Ploting Gene interruptions in R

IM

June 14, 2018

Visualization of Gene interruption. Example of Acinetobacter baumannii gene ComA

This scrips help to visualize genes interrupted in several strains. It imports different type of data: 1. Blast output 2. Sequence annotations 3. IS blast outpout.

Finally, it also make some sequence statistics as sequence length, SNPS, Synonimous /Non-synonimous modifications.

Add libraries

```
library("ggplot2")
library(PopGenome)
options(digits = 2)
```

Import the annotation of the sequence

This part import the output of the gene prediction on the sequences

```
gff <- read.table("~/Documents/EPFL/A_baumannii/Reference_sequences/ALL_Abaum_vf/ALL_Analysis_V2/db_gen
    sep = "\t")
# modify GFF names
strain <- lapply(strsplit(as.character(gff$V1), "_"), function(x) x[length(x)])
gff$V1 <- unlist(strain)
gff$V1 <- gsub(" ", "", gff$V1)
order_comA <- gsub(" ", "", gff$V1)

# head of annotation file
head(gff)
```

```
##           V1           V2  V3 V4   V5    V6 V7 V8
## 1      15A34 GeneMark.hmm CDS  1 2382 -3100  -  0
## 2       15A5 GeneMark.hmm CDS  1 2382 -3100  +  0
## 3     1656-2 GeneMark.hmm CDS  1 2382 -3100  -  0
## 4 2004BJAB14 GeneMark.hmm CDS  1 2382 -3100  -  0
## 5  2004ZJAB5 GeneMark.hmm CDS  1 2382 -3096  -  0
## 6  2004ZJAB6 GeneMark.hmm CDS  1 2382 -3096  +  0
##
## 1 gene_id=1, length=2382, gene_score=-3099.947127, rbs_score=0.033333, rbs_spacer=-1, stop_enforced=
## 2 gene_id=2, length=2382, gene_score=-3099.959756, rbs_score=0.033333, rbs_spacer=-1, stop_enforced=
## 3 gene_id=3, length=2382, gene_score=-3099.947127, rbs_score=0.033333, rbs_spacer=-1, stop_enforced=
## 4 gene_id=4, length=2382, gene_score=-3099.947127, rbs_score=0.033333, rbs_spacer=-1, stop_enforced=
## 5 gene_id=5, length=2382, gene_score=-3095.945666, rbs_score=0.033333, rbs_spacer=-1, stop_enforced=
## 6 gene_id=6, length=2382, gene_score=-3095.958310, rbs_score=0.033333, rbs_spacer=-1, stop_enforced=
```

Import info from IS elements in the sequence

We used the online IS elements database, to blast the Coding regions in the sequences and then imported the tabular output of this result.

```
MGE <- read.table("~/Documents/EPFL/A_baumannii/Reference_sequences/ALL_Abaum_vf/ALL_Analysis_V2/db_gen
    sep = "\t")
head(MGE)
```

```
##                           V1       V2 V3   V4 V5 V6   V7   V8   V9  V10
## 1    Seq_ACIAD2639_comA_6200 ISAba125 99 1087  8  0 1671 2757    1 1087
## 2    Seq_ACIAD2639_comA_6200  ISC1041 90  644 37 16 2120 2748  404 1037
## 3    Seq_ACIAD2639_comA_6200  ISC1041 96  317  5  7 1679 1989    1  314
## 4 Seq_ACIAD2639_comA_NCGM237 ISAba125 99 1087  9  0  757 1843 1087    1
## 5 Seq_ACIAD2639_comA_NCGM237  ISC1041 91  644 31 16  766 1394 1037  404
## 6 Seq_ACIAD2639_comA_NCGM237  ISC1041 95  317  8  7 1525 1835  314    1
##      V11  V12
## 1  0e+00 2091
## 2  0e+00  676
## 3 1e-130  466
## 4  0e+00 2083
## 5  0e+00  724
## 6 2e-123  442
```

```
# modify names
strain <- lapply(strsplit(as.character(MGE$V1), "_"), function(x) x[length(x)])
MGE$V1 <- unlist(strain)

# discard samples with IS > 1M MGE<-MGE[MGE$V1!='XH859',]
# MGE<-MGE[MGE$V1!='AC12',]
MGE <- MGE[, c(1, 2, 3, 7, 8, 4, 9, 10, 11)]
colnames(MGE) <- c("V1", "V2", "V3", "V4", "V5", "V6", "V7", "V8",
    "V9")

# head of data frame containing IS element info
head(MGE)
```

```
##        V1       V2 V3   V4   V5   V6   V7   V8     V9
## 1    6200 ISAba125 99 1671 2757 1087    1 1087  0e+00
## 2    6200  ISC1041 90 2120 2748  644  404 1037  0e+00
## 3    6200  ISC1041 96 1679 1989  317    1  314 1e-130
## 4 NCGM237 ISAba125 99  757 1843 1087 1087    1  0e+00
## 5 NCGM237  ISC1041 91  766 1394  644 1037  404  0e+00
## 6 NCGM237  ISC1041 95 1525 1835  317  314    1 2e-123
```

