bar(stat = "identity", fill="gray60") +
  theme_light() +
  scale_y_continuous(expand=c(0,0), limits = c(0, max(input_read_len_per_sample$Average_input_read_leng
  scale_x_discrete(expand=c(0,0)) +
  labs(y="Average Input Read Length", x = "") +
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))
```




```
min(num_input_reads_per_sample$Number_of_input_reads_initial)#56254714
```

```
## [1] 56254714
```

```
max(num_input_reads_per_sample$Number_of_input_reads_initial)#128402941
```

```
## [1] 128402941
```

```
mean(num_input_reads_per_sample$Number_of_input_reads_initial)#86852506
```

```
## [1] 86852506
```

```
median(num_input_reads_per_sample$Number_of_input_reads_initial)#92285172
```

```
## [1] 92285172
```

```
data_melted_beta_wall_clock$Unique_ID <- 1:nrow(data_melted_beta_wall_clock)
data_melted_beta_wall_clock$Unique_ID <- as.character(data_melted_beta_wall_clock$Unique_ID)
data_melted_beta_wall_clock$Wall_Clock <- data_melted_beta_wall_clock$Value #maintaining a copy of the

#Note that we have hours:minutes:seconds and minutes:seconds.milliseconds in our dataset - need to conv
data_melted_beta_wall_clock_min_sec_ms <- data_melted_beta_wall_clock[grepl('\\.', data_melted_beta_wall

data_melted_beta_wall_clock_min_sec_ms_split1 <- cSplit(data_melted_beta_wall_clock_min_sec_ms, "Value"

data_melted_beta_wall_clock_min_sec_ms_split <- cSplit(data_melted_beta_wall_clock_min_sec_ms_split1, "

colnames(data_melted_beta_wall_clock_min_sec_ms_split)[11] <- "minutes"
colnames(data_melted_beta_wall_clock_min_sec_ms_split)[12] <- "seconds"
colnames(data_melted_beta_wall_clock_min_sec_ms_split)[13] <- "milliseconds"

data_melted_beta_wall_clock_min_sec_ms_split$minutes <- as.numeric(as.character(data_melted_beta_wall_cl
data_melted_beta_wall_clock_min_sec_ms_split$seconds <- as.numeric(as.character(data_melted_beta_wall_cl
data_melted_beta_wall_clock_min_sec_ms_split$milliseconds <- as.numeric(as.character(data_melted_beta_wa

#Next we shall convert the different units to hours and sum them up for total time in hours
data_melted_beta_wall_clock_min_sec_ms_split <- data_melted_beta_wall_clock_min_sec_ms_split %>%  mutate
data_melted_beta_wall_clock_min_sec_ms_split <- data_melted_beta_wall_clock_min_sec_ms_split %>%  mutate
data_melted_beta_wall_clock_min_sec_ms_split <- data_melted_beta_wall_clock_min_sec_ms_split %>%  mutate
data_melted_beta_wall_clock_min_sec_ms_split <- data_melted_beta_wall_clock_min_sec_ms_split %>% mutate

#Second Subset - overall, these values have hours:minutes:seconds
data_melted_beta_wall_clock_hr_min_sec  <- as.data.frame(data_melted_beta_wall_clock %>% filter(Unique_I
data_melted_beta_wall_clock_hr_min_sec_split <- cSplit(data_melted_beta_wall_clock_hr_min_sec, "Value",
colnames(data_melted_beta_wall_clock_hr_min_sec_split)[11] <- "hrs"
colnames(data_melted_beta_wall_clock_hr_min_sec_split)[12] <- "minutes"
colnames(data_melted_beta_wall_clock_hr_min_sec_split)[13] <- "seconds"

data_melted_beta_wall_clock_hr_min_sec_split$hrs <- as.numeric(as.character(data_melted_beta_wall_clock_
data_melted_beta_wall_clock_hr_min_sec_split$minutes <- as.numeric(as.character(data_melted_beta_wall_cl
data_melted_beta_wall_clock_hr_min_sec_split$seconds <- as.numeric(as.character(data_melted_beta_wall_cl

#Next we shall convert the different units to hours and sum them up for total time in hours
data_melted_beta_wall_clock_hr_min_sec_split <- data_melted_beta_wall_clock_hr_min_sec_split %>%  mutate
data_melted_beta_wall_clock_hr_min_sec_split <- data_melted_beta_wall_clock_hr_min_sec_split %>%  mutate
data_melted_beta_wall_clock_hr_min_sec_split <- data_melted_beta_wall_clock_hr_min_sec_split %>% mutate

#Mutating speed based on wall clock (num of input reads/ time in hrs)
data_melted_beta_wall_clock_min_sec_ms_split <- data_melted_beta_wall_clock_min_sec_ms_split %>%  mutate
data_melted_beta_wall_clock_hr_min_sec_split <- data_melted_beta_wall_clock_hr_min_sec_split %>%  mutate

head(data_melted_beta_wall_clock_min_sec_ms_split)
```

```
##                 Sample      Param    Threads  Run Number_of_input_reads
```

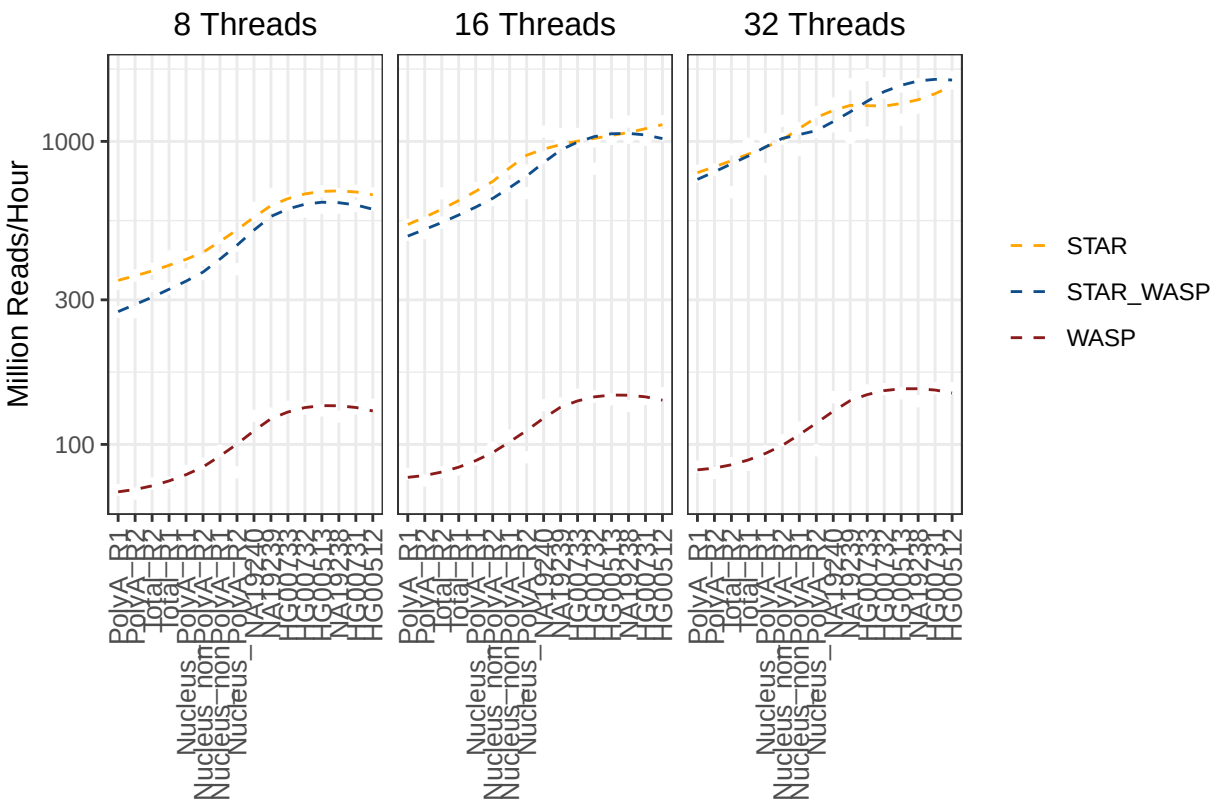
```
## 1:             HG00512 Wall_Clock 16 Threads STAR              66055710
## 2:             HG00513 Wall_Clock 16 Threads STAR              58601893
## 3:             HG00731 Wall_Clock 16 Threads STAR              56254714
## 4:             HG00732 Wall_Clock 16 Threads STAR              70029452
## 5:             HG00733 Wall_Clock 16 Threads STAR              92075712
## 6: Nucleus_nonPolyA_R1 Wall_Clock 16 Threads STAR             110469791
##    Mapping_speed_Million_of_reads_per_hour Average_input_read_length
## 1:                                1225.78                       102
## 2:                                1191.90                       102
## 3:                                1065.88                       102
## 4:                                 840.35                       102
## 5:                                1029.42                       102
## 6:                                 831.99                       152
##    Number_of_input_reads_initial Unique_ID Wall_Clock minutes seconds
## 1:                      66055710         1    3:24.40       3      24
## 2:                      58601893         2    3:06.91       3       6
## 3:                      56254714         3    3:20.14       3      20
## 4:                      70029452         4    5:10.74       5      10
## 5:                      92075712         5    5:33.08       5      33
## 6:                     110469791         6    8:10.55       8      10
##    milliseconds min_in_hrs seconds_in_hrs milliseconds_in_hrs   time_hrs
## 1:           40 0.05000000    0.006666667        1.111111e-05 0.05667778
## 2:           91 0.05000000    0.001666667        2.527778e-05 0.05169194
## 3:           14 0.05000000    0.005555556        3.888889e-06 0.05555944
## 4:           74 0.08333333    0.002777778        2.055556e-05 0.08613167
## 5:            8 0.08333333    0.009166667        2.222222e-06 0.09250222
## 6:           55 0.13333333    0.002777778        1.527778e-05 0.13612639
##    reads_per_hour
## 1:     1165460478
## 2:     1133675539
## 3:     1012513976
## 4:      813051166
## 5:      995389189
## 6:      811523702
```

