Graphical Representation of Percent Aligned Depending On Genome Used

Load required libraries

```r
library(plyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(tibble)
library(stringr)
library(biomaRt)
library(genefilter)
library(DESeq2)
```

```
## Loading required package: S4Vectors

## Loading required package: stats4

## Loading required package: BiocGenerics

## Loading required package: parallel

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB

## The following objects are masked from 'package:dplyr':
##
##     combine, intersect, setdiff, union

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind,
##     colMeans, colnames, colSums, dirname, do.call, duplicated,
##     eval, evalq, Filter, Find, get, grep, grepl, intersect,
##     is.unsorted, lapply, lengths, Map, mapply, match, mget, order,
##     paste, pmax, pmax.int, pmin, pmin.int, Position, rank, rbind,
##     Reduce, rowMeans, rownames, rowSums, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which, which.max,
##     which.min

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:dplyr':
##
##     first, rename

## The following object is masked from 'package:plyr':
##
##     rename

## The following object is masked from 'package:base':
##
##     expand.grid

## Loading required package: IRanges

##
## Attaching package: 'IRanges'

## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice

## The following object is masked from 'package:plyr':
##
##     desc

## Loading required package: GenomicRanges

## Loading required package: GenomeInfoDb

## Loading required package: SummarizedExperiment

## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

## Loading required package: DelayedArray

## Loading required package: matrixStats

##
## Attaching package: 'matrixStats'

## The following objects are masked from 'package:Biobase':
##
##     anyMissing, rowMedians

## The following objects are masked from 'package:genefilter':
##
```

```
##     rowSds, rowVars

## The following object is masked from 'package:dplyr':
##
##     count

## The following object is masked from 'package:plyr':
##
##     count

## Loading required package: BiocParallel

##
## Attaching package: 'DelayedArray'

## The following objects are masked from 'package:matrixStats':
##
##     colMaxs, colMins, colRanges, rowMaxs, rowMins, rowRanges

## The following objects are masked from 'package:base':
##
##     aperm, apply
```

```r
library(gplots)
```

```
##
## Attaching package: 'gplots'

## The following object is masked from 'package:IRanges':
##
##     space

## The following object is masked from 'package:S4Vectors':
##
##     space

## The following object is masked from 'package:stats':
##
##     lowess
```

```r
library(ggplot2)
library(RColorBrewer)
library(stringr)
library(devtools)
library(reshape2)
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:reshape2':
##
##     dcast, melt

## The following object is masked from 'package:SummarizedExperiment':
##
##     shift

## The following object is masked from 'package:GenomicRanges':
##
##     shift
## The following object is masked from 'package:IRanges':
##
##     shift
```

```
## The following objects are masked from 'package:S4Vectors':
##
##     first, second
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```r
library(purrr)
```

```
##
## Attaching package: 'purrr'
```

```
## The following object is masked from 'package:data.table':
##
##     transpose
```

```
## The following object is masked from 'package:DelayedArray':
##
##     simplify
```

```
## The following object is masked from 'package:GenomicRanges':
##
##     reduce
```

```
## The following object is masked from 'package:IRanges':
##
##     reduce
```

```
## The following object is masked from 'package:plyr':
##
##     compact
```

```r
library(forcats)
```

Purpose

To graphically present the percent mapped and percent aligned of reads when aligning the reads to the available genome of a given species versus aligning all species' reads to the human genome.