Merge the two data frames. Annotation + IS elements

```
ALL2 <- rbind.data.frame(gff, MGE)
```

Select only strains with interruption (selection by name)

```
ALL2 <- ALL2[with(ALL2, order(V1)), ]
ALL2 <- ALL2[ALL2$V1 %in% c("SDF", "6200", "AB0057", "WKA02", "2011ZJAB4",
```

```
    "NCGM237", "KAB05", "AbA118", "ATCC17978Yale", "ATCC19606"), ]

summary(ALL2)
```

```
##       V1                          V2        V3           V4
##  Length:30          GeneMark.hmm:23   CDS :23   Min.   :   1
##  Class :character   ISAba125    : 2   NA's: 7   1st Qu.:   1
##  Mode  :character   ISAba6      : 1             Median : 760
##                     ISC1041     : 4             Mean   : 936
##                                                 3rd Qu.:1677
##                                                 Max.   :2747
##
##        V5             V6           V7           V8
##  Min.   : 399   Min.   :-3103   -   :11   Min.   :   0
##  1st Qu.:1394   1st Qu.:-1870   +   :12   1st Qu.:   0
##  Median :2184   Median :-1024   NA's: 7   Median :   0
##  Mean   :1993   Mean   : -977             Mean   :  95
##  3rd Qu.:2396   3rd Qu.: -215             3rd Qu.:   0
##  Max.   :3472   Max.   : 1087             Max.   :1087
##
##
##  gene_id=11, length=399, gene_score=-501.312883, rbs_score=-0.013333, rbs_spacer=-1, stop_enforced=N
##  gene_id=12, length=162, gene_score=-202.567641, rbs_score=-0.013333, rbs_spacer=-1, stop_enforced=N
##  gene_id=12, length=837, gene_score=-1007.137868, rbs_score=0.033333, rbs_spacer=-1, stop_enforced=N
##  gene_id=13, length=1026, gene_score=-1333.766319, rbs_score=-1.164513, rbs_spacer=31, stop_enforced=
##  gene_id=13, length=717, gene_score=-939.159198, rbs_score=-0.013333, rbs_spacer=-1, stop_enforced=N
##  (Other)
##  NA's
```

Create skeletton of figure

We make the figure in two steps. 1. We firstcreate the tracks for the strains. we asign the number of strains to plot and the length of the sequences. 2. We plot the CDS ans IS elements on those tracks.

```
Xm <- list()
XM <- list()
for (i in levels(as.factor(ALL2$V1))) {
    Xm[[i]] <- min(ALL2[ALL2$V1 == i, 4:5])
    XM[[i]] <- max(ALL2[ALL2$V1 == i, 4:5])
}

# segment coordinates
df <- cbind.data.frame(unlist(Xm), unlist(XM))
df <- df[match(rle(ALL2$V1)[[2]], rownames(df)), ]
df <- cbind.data.frame(df, YM = seq(1, length(levels(as.factor(ALL2$V1)))) +
    0.02, Ym = seq(1, length(levels(as.factor(ALL2$V1)))) - 0.02)



df <- df[rownames(df) %in% c("SDF", "6200", "AB0057", "WKA02", "2011ZJAB4",
    "NCGM237", "KAB05", "AbA118", "ATCC17978Yale", "ATCC19606"), ]
# plot total segments
```

```
plotlines <- ggplot(aes(xmin = df[, 1], xmax = df[, 2], ymin = df[,
    3], ymax = df[, 4]), data = NULL) + geom_rect() + scale_y_discrete(breaks = seq(1:length(df$YM)),
    labels = rownames(df)) + annotate("text", x = -1000, y = (1:length(rownames(df))),
    label = rownames(df))
head(df)
```

```
##               unlist(Xm) unlist(XM) YM   Ym
## 2011ZJAB4              1       2378  1 0.98
## 6200                   1       3472  2 1.98
## AB0057                 1       2383  3 2.98
## AbA118                 1       2382  4 3.98
## ATCC17978Yale          1       2382  5 4.98
## ATCC19606              1       2382  6 5.98
```