```
head(data_melted_beta_wall_clock_hr_min_sec_split)
```

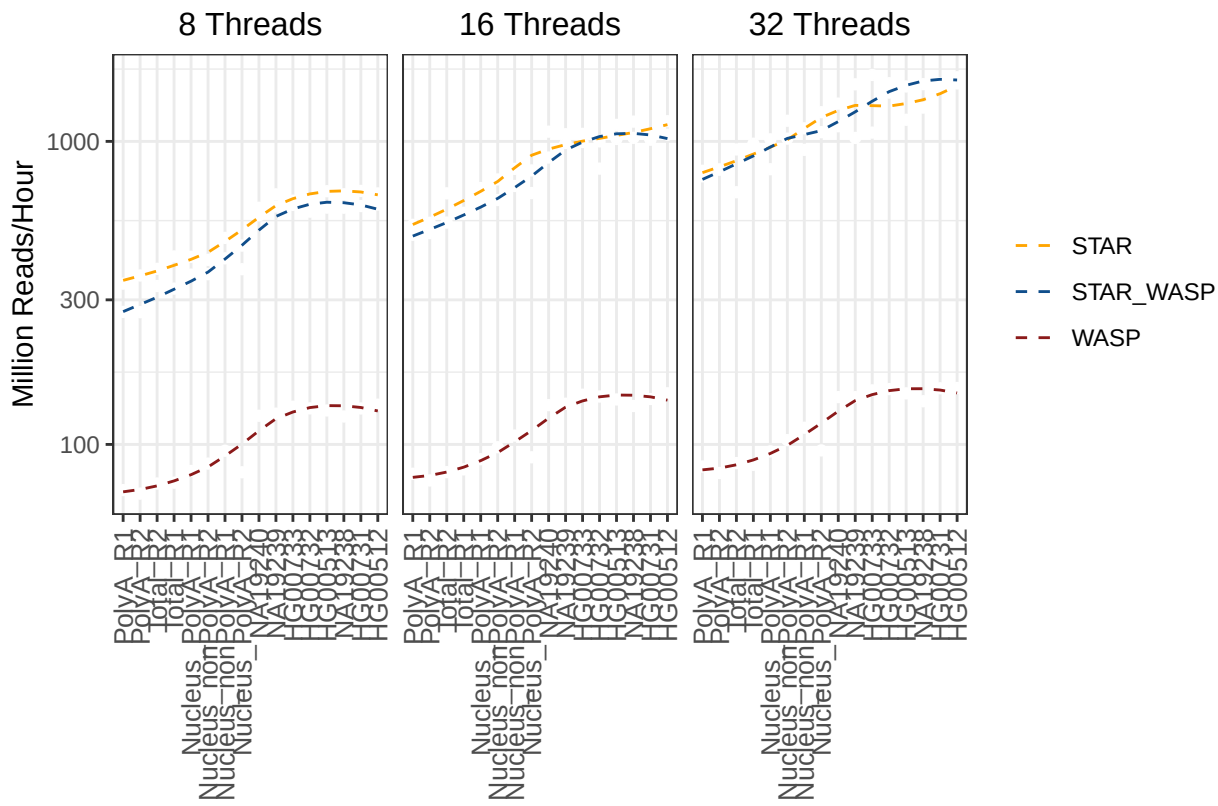
```
##                 Sample      Param    Threads  Run Number_of_input_reads
## 1: Nucleus_nonPolyA_R1 Wall_Clock 16 Threads WASP               8078842
## 2: Nucleus_nonPolyA_R2 Wall_Clock 16 Threads WASP               8737811
## 3:    Nucleus_PolyA_R1 Wall_Clock 16 Threads WASP              11671879
## 4:    Nucleus_PolyA_R2 Wall_Clock 16 Threads WASP              10382396
## 5:            PolyA_R1 Wall_Clock 16 Threads WASP              11045656
## 6:            PolyA_R2 Wall_Clock 16 Threads WASP              11009414
##    Mapping_speed_Million_of_reads_per_hour Average_input_read_length
## 1:                                 632.26                       152
## 2:                                 533.15                       152
## 3:                                 466.88                       152
## 4:                                 473.12                       152
## 5:                                 436.97                       202
## 6:                                 430.80                       202
##    Number_of_input_reads_initial Unique_ID Wall_Clock hrs minutes seconds
## 1:                     110469791       102    1:01:58   1       1      58
## 2:                     106919251       103    1:05:32   1       5      32
```

```
## 3:                    128402941       104    1:26:12   1      26      12
## 4:                    116517502       105    1:17:05   1      17       5
## 5:                     97548052       106    1:13:08   1      13       8
## 6:                     93555584       107    1:13:49   1      13      49
##    min_in_hrs seconds_in_hrs time_hrs reads_per_hour
## 1: 0.01666667    0.016111111 1.032778      106963757
## 2: 0.08333333    0.008888889 1.092222       97891481
## 3: 0.43333333    0.003333333 1.436667       89375597
## 4: 0.28333333    0.001388889 1.284722       90694704
## 5: 0.21666667    0.002222222 1.218889       80030307
## 6: 0.21666667    0.013611111 1.230278       76044277
```

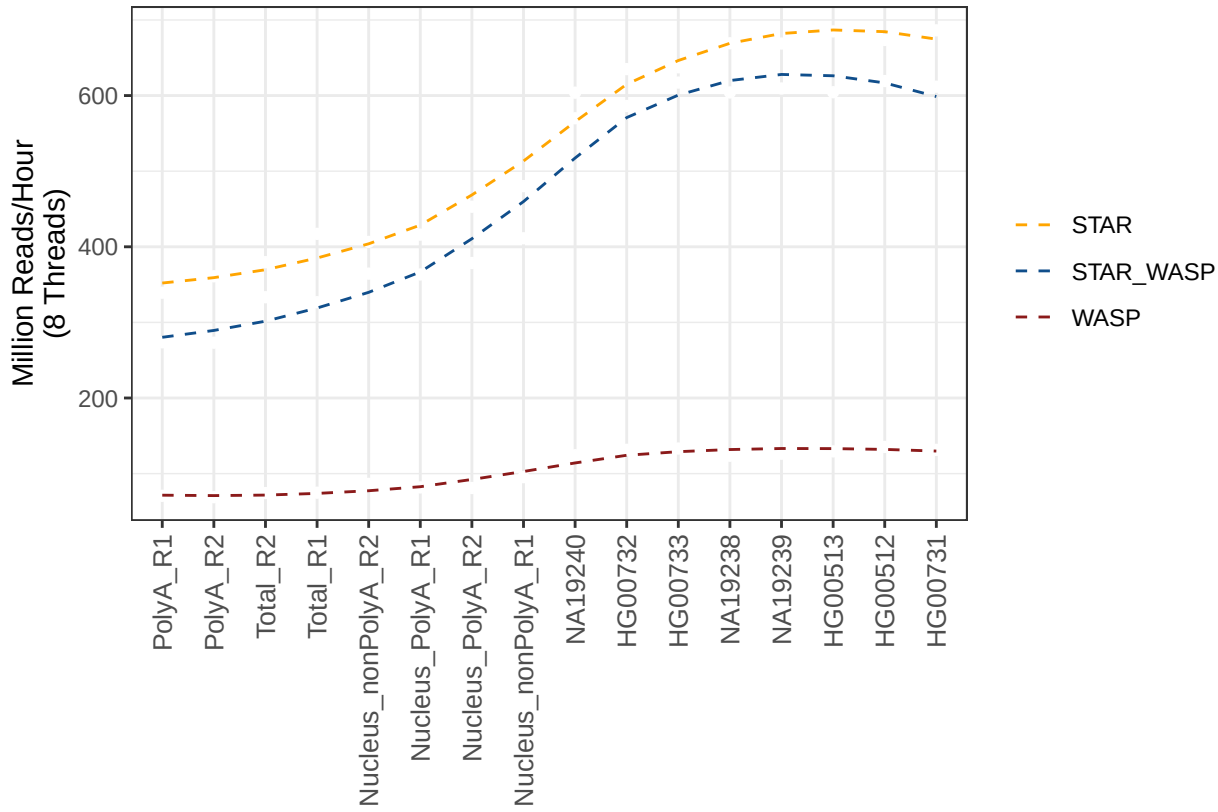
```
# Subsetting datasets to required plotting variables ( Sample      Param    Threads  Run Mapping_speed_
data_melted_beta_wall_clock_min_sec_ms_split <- data_melted_beta_wall_clock_min_sec_ms_split[,c(1:4, 6,
data_melted_beta_wall_clock_hr_min_sec_split <- data_melted_beta_wall_clock_hr_min_sec_split[,c(1:4, 6,

data_melted_beta_wall_clock_converted <- rbind(data_melted_beta_wall_clock_min_sec_ms_split, data_melte

# Plotting wall clock
data_melted_beta_wall_clock_converted %>%
  #mutate(Sample = fct_reorder(Sample, reads_per_hour)) %>%
  ggplot(aes(x = reorder(Sample, reads_per_hour/1000000), y = reads_per_hour/1000000, group=Run, color=
  #geom_line(aes(color = Run, linetype = Run)) +
 geom_point(color="white") +
  geom_smooth(se=FALSE, linetype="dashed", size=0.5) +
  scale_color_manual(values = global_colors) +
  facet_wrap(~Threads) +
  labs(y = "Million Reads/Hour", x = "")+
  theme_bw() + theme(legend.title = element_blank()) + scale_y_continuous(trans='log10') +
  theme(strip.background = element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x = element_markdown(hjust = 0.5, size=
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))
```



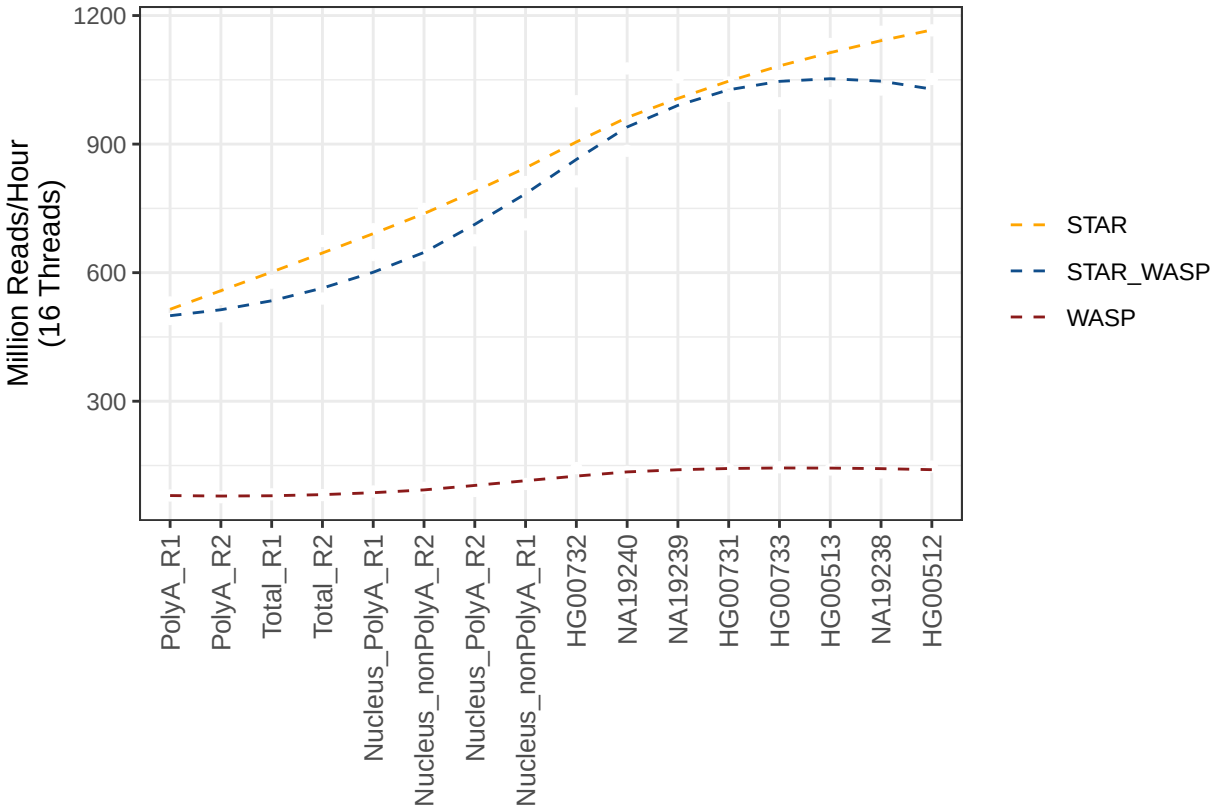