```r
fileName_NHP <- file.path("MultiQC_Reports_Species_Genomes/Stats")
NHP_files <- basename(Sys.glob(file.path(fileName_NHP, "*.tabular")))

fileName_human <- file.path("MultiQC_reports_human_genome/Stats")
human_files <- basename(Sys.glob(file.path(fileName_human, "*.tabular")))

reader <- function(files) {
  d <- read.delim(files)
  d
}

human_files_read <- lapply(file.path(fileName_human, human_files), reader)
names(human_files_read) <- sub('.tabular', '', human_files)
```

```
NHP_files_read <- lapply(file.path(fileName_NHP, NHP_files), reader)
names(NHP_files_read) <- sub('.tabular', '', NHP_files)

##To get our bearings of what columns are what
colnames(human_files_read[[1]])
```

```
##  [1] "Sample"
##  [2] "featureCounts_mqc.generalstats.featurecounts.percent_assigned"
##  [3] "featureCounts_mqc.generalstats.featurecounts.Assigned"
##  [4] "STAR_mqc.generalstats.star.uniquely_mapped_percent"
##  [5] "STAR_mqc.generalstats.star.uniquely_mapped"
##  [6] "FastQC_mqc.generalstats.fastqc.percent_duplicates"
##  [7] "FastQC_mqc.generalstats.fastqc.percent_gc"
##  [8] "FastQC_mqc.generalstats.fastqc.avg_sequence_length"
##  [9] "FastQC_mqc.generalstats.fastqc.percent_fails"
## [10] "FastQC_mqc.generalstats.fastqc.total_sequences"
```

```
##We want the sample name, the percent assigned, and the percent uniquely mapped
NHP_files_select <- lapply(NHP_files_read, "[", c(1, 2, 4))
human_files_select <- lapply(human_files_read, "[", c(1, 2, 4))

##Now we need to simplify the sample names of each row
##Important note: The original sample names for gorilla, bonobo have a shortened donor ID --
##they say just "320" instead of "PR230". These were changed in the files after download from
##Galaxy to make the simplification of donor names easier. The squirrel monkey sample names were also
##adjusted -- the samples prefaced with "7_Barcode_Splitter_on_data_13_data_14_and_data_3_A"
##were changed to read "data_3_SQMA" at the end;
##similarly "7_Barcode_Splitter_on_data_13_data_14_and_data_3_B"
##were changed to read "data_3_SQMB" at the end. The samples prefaced with
##"11_Barcode_Splitter_on_data_11_data_12_and_data_8_A_M" were changed to read "data_8_AG05311A_M"
##at the end. There was a typo in two lines of the pigtailed macaque where "PR00058" was missing the
##leading "P".
info_extract <- function(input) {
  input$donor <- str_extract(input$Sample, "PR\\d*|AG\\d*|S\\d{4,}|SQM\\w{1}|NHDF|AF|SR|C57\\w")
  input$treatment <- ifelse(grepl("mock|*M\\d|*M\\d\\d", input$Sample), "mock", "treated")
  input$replicate = ifelse(grepl("_A_|24A|mockA|treatedA|M01|M1|T1|T01", input$Sample), "A",
                  ifelse(grepl("_B_|24B|mockB|treatedB|M02|M2|T2|T02", input$Sample), "B", "C"))
  input$Sample <- sub("\\-library.*|\\_fastqsanger.*", "", input$Sample)
  input
  }

human_select_labeled <- lapply(human_files_select, info_extract)
NHP_select_labeled <- lapply(NHP_files_select, info_extract)

column_cleaning <- function(data){
  part_1 <- data[!is.na(data[,2]), 1:2]
  part_2 <- data[!is.na(data[,3]), c(1,3:6)]