Organize data to plot

We organize the data into a compatible data frame and the plot the info of CDS and IS elements on the previous tracks.

```
################
CDS_Ym <- rep(1:length(levels(as.factor(ALL2$V1))), rle(ALL2$V1)[[1]]) -
    0.45
CDS_YM <- rep(1:length(levels(as.factor(ALL2$V1))), rle(ALL2$V1)[[1]]) +
    0.45
df_CDS <- cbind.data.frame(ALL2$V4, ALL2$V5, CDS_Ym, CDS_YM)



# colors non automatic
colors_CDS <- rep("A", length(ALL2$V1))
ALL2$V3 <- as.character(ALL2$V3)

colors_CDS[ALL2$V3 == "CDS"] <- "black"
colors_CDS[is.na(ALL2$V3)] <- "red"

# final plot
plotlines2 <- plotlines + geom_rect(aes(xmin = ALL2$V4, xmax = ALL2$V5,
    ymin = CDS_Ym, ymax = CDS_YM), data = NULL, fill = colors_CDS,
    alpha = 2/4) + theme_bw() + theme(panel.border = element_blank(),
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    axis.line = element_line(colour = "black"), axis.title.x = element_blank(),
    axis.title.y = element_blank(), axis.ticks.x = element_blank()) +
    theme(plot.margin = unit(c(1, 1, 1.5, 1.2), "cm"))
head(df_CDS)
```

```
##   ALL2$V4 ALL2$V5 CDS_Ym CDS_YM
## 1       1     651   0.55    1.4
## 2     651    2378   0.55    1.4
## 3       1    1680   1.55    2.5
## 4    1725    2750   1.55    2.5
## 5    2747    3472   1.55    2.5
## 6    1671    2757   1.55    2.5
```

```
plotlines2
```