```
  data_melted_beta_wall_clock_converted %>%
  #mutate(Sample = fct_reorder(Sample, reads_per_hour)) %>%
  ggplot(aes(x = reorder(Sample, reads_per_hour/1000000), y = reads_per_hour/1000000, group=Run, color=
  #geom_line(aes(color = Run, linetype = Run)) +
geom_point(color="white") +
  geom_smooth(se=FALSE, linetype="dashed", size=0.5) +
  scale_color_manual(values=global_colors)+
  facet_wrap(~Threads) +
  labs(y = "Million Reads/Hour", x = "")+
  theme_bw() + theme(legend.title = element_blank()) + scale_y_continuous(trans='log10') +
  theme(strip.background = element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x = element_markdown(hjust = 0.5, size=
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))
```



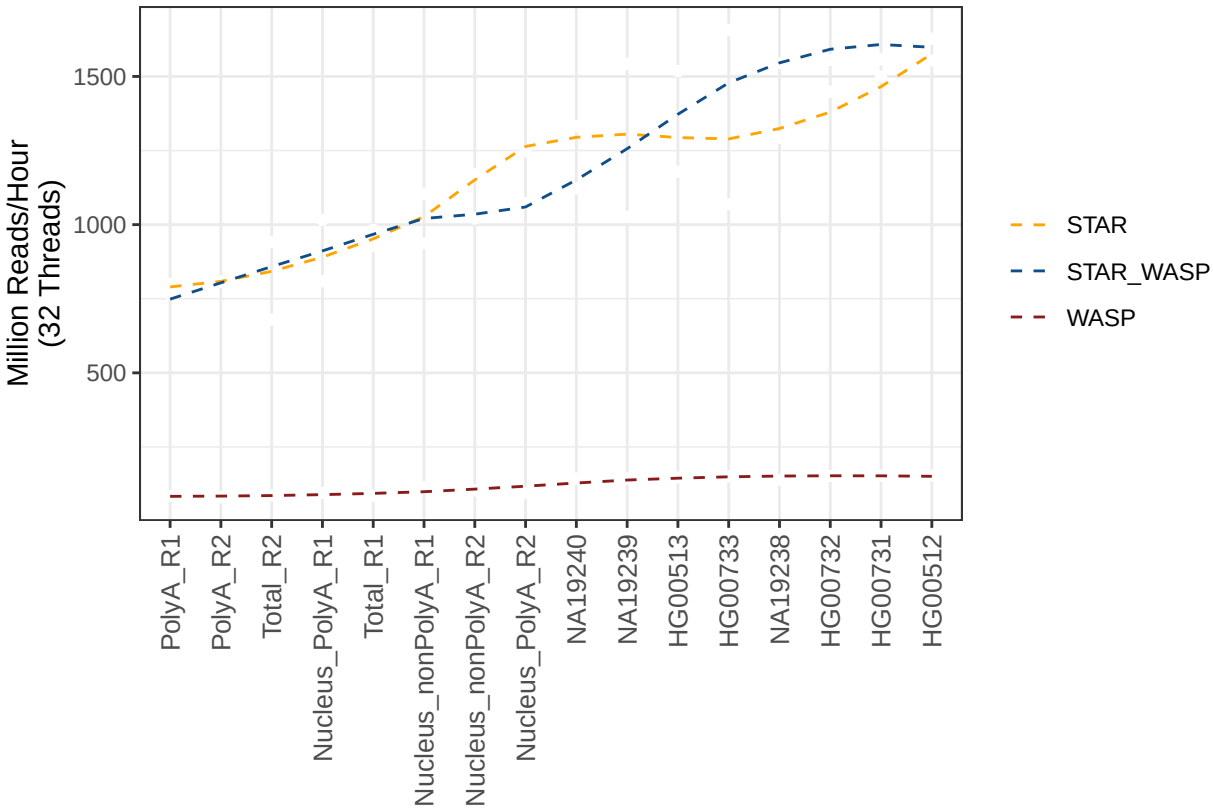

```
# 8 threads
data_melted_beta_wall_clock_converted %>% filter(Threads == "8 Threads") %>%
  ggplot(aes(x = reorder(Sample, reads_per_hour/1000000), y = reads_per_hour/1000000, group=Run, color=
   geom_point(color="white") +
  geom_smooth(se=FALSE, linetype="dashed", size=0.5) +
  scale_color_manual(values = global_colors) +
  labs(y = paste0("Million Reads/Hour", "\n", "(8 Threads)"), x = "")+
  theme_bw() + theme(legend.title = element_blank()) +
  theme(strip.background =element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x = element_markdown(hjust = 0.5, size=
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))
```



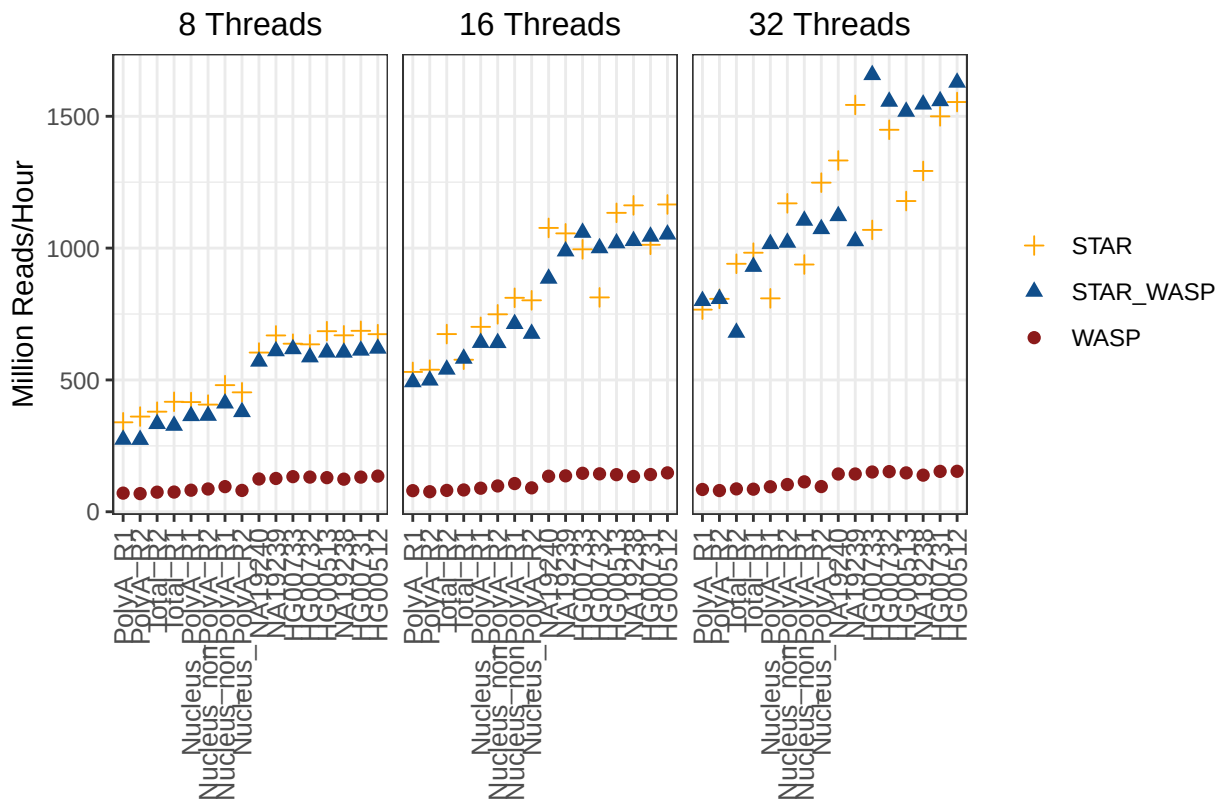

```
# 16 threads
data_melted_beta_wall_clock_converted %>% filter(Threads == "16 Threads") %>%
  ggplot(aes(x = reorder(Sample, reads_per_hour/1000000), y = reads_per_hour/1000000, group=Run, color=
  geom_point(color="white") +
  geom_smooth(se=FALSE, linetype="dashed", size=0.5) +
  scale_color_manual(values = global_colors) +
  labs(y = paste0("Million Reads/Hour", "\n", "(16 Threads)"), x = "")+
  theme_bw() + theme(legend.title = element_blank()) +
  theme(strip.background =element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x = element_markdown(hjust = 0)) +
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))
```



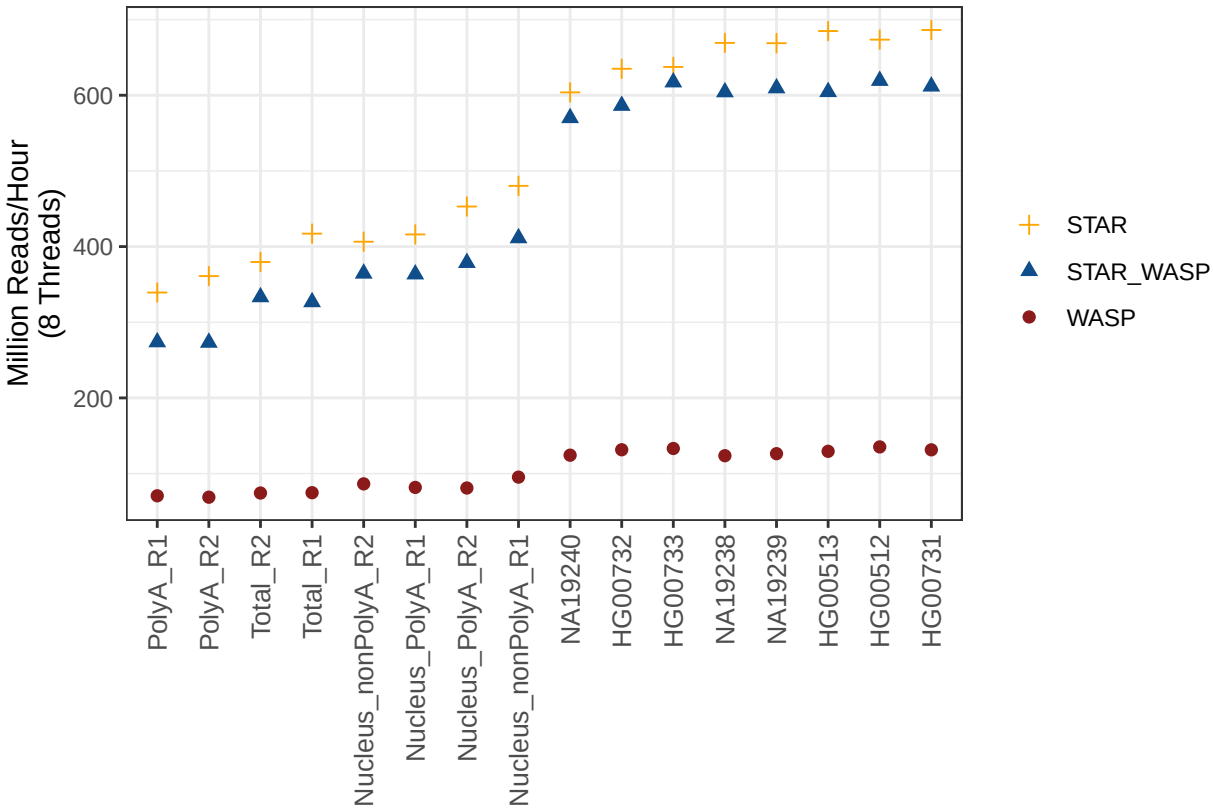