  complete <- full_join(part_1, part_2, by = "Sample")
  complete
}

human_cleaned <- lapply(human_select_labeled, column_cleaning)
NHP_cleaned <- lapply(NHP_select_labeled, column_cleaning)
```

```
human_cleaned_df <- do.call(rbind, human_cleaned)

NHP_cleaned_df <- do.call(rbind, NHP_cleaned)

all_df <- rbind(human_cleaned_df, NHP_cleaned_df)
all_df$GenomeMapped <- ifelse(grepl("_human_genome",
                                    rownames(all_df), ignore.case = TRUE), "human", "species")
all_df$Origin <- ifelse(grepl("AG07923|AG08490|PR0058", all_df$donor), "pigtailed macaque",
                  ifelse(grepl("PR00033|PR00036|PR00039", all_df$donor), "olive baboon",
                  ifelse(grepl("AG05311|SQMA|SQMB", all_df$donor), "squirrel monkey",
                  ifelse(grepl("AG06105|PR00054|PR01109", all_df$donor), "orangutan",
                  ifelse(grepl("PR230|PR573|PR107", all_df$donor), "gorilla",
                  ifelse(grepl("PR111|PR235|PR248", all_df$donor), "bonobo",
                  ifelse(grepl("S4933|S3611|S3649", all_df$donor), "chimpanzee",
                  ifelse(grepl("AG08308|AG08312|AG08305", all_df$donor), "rhesus macaque",
                  ifelse(grepl("NHDF|AF|SR", all_df$donor), "human",
                         ifelse(grepl("C57", all_df$donor), "mouse",
                         "nothing"))))))))))
colnames(all_df)[2] <- c("Percent Assigned")
colnames(all_df)[3] <- c("Percent Aligned")

##double checking there are no missing values
anyNA(all_df)
```

```
## [1] FALSE
```

```
all_df$Name <- paste(all_df$donor, all_df$treatment, all_df$replicate)
##Melting for graphing
all_m <- melt(all_df)
```

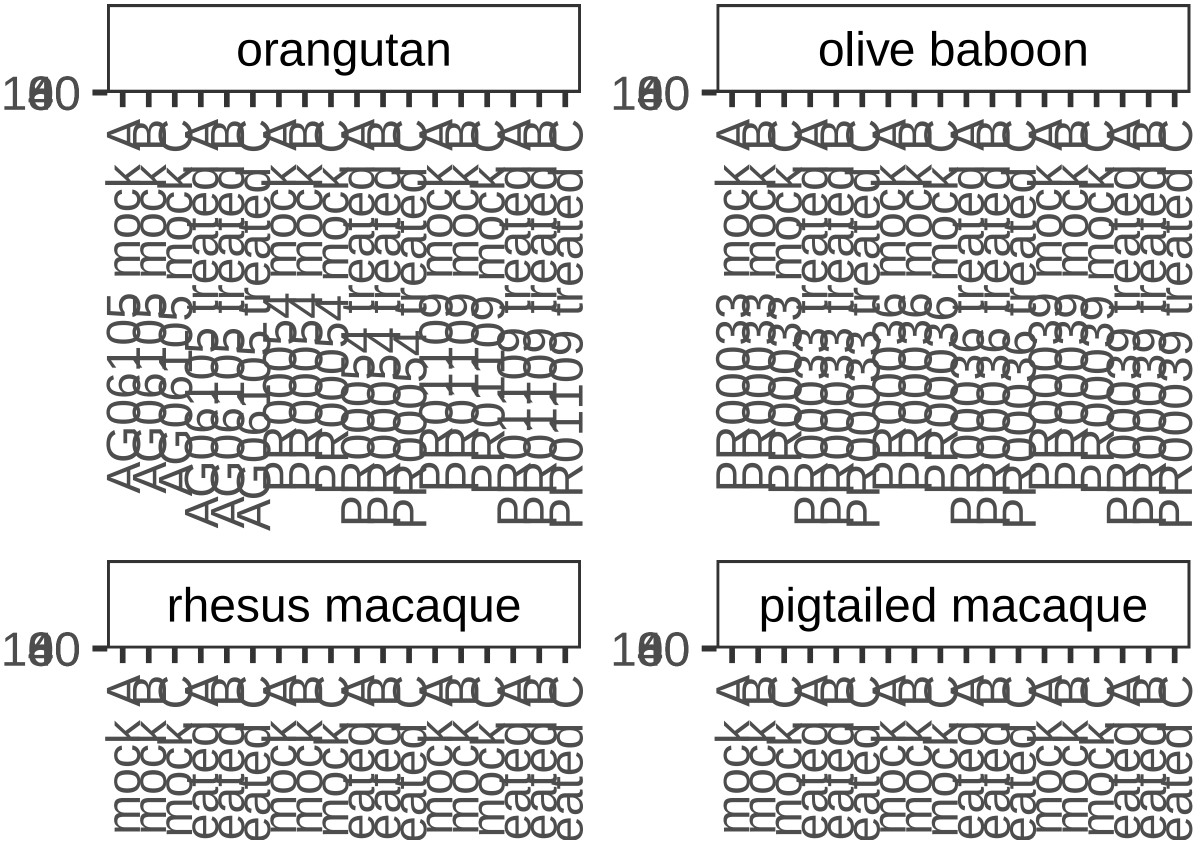
```
## Using Sample, donor, treatment, replicate, GenomeMapped, Origin, Name as id variables
```

```
all_m$Complete <- paste(all_m$GenomeMapped, all_m$variable)

all_m$Complete <- factor(all_m$Complete, levels = c("human Percent Aligned", "species Percent Aligned",
                                                 "human Percent Assigned", "species Percent Assigned"))
all_m$Origin <- factor(all_m$Origin, levels = c("human", "chimpanzee", "bonobo", "gorilla", "orangutan"
                                                "olive baboon", "rhesus macaque", "pigtailed macaque",
                                                "squirrel monkey", "mouse"))

write.csv(all_m, "PercentAssigned_PercentMapped_InputData.csv")

##Plot
plot <- ggplot(all_m, aes(fill = Complete, x = Name, y = value)) +
  geom_bar(stat="identity", position = "dodge") +
  facet_wrap(~Origin, scales = "free", ncol = 2) +
  scale_fill_manual(values = c("#ca0020", "#f4a582", "#0571b0", "#92c5de")) +
  theme_bw(base_size = 22) + coord_cartesian(ylim = c(40, 100)) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5), panel.grid.major = element_line
        panel.grid.minor = element_line("black"), axis.title.y = element_blank(), axis.title.x = elemen
        legend.title = element_blank(), legend.position = 'bottom') +
  theme(strip.background =element_rect(fill="white")) +
  theme(strip.text = element_text(colour = "black"))