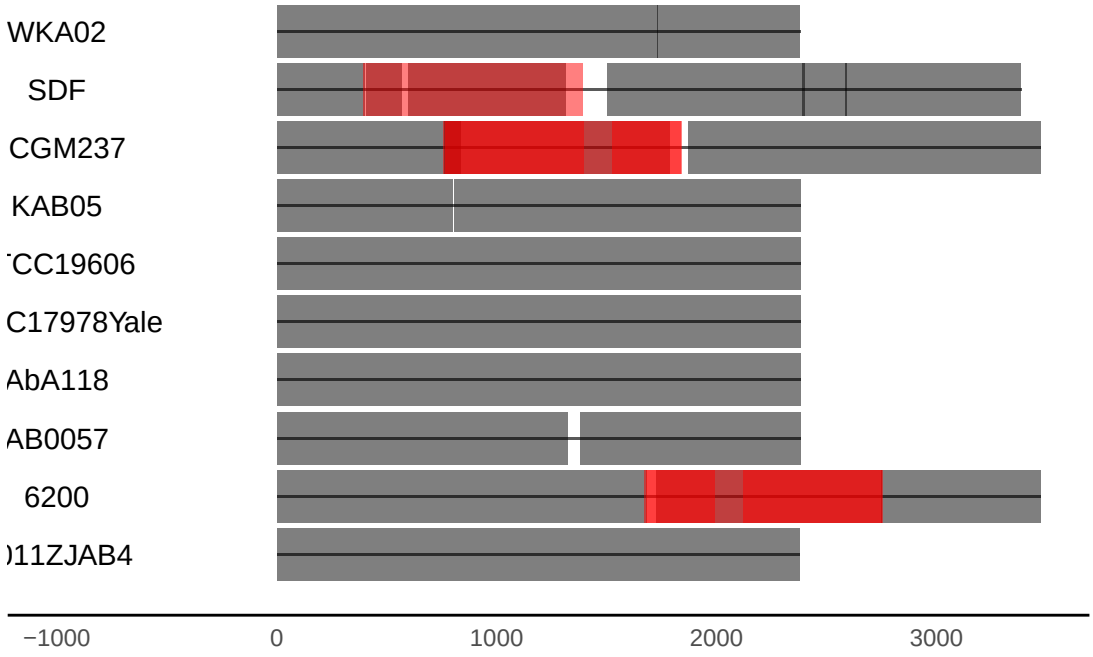

Figure 1: Sequence interruption

We observe that several strains do not have the ComA gene interrupted (WKA2, KAB05, ATCC19606, ATCC17978Yale, Aba118 and 2011ZJAB4). We also observe that the SDF, NCGM237 and 6200 strains, have an insertion of an IS element. The interruption in ComA show us that different IS elements are inserted in different parts of the gene on the different strains. finally we observe a insertion of few nucleotides on AB0057 creating a frameshift of the sequence.

Sequence statistics

Align all the sequences using ClustalO, then analyse nb of sites, SNPs, and Syn/non-Syn statistics.

```
setwd("~/Documents/EPFL/A_baumannii/Reference_sequences/ALL_Abaum_vf/ALL_Analysis_V2/db_genomes_all/V2_a

# NEED TO ALIGN SEQUENCES AND PLACE IN FOLDER BEFOREHAND clustalo
# --in ALL_V2_pilCblast2.fa --out ALL_V2_pilCblast2_align.fa

# import aligned fasta
GENOME.class <- readData("~/Documents/EPFL/A_baumannii/Reference_sequences/ALL_Abaum_vf/ALL_Analysis_V2/
    include.unknown = F)
```

```
## |           :           |           :           | 100 %
## |=================================================
```

```
# calculate statistics
GENOME.class <- F_ST.stats(GENOME.class)
```

```
## nucleotide
## |           :           |           :           | 100 %
## |=================================================
## haplotype
```

```
## |            :            |            :            | 100 %
## |==================================================
```

```
GENOME.class <- neutrality.stats(GENOME.class)
```

```
## |            :            |            :            | 100 %
## |==================================================
```

```
# get summary statistics
get.sum.data(GENOME.class)
```

```
##                       n.sites n.biallelic.sites n.gaps n.unknowns
## ALL_V2_comA_align.fa    3706               457   2707          0
##                       n.valid.sites n.polyallelic.sites trans.transv.ratio
## ALL_V2_comA_align.fa            566                 433               0.52
```

```
# calculate Nb of synonimouss and nonsyn
table(GENOME.class@region.data@synonymous[[1]])  # false is non-syn, True is syn
```

```
##
## FALSE  TRUE
##   445    12
```

```
# biallelic + syn
syn <- GENOME.class@region.data@synonymous[[1]]
syn[syn == TRUE] <- "Syn"
syn[syn == FALSE] <- "Non_Syn"
```

Plot Sequence statistics

```
for (i in 1:dim(get.sum.data(GENOME.class))[1]) {

    syn <- GENOME.class@region.data@synonymous[[i]]
    syn[syn == TRUE] <- "Syn"
    syn[syn == FALSE] <- "Non_Syn"

    stat1 <- c(total_sites = get.sum.data(GENOME.class)[i, 1], gaps = get.sum.data(GENOME.class)[i,
        3], na = get.sum.data(GENOME.class)[i, 4], valid_sites = get.sum.data(GENOME.class)[i,
        5])
    stat2 <- c(biallelic_sites = get.sum.data(GENOME.class)[i, 2],
        syn = length(syn[syn == "Syn"]), non_syn = length(syn[syn ==
            "Non_Syn"]), transl_transv_ratio = get.sum.data(GENOME.class)[i,
            7])
    stat2 <- round(stat2, digits = 2)

    barplot(stat1, col = "black", names.arg = names(stat1), las = 2,
        ylim = c(0, max(stat1) * 1.2), main = "Sites stats")
    text(x = seq(1, length(stat1)), y = stat1, label = stat1, pos = 3,
        cex = 1, col = "black")

    barplot(stat2, col = "black", names.arg = names(stat2), las = 2,
        ylim = c(0, max(stat2) * 1.2), main = "Biallelic sites stats")
    text(x = seq(1, length(stat2)), y = stat2, label = stat2, pos = 3,
        cex = 1, col = "black")
}
```