```
# 32 threads
data_melted_beta_wall_clock_converted %>%  filter(Threads == "32 Threads") %>%
  ggplot(aes(x = reorder(Sample, reads_per_hour/1000000), y = reads_per_hour/1000000, group=Run, color=
  geom_point(color="white") +
  geom_smooth(se=FALSE, linetype="dashed", size=0.5) +
  scale_color_manual(values =global_colors) +
  labs(y = paste0("Million Reads/Hour", "\n", "(32 Threads)"), x = "")+
  theme_bw() + theme(legend.title = element_blank()) +
  theme(strip.background =element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x = element_markdown(hjust = 0.5, size=
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))
```



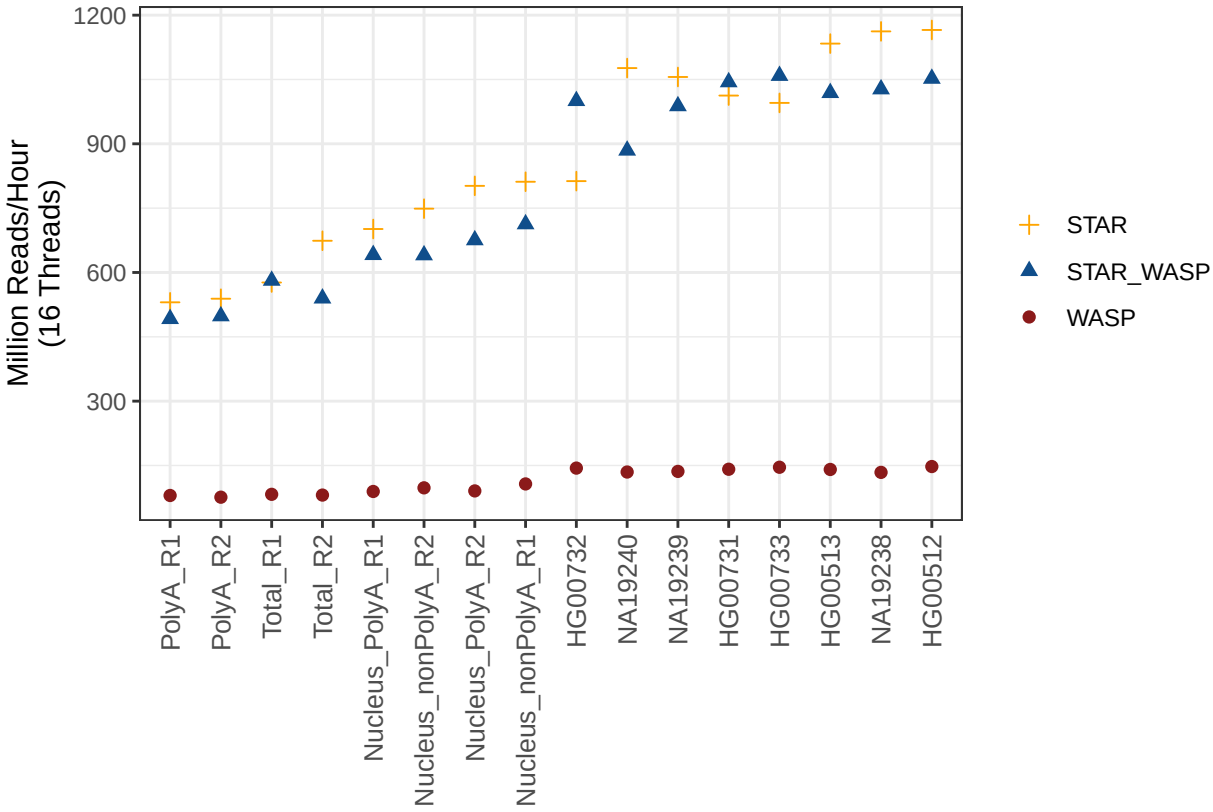

```
# Representation 2
data_melted_beta_wall_clock_converted %>%
  ggplot(aes(x = reorder(Sample, reads_per_hour/1000000), y = reads_per_hour/1000000, group=Run)) +
  geom_point(aes(color=factor(Run),shape=factor(Run),fill=factor(Run)), size=2)+#,alpha=.8
  scale_shape_manual(values=c(3, 17, 16))+
  facet_wrap(~Threads) +
  #scale_color_manual(values=c('gray50','firebrick3', '#56B4E9'))+
  scale_color_manual(values=global_colors)+
  labs(y = "Million Reads/Hour", x="")+
  theme_bw() + theme(legend.title = element_blank()) +
  theme(strip.background =element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x =  element_markdown(hjust = 0.5, size=
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))
```



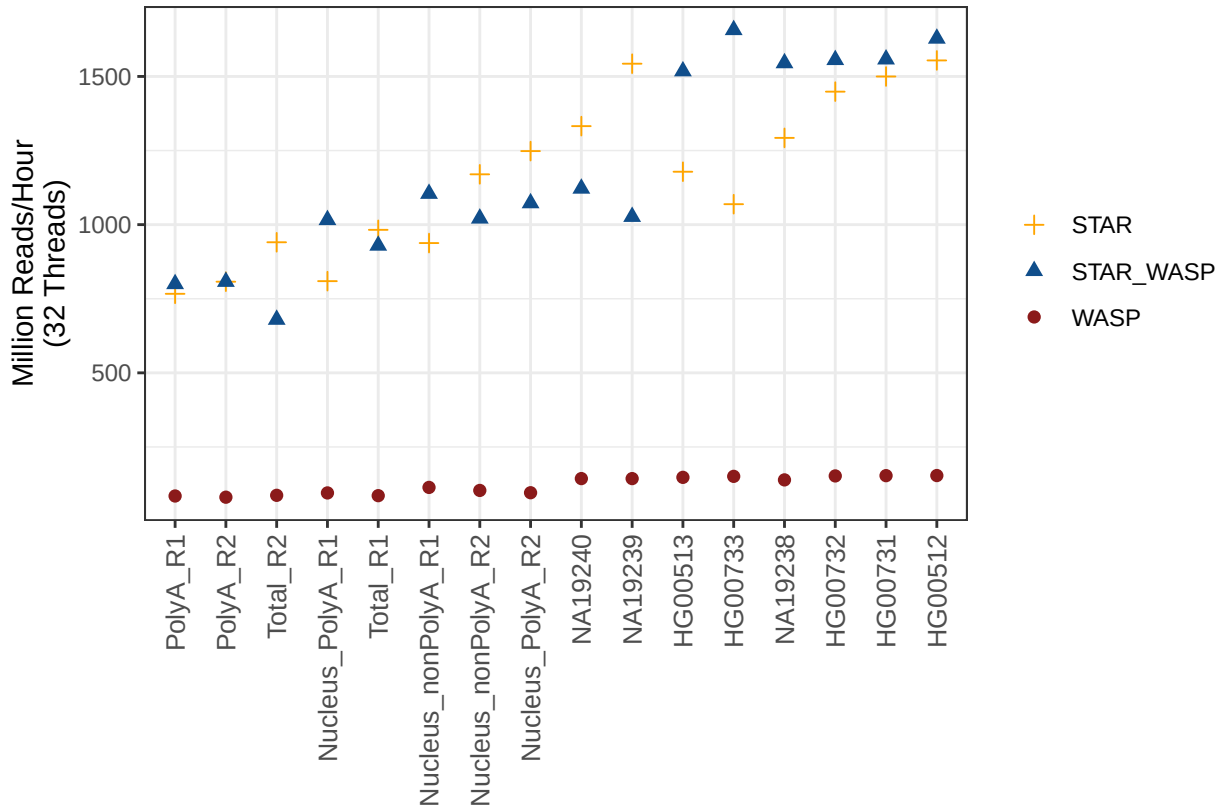

```
data_melted_beta_wall_clock_converted %>% filter(Threads == "8 Threads") %>%
  ggplot(aes(x = reorder(Sample, reads_per_hour/1000000), y = reads_per_hour/1000000, group=Run)) +
  geom_point(aes(color=factor(Run),shape=factor(Run),fill=factor(Run)), size=2)+#,alpha=.8
  scale_shape_manual(values=c(3, 17, 16))+
  #scale_color_manual(values=c('gray50','firebrick3', '#56B4E9'))+
  scale_color_manual(values=global_colors)+
  labs(y = paste0("Million Reads/Hour", "\n", "(8 Threads)"), x="")+
  theme_bw() + theme(legend.title = element_blank()) +
  theme(strip.background =element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x =  element_markdown(hjust = 0.5, size=
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))
```




```
data_melted_beta_wall_clock_converted %>% filter(Threads == "16 Threads") %>%
  ggplot(aes(x = reorder(Sample, reads_per_hour/1000000), y = reads_per_hour/1000000, group=Run)) +
  geom_point(aes(color=factor(Run),shape=factor(Run),fill=factor(Run)), size=2)+#,alpha=.8
  scale_shape_manual(values=c(3, 17, 16))+
  scale_color_manual(values=global_colors)+
  labs(y = paste0("Million Reads/Hour", "\n", "(16 Threads)"), x="")+
  theme_bw() + theme(legend.title = element_blank()) +
  theme(strip.background =element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x =  element_markdown(hjust = 0.5, size=
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))
```




```
data_melted_beta_wall_clock_converted %>% filter(Threads == "32 Threads") %>%
  ggplot(aes(x = reorder(Sample, reads_per_hour/1000000), y = reads_per_hour/1000000, group=Run)) +
  geom_point(aes(color=factor(Run),shape=factor(Run),fill=factor(Run)), size=2)+#,alpha=.8
  scale_shape_manual(values=c(3, 17, 16))+
  scale_color_manual(values=global_colors)+
  labs(y = paste0("Million Reads/Hour", "\n", "(32 Threads)"), x="")+
  theme_bw() + theme(legend.title = element_blank()) +
  theme(strip.background =element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x =  element_markdown(hjust = 0.5, size=
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))
```

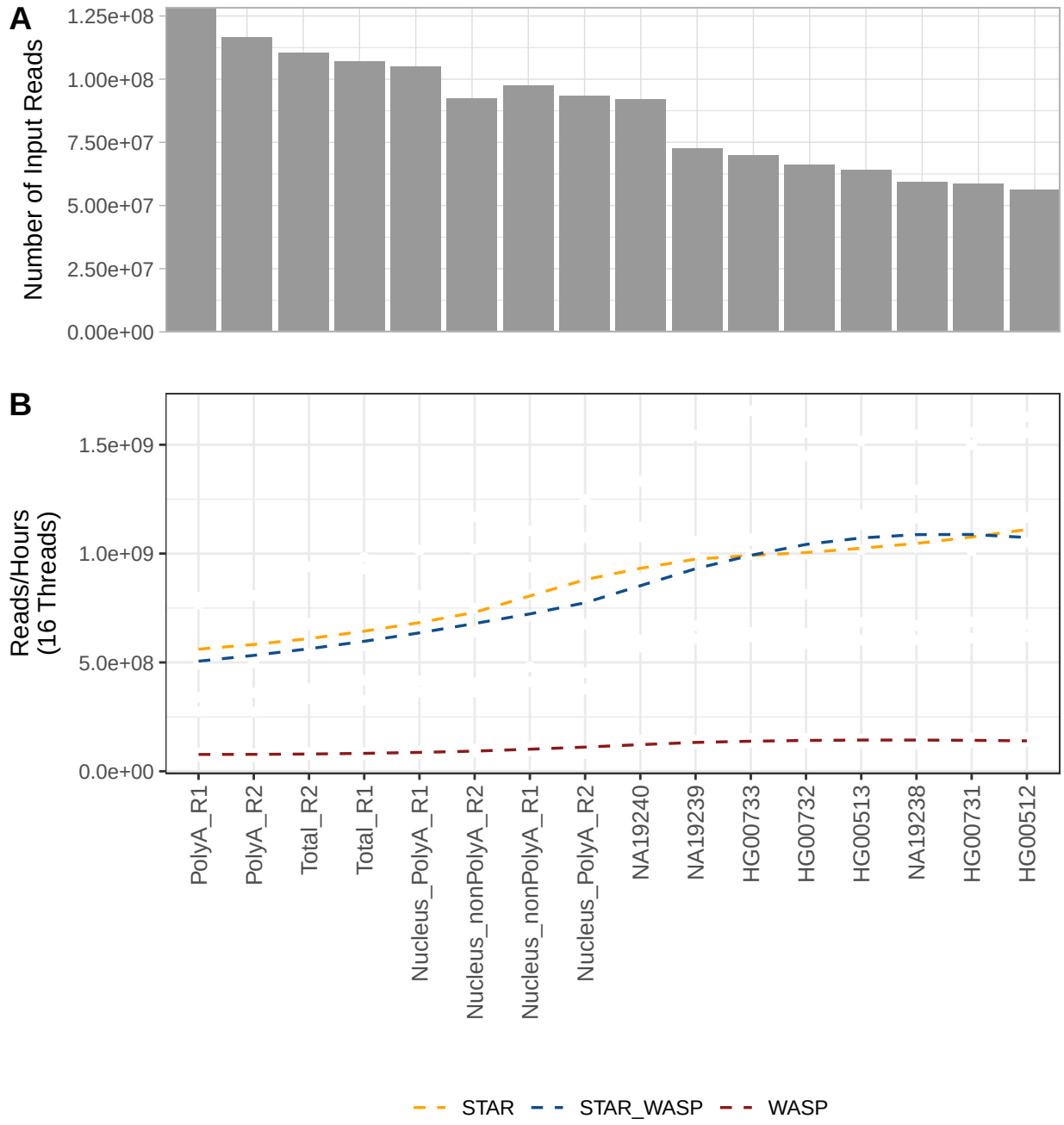