ggsave(file = paste(Sys.Date(), "PercentAssigned_PercentMapped.pdf"), plot = plot,
       height = 20, width = 20, device = "pdf")
print(plot)
```



```
sessionInfo()
```

```
## R version 3.5.2 (2018-12-20)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Sierra 10.12.6
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] parallel  stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] forcats_0.4.0               purrr_0.2.5
##  [3] data.table_1.12.0           reshape2_1.4.3
##  [5] usethis_1.4.0               devtools_2.0.1
##  [7] RColorBrewer_1.1-2          ggplot2_3.1.0
##  [9] gplots_3.0.1                DESeq2_1.22.2
## [11] SummarizedExperiment_1.12.0 DelayedArray_0.8.0
## [13] BiocParallel_1.16.5        matrixStats_0.54.0
## [15] Biobase_2.42.0             GenomicRanges_1.34.0
## [17] GenomeInfoDb_1.18.1        IRanges_2.16.0
## [19] S4Vectors_0.20.1           BiocGenerics_0.28.0
## [21] genefilter_1.64.0          biomaRt_2.38.0
## [23] stringr_1.3.1              tibble_2.0.1
## [25] dplyr_0.7.8                plyr_1.8.4
##
## loaded via a namespace (and not attached):
##  [1] fs_1.2.6               bitops_1.0-6           bit64_0.9-7
##  [4] progress_1.2.0         httr_1.4.0             rprojroot_1.3-2
##  [7] tools_3.5.2            backports_1.1.3        R6_2.3.0
## [10] rpart_4.1-13           KernSmooth_2.23-15     Hmisc_4.1-1
## [13] DBI_1.0.0              lazyeval_0.2.1         colorspace_1.4-0
## [16] nnet_7.3-12            withr_2.1.2            processx_3.2.1
## [19] tidyselect_0.2.5       gridExtra_2.3          prettyunits_1.0.2
## [22] bit_1.1-14             compiler_3.5.2         cli_1.0.1
## [25] htmlTable_1.13.1       desc_1.2.0             labeling_0.3
## [28] caTools_1.17.1.1       scales_1.0.0           checkmate_1.9.1
## [31] callr_3.1.1            digest_0.6.18          foreign_0.8-71
## [34] rmarkdown_1.11         XVector_0.22.0         base64enc_0.1-3
## [37] pkgconfig_2.0.2        htmltools_0.3.6        sessioninfo_1.1.1
## [40] htmlwidgets_1.3        rlang_0.3.1            rstudioapi_0.9.0
## [43] RSQLite_2.1.1          bindr_0.1.1            gtools_3.8.1
## [46] acepack_1.4.1          RCurl_1.95-4.11        magrittr_1.5
## [49] GenomeInfoDbData_1.2.0 Formula_1.2-3          Matrix_1.2-15
## [52] Rcpp_1.0.0             munsell_0.5.0          stringi_1.2.4
## [55] yaml_2.2.0             zlibbioc_1.28.0        pkgbuild_1.0.2
## [58] grid_3.5.2             blob_1.1.1             gdata_2.18.0
## [61] crayon_1.3.4           lattice_0.20-38        splines_3.5.2
## [64] annotate_1.60.0        hms_0.4.2              locfit_1.5-9.1
## [67] ps_1.3.0               knitr_1.21             pillar_1.3.1
## [70] pkgload_1.0.2          geneplotter_1.60.0     XML_3.98-1.16
## [73] glue_1.3.0             evaluate_0.12          latticeExtra_0.6-28
## [76] remotes_2.0.2          gtable_0.2.0           assertthat_0.2.0
## [79] xfun_0.4               xtable_1.8-3           survival_2.43-3
## [82] AnnotationDbi_1.44.0   memoise_1.1.0          bindrcpp_0.2.2
## [85] cluster_2.0.7-1
```