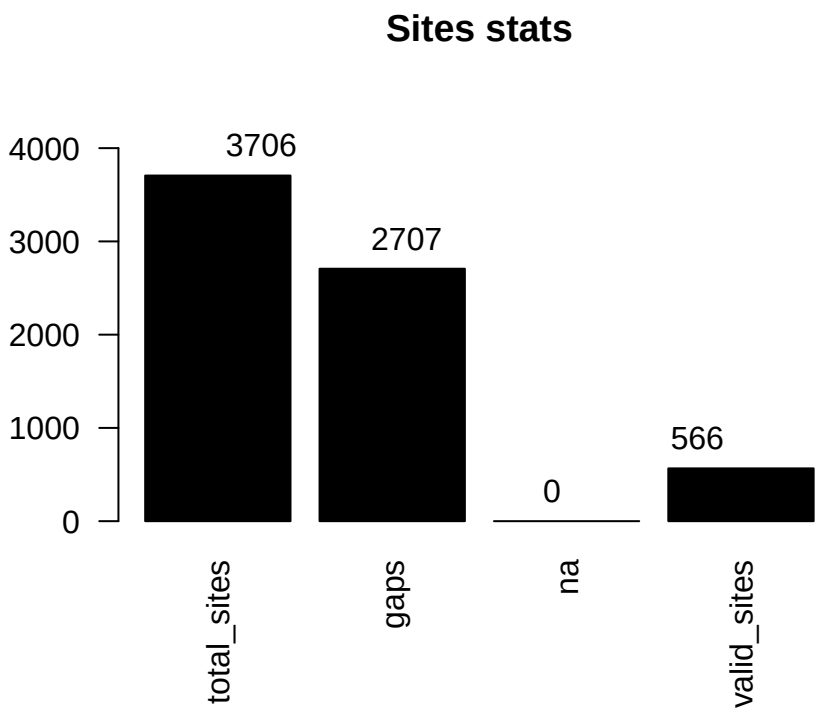

Figure 2: Sequence statistics

This Statistics show that the total number of sites is 3706. It also tell us that the minimum sequence length is about 2.7 kb (Figure 2). and that there were 566 sites that a SNP is present.

We then observed that the interrupted ComA gene sequence is very similar across the strains. There is only 12 synonimoius mutations. out of 457. This result could be the result of the presence of frameshifts and insertion of IS elements on the different strains.

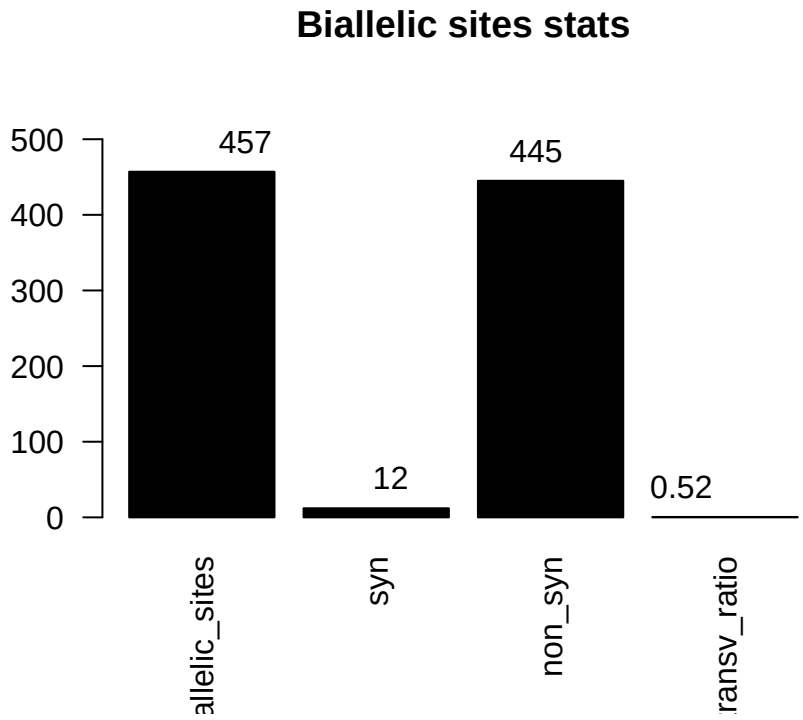

Figure 3: Sequence statistics