```
## Passing sample order to number of reads per sample plot
order_df <- c(
  "Nucleus_PolyA_R1", "Nucleus_PolyA_R2",
  "Nucleus_nonPolyA_R1", "Nucleus_nonPolyA_R2",
  "Total_R1", "Total_R2",
  "PolyA_R1", "PolyA_R2",
  "HG00733", "NA19239", "HG00732", "HG00512","NA19238","NA19240","HG00513", "HG00731")

# num_input_reads_per_sample %>%
#     ggplot(aes(x= factor(Sample, levels=order_df), y=Number_of_input_reads_initial)) +
#   geom_bar(stat = "identity", fill="gray60") +
#   theme_light() +
#   scale_y_continuous(expand=c(0,0))+
#   scale_x_discrete(expand=c(0,0)) +
#   labs("Number of Input Reads", x = "") +
#   theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))


a <- num_input_reads_per_sample %>%
#  ggplot(aes(reorder(x=Sample, -Number_of_input_reads_initial), y=Number_of_input_reads_initial)) +
    ggplot(aes(x= factor(Sample, levels=order_df), y=Number_of_input_reads_initial)) +
  geom_bar(stat = "identity", fill="#999999") +
  theme_light() +
  scale_y_continuous(expand=c(0,0)) +#, limits = c(0, max(num_input_reads_per_sample$Number_of_input_re
  scale_x_discrete(expand=c(0,0)) +
  labs(y="Number of Input Reads", x = "") +
```

```
  theme(axis.text.x = element_blank(), axis.ticks.x = element_blank())


b <- data_melted_beta_wall_clock_converted %>%
  ggplot(aes(x = reorder(Sample, reads_per_hour), y = reads_per_hour, group=Run, color=Run)) +
  geom_point(color="white") +
  geom_smooth(se=FALSE, linetype="dashed", size=0.6) +
 # scale_color_manual(values = c("black", "gray50", "gray80" )) +
    scale_color_manual(values=global_colors)+
  labs(y = paste0("Reads/Hours", "\n", "(16 Threads)"), x = "")+
  theme_bw() + theme(legend.title = element_blank(), legend.position = "bottom") +
  theme(strip.background =element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x = element_markdown(hjust = 0)) +
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))

plot_grid(a, b, ncol = 1, nrow=2, labels = "AUTO", rel_heights = c(0.5,1),  rel_widths = c(1,1), align =
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
## Warning: Graphs cannot be horizontally aligned unless the axis parameter is set.
## Placing graphs unaligned.
```

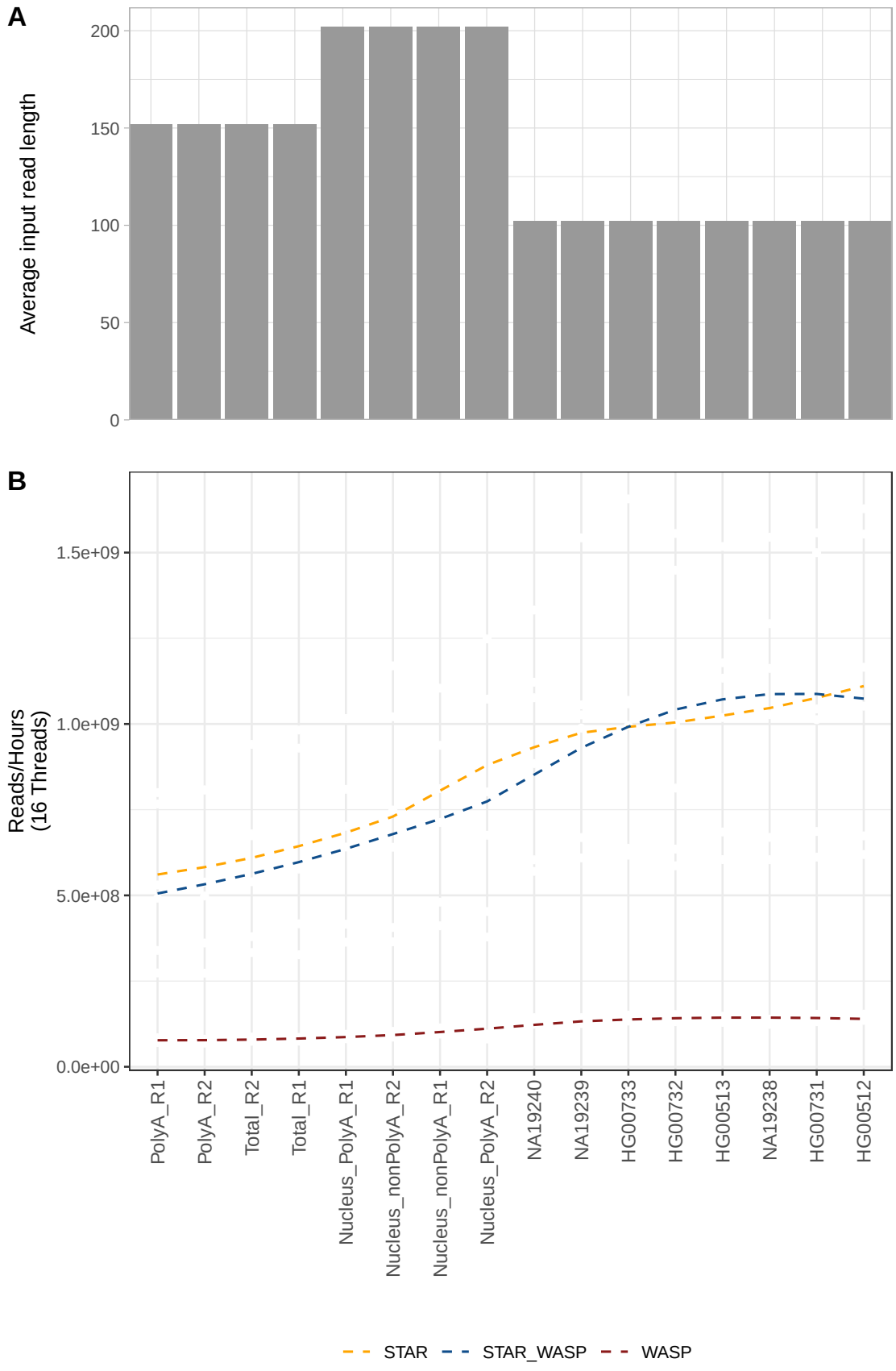



```
## By average input read length
c <- input_read_len_per_sample %>%
  ggplot(aes(x= factor(Sample, levels=order_df), y=Average_input_read_length)) +
  geom_bar(stat = "identity", fill="gray60") +
  theme_light() +
  scale_y_continuous(expand=c(0,0), limits = c(0, max(input_read_len_per_sample$Average_input_read_leng
  scale_x_discrete(expand=c(0,0)) +
  labs(y="Average input read length", x = "") +
  theme(axis.text.x = element_blank(), axis.ticks.x = element_blank())

d <- data_melted_beta_wall_clock_converted %>%
```

```
  ggplot(aes(x = reorder(Sample, reads_per_hour), y = reads_per_hour, group=Run, color=Run)) +
  geom_point(color="white") +
  geom_smooth(se=FALSE, linetype="dashed", size=0.6) +
 # scale_color_manual(values = c("black", "gray50", "gray80" )) +
    scale_color_manual(values=global_colors)+
  labs(y = paste0("Reads/Hours", "\n", "(16 Threads)"), x = "")+
  theme_bw() + theme(legend.title = element_blank(), legend.position = "bottom") +
  theme(strip.background =element_rect(fill="white", colour = "white"))+
  theme(strip.text = element_text(colour = 'black'), strip.text.x = element_markdown(hjust = 0)) +
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))

plot_grid(c, d, ncol = 1, nrow=2, labels = "AUTO", rel_heights = c(0.5,1),  rel_widths = c(1,1), align =
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
## Warning: Graphs cannot be horizontally aligned unless the axis parameter is set.
## Placing graphs unaligned.
```


A
Average input read length
200
150
100
50
0
B
Reads/Hours
(16 Threads)
1.5e+09
1.0e+09
5.0e+08
0.0e+00
PolyA_R1
PolyA_R2
Total_R2
Total_R1
Nucleus_PolyA_R1
Nucleus_nonPolyA_R2
Nucleus_nonPolyA_R1
Nucleus_PolyA_R2
NA19240
NA19239
HG00733
HG00732
HG00513
NA19238
HG00731
HG00512
STAR
STAR_WASP
WASP

4.0 vW_Tag Distributions for alignments that did not pass WASP filtering

4.1 Overal Read Distribution (Reads with vs those without tags (did not overlap variants))

```
num_reads <- as.data.frame(read.table("/home/asiimwe/projects/run_env/alpha_star_wasp_benchmarking/datal

colnames(num_reads)
```

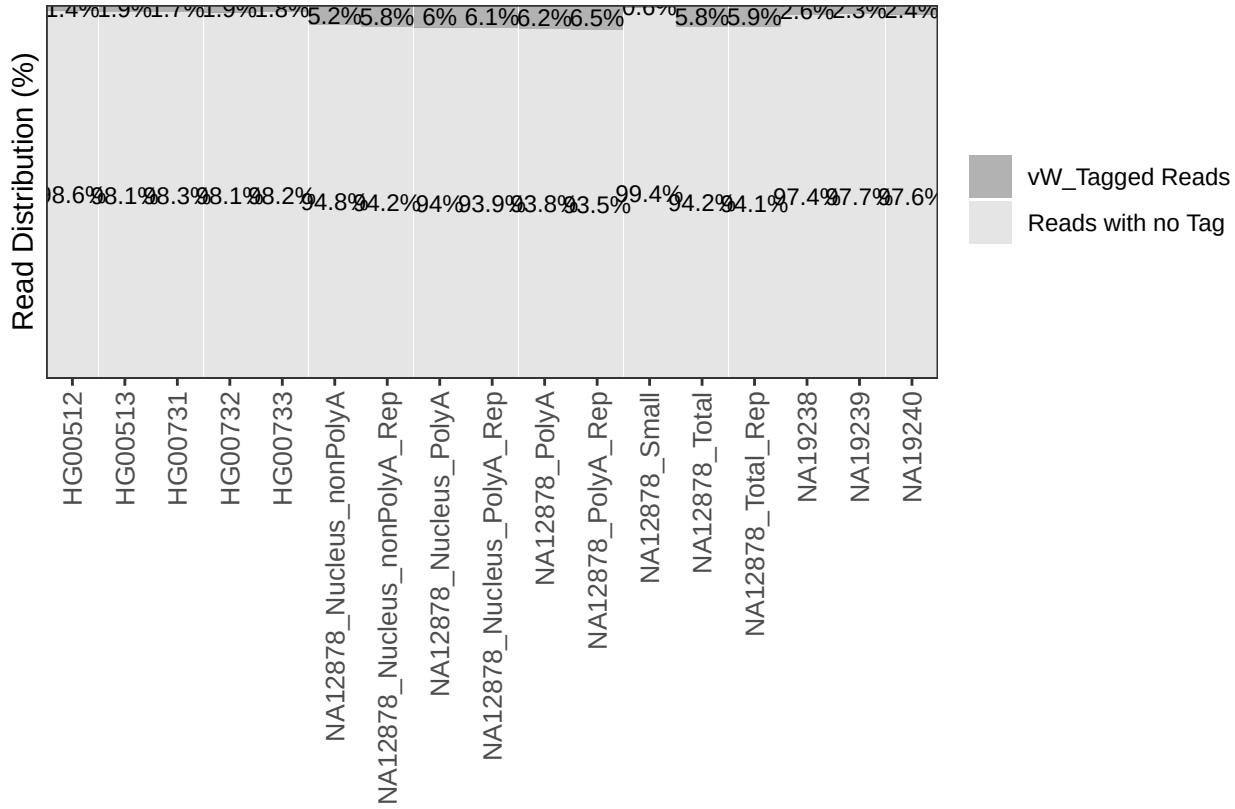
```
## [1] "V1" "V2"
```

```
colnames(num_reads) <- c("Number_of_Reads", "Read_File_Path")
num_reads <- cSplit(num_reads, "Read_File_Path", "/", direction = "wide")
num_reads <- num_reads[,-c(2:9, 11)]
num_reads$Read_File_Path_09 <- as.character(num_reads$Read_File_Path_09)
num_reads$Read_File_Path_11 <- as.character(num_reads$Read_File_Path_11)

colnames(num_reads) <- c("Number_of_Reads", "Sample", "Flag")
num_reads$Flag[num_reads$Flag == "WASP_Reads_Sorted_Unique"] <- "Num_Reads"
num_reads$Flag[num_reads$Flag == "STAR_WASP_vW_Tagged_Reads_Unique"] <- "vW_Tagged_Reads"

#Converting data frame to short format
num_reads_reshaped <- reshape(num_reads, idvar = "Sample", timevar = "Flag", direction = "wide")
colnames(num_reads_reshaped)[2] <- "Num_Reads"
colnames(num_reads_reshaped)[3] <- "vW_Tagged_Reads"

# Mutating frequencies
num_reads_reshaped <- num_reads_reshaped %>% mutate("perc_tagged" = ((vW_Tagged_Reads/Num_Reads)*100))
num_reads_reshaped <- num_reads_reshaped %>% mutate("perc_untagged" = 100 - perc_tagged)

# Plotting distributions
num_reads_reshaped_tagged <- as.data.frame(num_reads_reshaped[, c(1,4)])
num_reads_reshaped_notags <- as.data.frame(num_reads_reshaped[, c(1,5)])
colnames(num_reads_reshaped_tagged)[2] <- "perc"
colnames(num_reads_reshaped_notags)[2] <- "perc"
num_reads_reshaped_tagged$Flag <- "vW_Tagged Reads"
num_reads_reshaped_notags$Flag <- "Reads with no Tag"

num_reads_reshaped2 <- rbind(num_reads_reshaped_tagged, num_reads_reshaped_notags)


num_reads_reshaped2$Flag <- ordered(num_reads_reshaped2$Flag , levels = c("vW_Tagged Reads", "Reads witl

unique(num_reads_reshaped2$Sample)
```

```
##  [1] "HG00731"                     "NA12878_Small"
##  [3] "NA12878_Nucleus_PolyA_Rep"   "HG00513"
##  [5] "NA12878_RAMPAGE"             "HG00512"
##  [7] "NA12878_Nucleus_nonPolyA_Rep" "NA12878_RAMPAGE_Rep"
##  [9] "NA12878_Nucleus_nonPolyA"    "NA12878_Nucleus_PolyA"
## [11] "NA12878_Total"               "NA19238"
## [13] "HG00733"                     "NA12878_PolyA"
## [15] "HG00732"                     "NA12878_PolyA_Rep"
```

```
## [17] "NA12878_Total_Rep"            "NA19239"
## [19] "NA19240"
```

```
num_reads_reshaped2 <- num_reads_reshaped2 %>% filter(Sample != "NA12878_RAMPAGE_Rep" & Sample != "NA128

# PLot
num_reads_reshaped2[order(num_reads_reshaped2$Flag, decreasing = T),] %>%
ggplot() +
  geom_bar(aes(x = Sample,
               y = perc,
               group = Sample,
               fill = factor(Flag, levels=c("vW_Tagged Reads","Reads with no Tag" ))),
           stat = "identity", width = 0.99, alpha=0.5) +
  geom_text(aes(x = Sample,
                y = perc,
                label = paste0(round(perc,1), "%")), size=3,
            position = position_stack(vjust = 0.5)) + theme_test() +
 scale_color_manual(values=c('gray40', "gray80"))+ #'#b0b4b6', "#dadedf"   '#f5b85a', "#8baecf"
  #scale_color_manual(values=c('darkorange2', "dodgerblue4"))+
 scale_fill_manual(values=c('gray40', "gray80"))+ #'#b0b4b6', "#dadedf"   '#f5b85a', "#8baecf"
  labs(y="Read Distribution (%)", x="")+
theme(legend.title = element_blank()) +
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))+
  scale_y_discrete(expand = c(0,0)) + scale_x_discrete(expand = c(0,0))
```

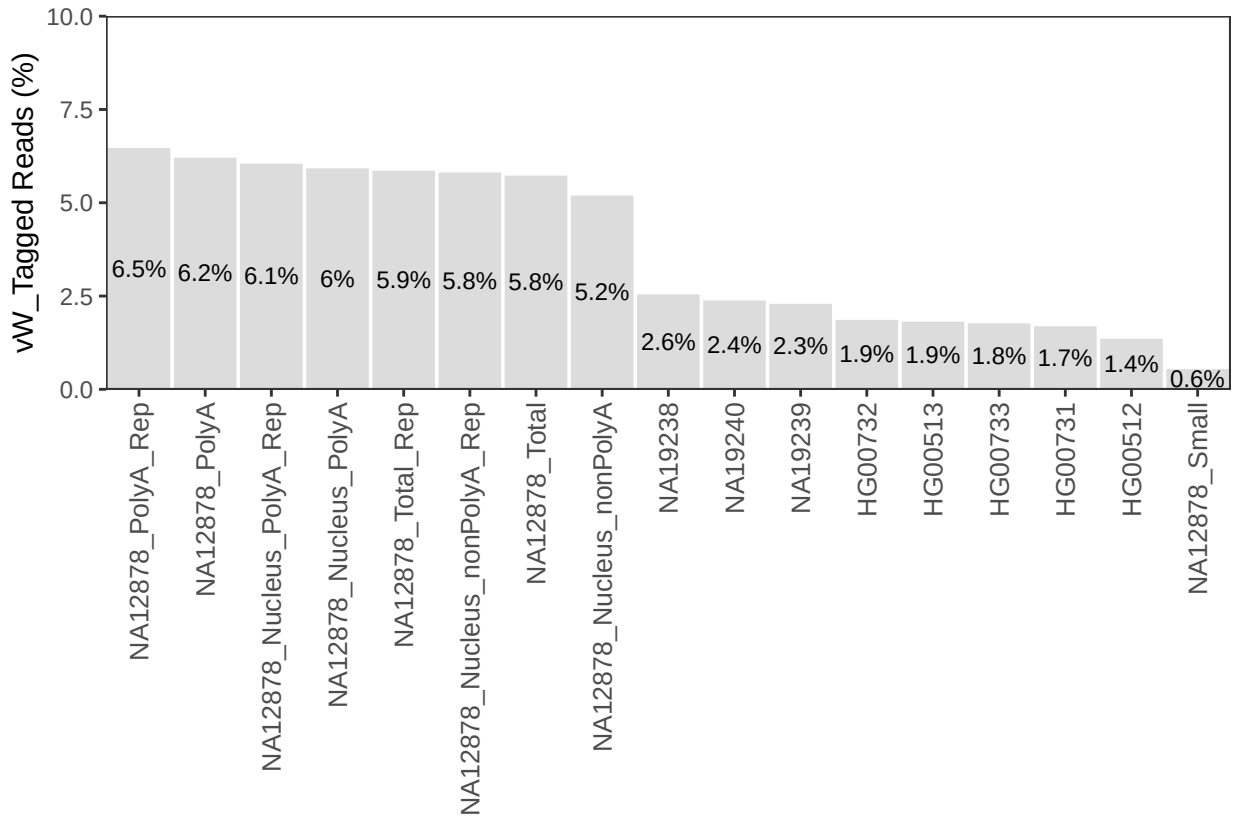


```
#ggsave("~/Desktop/dist.pdf", plot=plot1,width=10, height=6, device="pdf")

#Representation 2 - plotting only vW_Tagged read percentages
theme_asiimwe <- function(base_size = 12) {
  theme_bw(base_size = base_size) %+replace%
    theme(
      plot.title = element_text(size = rel(1), face = "bold", margin = margin(0,0,5,0), hjust = 0),
      panel.grid.minor = element_blank(),
      panel.border = element_blank(),
      axis.title = element_text(size = rel(0.85), face = "bold"),
      axis.text = element_text(size = rel(0.70), face = "bold"),
      axis.line = element_line(color = "black", arrow = arrow(length = unit(0.3, "lines"), type = "clos
      legend.title = element_text(size = rel(0.85), face = "bold"),
      legend.text = element_text(size = rel(0.70), face = "bold"),
      legend.key = element_rect(fill = "transparent", colour = NA),
      legend.key.size = unit(1.5, "lines"),
      legend.background = element_rect(fill = "transparent", colour = NA),
      strip.background = element_rect(fill = "#17252D", color = "#17252D"),
      strip.text = element_text(size = rel(0.85), face = "bold", color = "white", margin = margin(5,0,5
    )
}

# Overal dist reads with and without tags
num_reads_reshaped2
```

```
##                         Sample       perc              Flag
## 1                       HG00731  1.7279459   vW_Tagged Reads
## 2                  NA12878_Small  0.5798579   vW_Tagged Reads
## 3      NA12878_Nucleus_PolyA_Rep  6.0849228   vW_Tagged Reads
## 4                       HG00513  1.8516091   vW_Tagged Reads
## 5                       HG00512  1.3943094   vW_Tagged Reads
## 6   NA12878_Nucleus_nonPolyA_Rep  5.8472959   vW_Tagged Reads
## 7       NA12878_Nucleus_nonPolyA  5.2313759   vW_Tagged Reads
## 8          NA12878_Nucleus_PolyA  5.9606049   vW_Tagged Reads
## 9                  NA12878_Total  5.7643115   vW_Tagged Reads
## 10                      NA19238  2.5797105   vW_Tagged Reads
## 11                      HG00733  1.8069782   vW_Tagged Reads
## 12                 NA12878_PolyA  6.2438961   vW_Tagged Reads
## 13                      HG00732  1.8992709   vW_Tagged Reads
## 14             NA12878_PolyA_Rep  6.5065801   vW_Tagged Reads
## 15             NA12878_Total_Rep  5.8963314   vW_Tagged Reads
## 16                      NA19239  2.3300029   vW_Tagged Reads
## 17                      NA19240  2.4180752   vW_Tagged Reads
## 18                      HG00731 98.2720541 Reads with no Tag
## 19                 NA12878_Small 99.4201421 Reads with no Tag
## 20     NA12878_Nucleus_PolyA_Rep 93.9150772 Reads with no Tag
## 21                      HG00513 98.1483909 Reads with no Tag
## 22                      HG00512 98.6056906 Reads with no Tag
## 23  NA12878_Nucleus_nonPolyA_Rep 94.1527041 Reads with no Tag
## 24      NA12878_Nucleus_nonPolyA 94.7686241 Reads with no Tag
## 25         NA12878_Nucleus_PolyA 94.0393951 Reads with no Tag
## 26                 NA12878_Total 94.2356885 Reads with no Tag
## 27                      NA19238 97.4202895 Reads with no Tag
```

```
## 28                    HG00733 98.1930218 Reads with no Tag
## 29               NA12878_PolyA 93.7561039 Reads with no Tag
## 30                    HG00732 98.1007291 Reads with no Tag
## 31           NA12878_PolyA_Rep 93.4934199 Reads with no Tag
## 32           NA12878_Total_Rep 94.1036686 Reads with no Tag
## 33                    NA19239 97.6699971 Reads with no Tag
## 34                    NA19240 97.5819248 Reads with no Tag
```

```
num_reads_reshaped2[order(num_reads_reshaped2$Flag, decreasing = T),] %>%  filter(Flag == "vW_Tagged Rea
ggplot() +
  geom_bar(aes(x = reorder(Sample, -perc),
               y = perc,
               group = Sample,
               fill = factor(Flag, levels=c("vW_Tagged Reads","Reads with no Tag" ))),
           stat = "identity", width = 0.99, alpha=0.9, color= "white") +
  geom_text(aes(x = Sample,
                y = perc,
                label = paste0(round(perc,1), "%")), size=3,
            position = position_stack(vjust = 0.5)) + theme_test() +
 # scale_color_manual(values="gray50")+
    scale_fill_manual(values="gray85")+
  labs(y="vW_Tagged Reads (%)", x="")+
theme(legend.title = element_blank(), legend.position = "none") +
  theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10))+
 scale_y_continuous(expand = c(0,0), limits = c(0, 10)) + scale_x_discrete(expand = c(0,0))
```



```
#ggsave("~/Desktop/dist.pdf", plot=plot1,width=10, height=6, device="pdf")

colnames(num_reads_reshaped) <- c("Sample", "Num_Reads",       "vW_Tagged_Reads" ,"Percentage_Tagged" ,
#Table with percent distribution of reads with and without tags
knitr::kable(
  num_reads_reshaped, row.names = FALSE, #row.names = TRUE,
 # caption = '<b>Reads</b>', format = 'html'
)
```

Sample	Num_Reads	vW_Tagged_Reads	Percentage_Tagged	Percentage_Untagged
HG00731	56254714	972051	1.7279459	98.27205
NA12878_Small	221362695	1283589	0.5798579	99.42014
NA12878_Nucleus_PolyA_Rep	116517502	7090000	6.0849228	93.91508
HG00513	58601893	1085078	1.8516091	98.14839
NA12878_RAMPAGE	32796876	384173	1.1713707	98.82863
HG00512	66055710	921021	1.3943094	98.60569
NA12878_Nucleus_nonPolyA_Rep	106919251	6251885	5.8472959	94.15270
NA12878_RAMPAGE_Rep	36145096	355632	0.9839011	99.01610
NA12878_Nucleus_nonPolyA	110469791	5779090	5.2313759	94.76862
NA12878_Nucleus_PolyA	128402941	7653592	5.9606049	94.03940
NA12878_Total	105089150	6057666	5.7643115	94.23569
NA19238	63949386	1649709	2.5797105	97.42029
HG00733	92075712	1663788	1.8069782	98.19302
NA12878_PolyA	97548052	6090799	6.2438961	93.75610
HG00732	70029452	1330049	1.8992709	98.10073
NA12878_PolyA_Rep	93555584	6087269	6.5065801	93.49342
NA12878_Total_Rep	92494632	5453790	5.8963314	94.10367
NA19239	72457249	1688256	2.3300029	97.67000
NA19240	59219085	1431962	2.4180752	97.58192

4.2 Sample-specific vW_Tag Distributions for alignments that did not pass WASP filtering

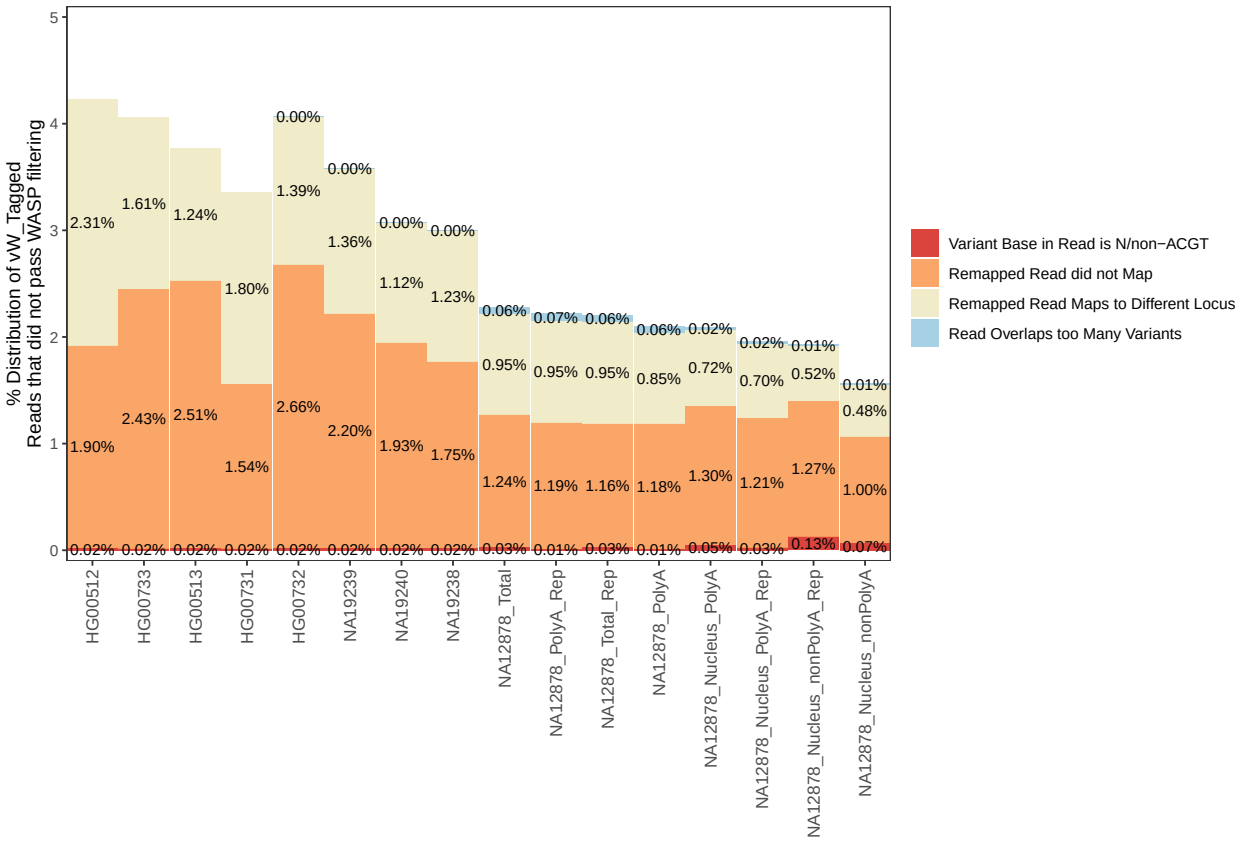
```
#revist expunge TD
samples <- c("HG00512","HG00513","HG00731","HG00732","HG00733","NA12878_Nucleus_nonPolyA","NA12878_Nucl
"NA12878_Nucleus_PolyA", "NA12878_Nucleus_PolyA_Rep", "NA12878_PolyA","NA12878_PolyA_Rep","NA12878_Total

files_list <- list()
for(i in samples){
#path <- paste0("/home/asiimwe/Desktop/TD/pass2_to_keep_", i , ".txt")
path <- paste0("/home/asiimwe/projects/run_env/alpha_star_wasp_benchmarking/WASP/WASP_Runs/", i, "/32th
files_list[[i]] <- path
}
files = do.call(rbind, files_list)

# Reading and merging data from all samples
merger <- lapply(files, function(x) {
  merger <- read.table(x, comment.char = "", sep = '', header = FALSE, stringsAsFactors = FALSE)
  merger$source <- x
  return(merger)
}) %>%
bind_rows()
```

```
merger <- as.data.frame(merger)
merger %>% head(n=2)
merger$Sample <- merger$source
merger$source <- NULL
merger$Sample <- gsub("/home/asiimwe/projects/run_env/alpha_star_wasp_benchmarking/WASP/WASP_Runs/", ""
merger$Sample <- gsub("/32threads/pass2_to_keep.txt", "", merger$Sample)
merger %>% head(n=2)
unique(merger$Sample)#validate called samples

merger <- as.data.frame(merger)
colnames(merger) <- c("Read_ID", "vW_Tag", "Reads_to_remap", "Reads_to_keep", "Sample")
merger %>% head(n=2)

merger <- cSplit(merger, "vW_Tag", ":", direction = "wide", type.convert="as.character")
merger$vW_Tag_1 <- NULL
merger$vW_Tag_2 <- NULL
colnames(merger)[5] <- "vW_Tag"
merger %>% head(n=2)

merger$vW_Tag <- as.integer(as.character(merger$vW_Tag))
merger <- as.data.frame(merger)
write.csv(merger, file="/home/asiimwe/projects/run_env/alpha_star_wasp_benchmarking/Downstream_Analysis/

#Creating dataframe with mutated percentages per tag for plotting
perclist <- list()

for (i in unique(merger$Sample)){
  #print(i)
  perc <- merger %>% filter(Sample == i)
  perc <- as.data.frame(perc %>%
  group_by(vW_Tag) %>%
  dplyr::summarise(cnt = n()) %>%
  mutate(freq = formattable::percent(cnt / sum(cnt))))

  perc$Sample <- i
  perclist[[i]] <- perc
}

vW_Tag_percentages = do.call(rbind, perclist)

merger_vW_Tag_percentages <- as.data.frame(vW_Tag_percentages)
orders <- as.data.frame(merger_vW_Tag_percentages %>% filter(vW_Tag != 1) %>%
                          group_by(Sample) %>%
                          dplyr::summarise(sum(cnt)))

colnames(orders)[2] <- "order"
vW_Tag_percentages <- inner_join(merger_vW_Tag_percentages, orders, by = "Sample")

# Plotting
unique(vW_Tag_percentages$vW_Tag) #1 3 4 6 7
colourCount = length(unique(vW_Tag_percentages$vW_Tag))
getPalette = colorRampPalette(brewer.pal(8,"RdYlBu"))#RdYlBu Spectral Greys
```

```
vW_Tag_percentages$vW_Tag <- as.factor(vW_Tag_percentages$vW_Tag)

vW_Tag_percentages$Freq <- vW_Tag_percentages$freq
vW_Tag_percentages$Freq <- gsub("%", "", vW_Tag_percentages$Freq)
vW_Tag_percentages$Freq <- as.numeric(as.character(vW_Tag_percentages$Freq))

#Sneak pick into distributions of tags overall
vW_Tag_percentages %>% group_by(vW_Tag) %>% dplyr::summarise(sum(cnt))
# A tibble: 5 × 2
#  vW_Tag `sum(cnt)`
#  <fct>       <int>
#1 1        97368800
#2 3           40555
#3 4          864141
#4 6          571383
#5 7           34311

#https://htmlcolorcodes.com/
#"#D73027" - red var
#"#FA9D59" -orange var
#"#EFE9C4" - light or
#"#9DCDE3" -blue
#"#4575B4" - dark blue

vW_Tag_percentages$vW_Tag_desc <- vW_Tag_percentages$vW_Tag
vW_Tag_percentages$vW_Tag_desc <- as.character(vW_Tag_percentages$vW_Tag_desc)
vW_Tag_percentages$vW_Tag_desc[vW_Tag_percentages$vW_Tag_desc == "1"] <- "Alignment Passed WASP Filteri
vW_Tag_percentages$vW_Tag_desc[vW_Tag_percentages$vW_Tag_desc == "3"] <- "Variant Base in Read is N/non
vW_Tag_percentages$vW_Tag_desc[vW_Tag_percentages$vW_Tag_desc == "4"] <- "Remapped Read did not Map"
vW_Tag_percentages$vW_Tag_desc[vW_Tag_percentages$vW_Tag_desc == "6"] <- "Remapped Read Maps to Differe
vW_Tag_percentages$vW_Tag_desc[vW_Tag_percentages$vW_Tag_desc == "7"] <- "Read Overlaps too Many Varian

write.csv(vW_Tag_percentages, file="/home/asiimwe/projects/run_env/alpha_star_wasp_benchmarking/Downstr
```

```
vW_Tag_percentages <- read.csv(file="/home/asiimwe/projects/run_env/alpha_star_wasp_benchmarking/Downsti

(plot <- vW_Tag_percentages %>% filter(vW_Tag != 1) %>%
    #mutate(Sample = fct_reorder(Sample, order)) %>%
    ggplot() +
    geom_bar(aes(x = reorder(Sample,
                -Freq), Freq,
                group = Sample,
                fill = vW_Tag_desc),
             stat = "identity", width = 0.99, alpha=0.9) +
    geom_text(aes(x = Sample,
                  y =Freq,
                  label = freq), size = 3,
              position = position_stack(vjust = 0.5)
              ) +
    #scale_fill_manual(values = getPalette(colourCount))+
    #scale_fill_manual(values = c("#9DCDE3", "#D73027", "#4575B4", "#FA9D59", "#EFE9C4")) +
    scale_fill_manual(values =c("Variant Base in Read is N/non-ACGT" = "#D73027",
                                "Remapped Read did not Map" = "#FA9D59",
```

```
                                "Remapped Read Maps to Different Locus" ="#EFE9C4",
                                "Read Overlaps too Many Variants"="#9DCDE3"))+
    theme_test()+
    labs(y=paste0("% Distribution of vW_Tagged", "\n", "Reads that did not pass WASP filtering"), x="")+
    theme(legend.title = element_blank(), legend.position="right")  + #legend.position = c(0.8, 0.65)
    theme(axis.text.x = element_text(angle = 90,hjust = 1,vjust = 0.5, size=10)) +
    scale_y_continuous(expand = c(0.02,0), limits = c(0, 5)) +
    scale_x_discrete(expand = c(0,0)))# + coord_flip()
```



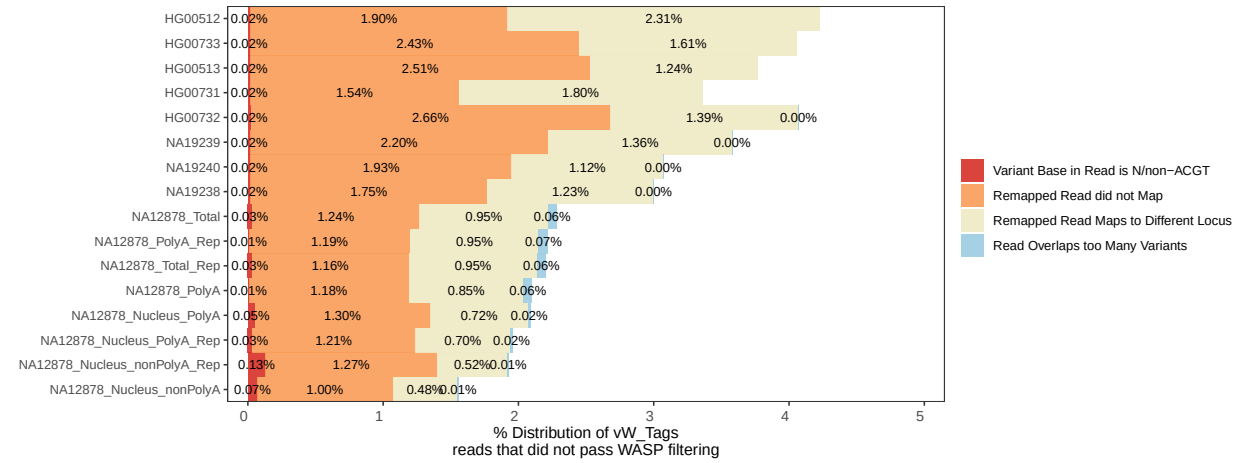

```
#ggsave("~/Desktop/merger/plot_nocolor.pdf", plot=plot,width=10, height=15, device="pdf")
#ggsave("~/Desktop/merger/plot_color.pdf", plot=plot,width=10, height=15, device="pdf")#output with col
```

```
#Flipped
unique(vW_Tag_percentages$vW_Tag_desc)
```

```
## [1] Alignment Passed WASP Filtering       Variant Base in Read is N/non-ACGT
## [3] Remapped Read did not Map             Remapped Read Maps to Different Locus
## [5] Read Overlaps too Many Variants
## 5 Levels: Alignment Passed WASP Filtering ... Variant Base in Read is N/non-ACGT
```

```
(plot <- vW_Tag_percentages %>% filter(vW_Tag != 1) %>%
    ggplot() +
    geom_bar(aes(x = reorder(Sample,
                  Freq), Freq,
```

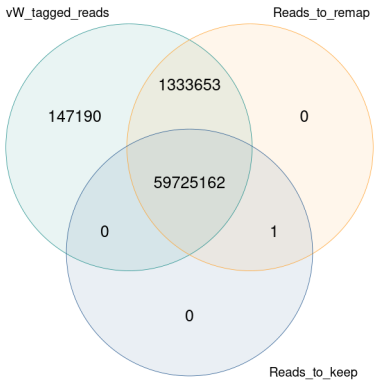
```
                group = Sample,
                fill = vW_Tag_desc),
            stat = "identity", width = 0.99, alpha = 0.9) +
   geom_text(aes(x = Sample,
                 y =Freq,
                 label = freq), size = 3,
             position = position_stack(vjust = 0.5)
             ) +
scale_fill_manual(values =c("Variant Base in Read is N/non-ACGT" = "#D73027",
                            "Remapped Read did not Map" = "#FA9D59",
                            "Remapped Read Maps to Different Locus" ="#EFE9C4",
                            "Read Overlaps too Many Variants"="#9DCDE3"))+
    theme_test()+
    labs(y=paste0("% Distribution of vW_Tags", "\n", "reads that did not pass WASP filtering"), x="") +
    theme(legend.title = element_blank(), legend.position = "right")  + #, legend.position = "bottom"
    theme(axis.text.x = element_text(angle = 0,hjust = 1,vjust = 0.5, size=10)) +
    scale_y_continuous(expand = c(0.03,0), limits = c(0, 5)) +
    scale_x_discrete(expand = c(0,0))) + coord_flip()
```



Venn Diagrams - Overall

```
merger$to_keep_recoded <- merger$Reads_to_keep
merger$to_remap_recoded <- merger$Reads_to_remap

sum(is.na(merger$to_remap))#147190
sum(is.na(merger$to_keep))#1480843

merger$to_remap_recoded <- as.character(merger$to_remap_recoded)
merger$to_keep_recoded <- as.character(merger$to_keep_recoded)

merger$to_remap_recoded[is.na(merger$to_remap_recoded)] <- 0
merger$to_remap_recoded[merger$to_remap_recoded != 0] <- 1
merger$to_keep_recoded[is.na(merger$to_keep_recoded)] <- 0
merger$to_keep_recoded[merger$to_keep_recoded != 0] <- 1
head(merger)
```

```
merger$vW_tag <- as.integer(merger$vW_Tag)
merger$to_remap_recoded <- as.integer(merger$to_remap_recoded)
merger$to_keep_recoded <- as.integer(merger$to_keep_recoded)

#Creating comparative read sets for venn diagram
merger$Reads_to_remap <- as.character(merger$Reads_to_remap)
merger$Reads_to_keep <- as.character(merger$Reads_to_keep)

merger$Reads_to_remap[is.na(merger$Reads_to_remap)] <- "0" #Venn diagrams don't work well with NAs so w
merger$Reads_to_keep[is.na(merger$Reads_to_keep)] <- "0"
str(merger)

# merger_HG00512 <- merger %>% filter(Sample == "HG00512")
# set1 <- as.character(merger_HG00512$Read_ID)
# set2 <-  as.character(merger_HG00512$Reads_to_remap)
# set3 <-  as.character(merger_HG00512$Reads_to_keep)

set1 <- as.character(merger$Read_ID)
set2 <-  as.character(merger$Reads_to_remap)
set3 <-  as.character(merger$Reads_to_keep)

myCol <- brewer.pal(3, "Pastel2")

venn.diagram(
  x = list(set1, set2, set3),
  category.names = c("vW_tagged_reads" , "Reads_to_remap" , "Reads_to_keep"),
  filename = '/home/asiimwe/projects/run_env/alpha_star_wasp_benchmarking/Downstream_Analysis/Read_Overl
  output = TRUE ,
  imagetype="png" ,
  height = 600 ,
  width = 600 ,
  resolution = 300,
  compression = "lzw",
  lwd = 0.25,
  col=c('#21908dff', '#ffa533', "#2a5c92"),
  fill = c(alpha("#21908dff",0.2), alpha('#ffa533',0.2), alpha('#2a5c92',0.2)),
  cex = 0.5,
  fontfamily = "sans",
  cat.cex = 0.4,
  cat.default.pos = "outer",
  cat.pos = c(-27, 27, 135),
  cat.dist = c(0.055, 0.055, 0.085),
  cat.fontfamily = "sans",
  #cat.col = c("#21908dff", '#ffa533', '#ffa533'),
  cat.col = "black",
  rotation = 1
)
```



Per sample

```
for(i in sample){
merger_sample <- merger %>% filter(Sample == i)
merger_sample <- as.character(merger_sample$Read_ID)
merger_sample <-  as.character(merger_sample$Reads_to_remap)
merger_sample <-  as.character(merger_sample$Reads_to_keep)

venn.diagram(
  x = list(set1_HG00512, set2_HG00512, set3_HG00512),
  category.names = c("vW_tagged_reads" , "Reads_to_remap" , "Reads_to_keep"),
  filename = paste0('/home/asiimwe/projects/run_env/alpha_star_wasp_benchmarking/Downstream_Analysis/Ver
  output = TRUE ,
  imagetype="png" ,
  height = 600 ,
  width = 600 ,
  resolution = 300,
  compression = "lzw",
  lwd = 0.25,
  col=c('#21908dff', '#ffa533', "#2a5c92"),
  fill = c(alpha("#21908dff",0.2), alpha('#ffa533',0.2), alpha('#2a5c92',0.2)),
  cex = 0.5,
  fontfamily = "sans",
  cat.cex = 0.4,
  cat.default.pos = "outer",
  cat.pos = c(-27, 27, 135),
  cat.dist = c(0.055, 0.055, 0.085),
  cat.fontfamily = "sans",
  #cat.col = c("#21908dff", '#ffa533', '#ffa533'),
  cat.col = "black",
  rotation = 1
)}
```


vW_tagged_reads
Reads_to_remap
38753
60
0
882208
0
1
0
Reads_to_keep