Analysis of annotated D. firmibasis JAVFKY000000000 assembly

Bart Edelbroek

Load packages and define functions

```
library(circlize)
library(RColorBrewer)
library(tidyverse)
library(ggplot2)
library(karyoploteR)

scale_rows = function(x){
    m = apply(x, 1, mean, na.rm = T)
    s = apply(x, 1, sd, na.rm = T)
    return((x - m) / s)
}

interpolate = function(input_data, interpol=10){
  rslice <- input_data[1,]
  save_mean <- data.frame(input_data[1,-(1:3)])
  reduced_data <- data.frame()
  for (i in 2:nrow(input_data)) {
    if (input_data[i,1]==rslice[1] & nrow(save_mean) < interpol) {
      rslice[3] <- input_data[i,3]
      save_mean <- rbind(save_mean, input_data[i,-(1:3)])
    } else  {
      rslice[-(1:3)] <- colMeans(save_mean)
      reduced_data <- rbind(reduced_data, rslice)
      rslice <- input_data[i,]
      save_mean <- data.frame(input_data[1,-(1:3)])
    }
  }
    rslice[-(1:3)] <- colMeans(save_mean)
  reduced_data <- rbind(reduced_data, rslice)
  return(reduced_data)
}

theme_set(theme_bw())
theme_update(text = element_text(size = 8))
```

Visualization of the D. firmibasis assembly and read coverage.

mRNA and gDNA read mapping data was analyzed in 2500bp intervals and counted, with regions and counts located in the *count_regions* folder. Genome data such as gene locations generated with scripts in *genomes*

folder.

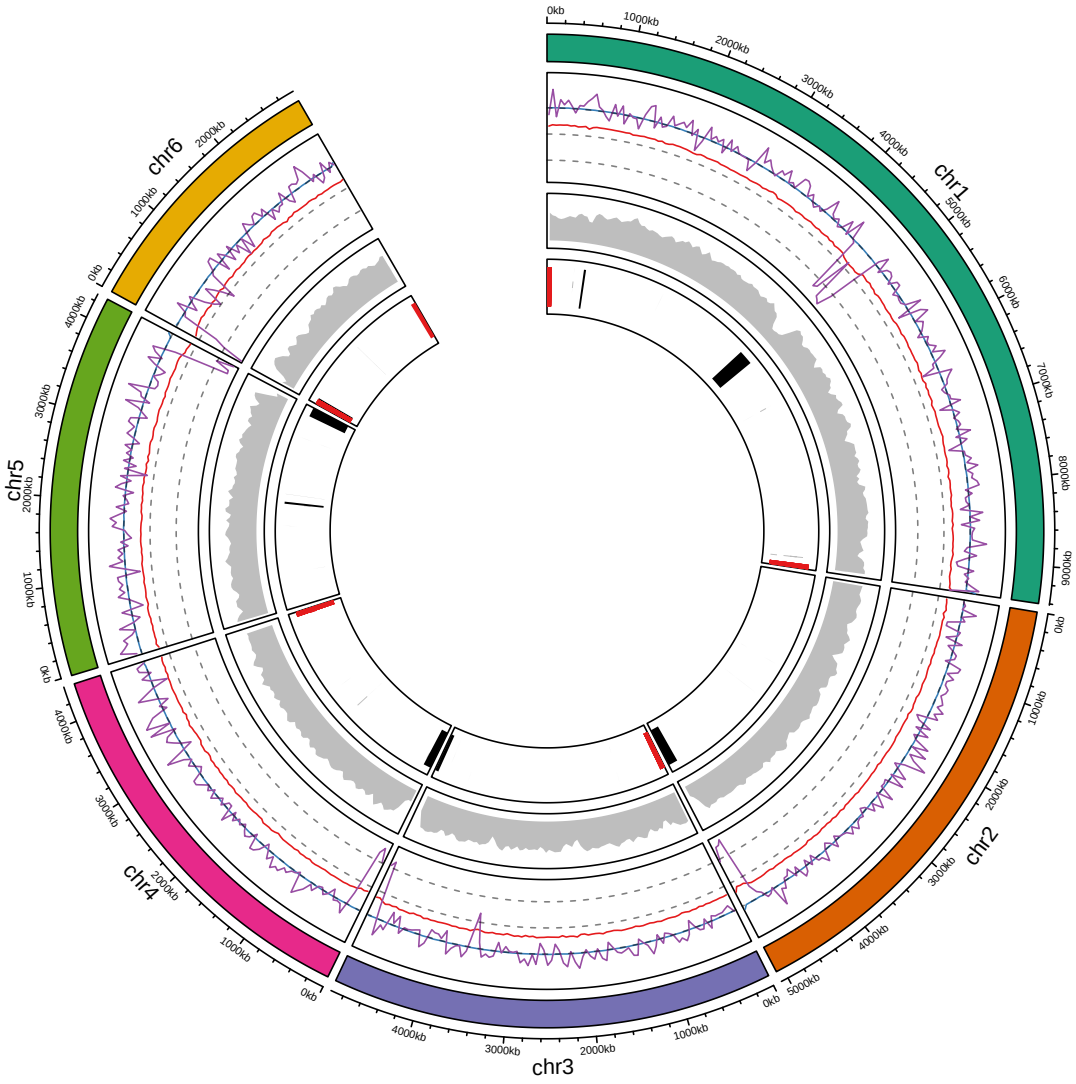
```
options(scipen = 999)

#Read in data and reformat

dfir_index <- read.table("genomes/dfir_genome.fa.fai", sep = "\t")
firmibasis <- data.frame(name = dfir_index[,1],
                         start = rep(0,nrow(dfir_index)),
                         end = dfir_index[,2])
firmibasis$name <- factor(firmibasis$name, levels = firmibasis$name)
firmibasis$alias <- c(paste0("Dfir_chr",1:6),"","","Dfir_mtDNA",rep("",3))

ddis_index <- read.table("genomes/ddis_genome.fasta.fai", sep = "\t")[c(5:11,1:4),]
discoideum <- data.frame(name = ddis_index[,1],
                         start = rep(0,nrow(ddis_index)),
                         end = ddis_index[,2])
discoideum$name <- factor(discoideum$name, levels = discoideum$name)

dfir_gene_locations <- read.table("genomes/dfir_genes_location.txt", sep = " ")


dfir_sRNA <- read.table("count_regions/regions_sRNA.txt", sep = "\t", header = T)
dfir_sRNA <- dfir_sRNA[order(dfir_sRNA$Chr),]

dfir_mRNA_gDNA_cov <- read.table("coverage/regions_cov.txt", sep = "\t", header = T)
dfir_mRNA_gDNA_cov <- dfir_mRNA_gDNA_cov[order(dfir_mRNA_gDNA_cov$Contig),]
dfir_coverage <- cbind.data.frame(dfir_mRNA_gDNA_cov, sRNA =   dfir_sRNA[,7] )

dfir_TE <- read.table("transposable_elements/out.txt", sep = "\t", header = F)
dfir_TE <- dfir_TE[with(dfir_TE, order(V1,V4,V2)),]
colnames(dfir_TE) <- c("chr","start","end","TE_ID","length","eval","bitscore")
dfir_TE$min <- apply(dfir_TE[,2:3],1,min)
dfir_TE$max <- apply(dfir_TE[,2:3],1,max)
dfir_dirs_TE<- dfir_TE[dfir_TE$TE_ID=="DIRS1",]

rRNAs <- read.table("genomes/dfir_rRNA_location.txt",sep = " ")

dfir_telomeres <- read.table("genomes/dfir_telomere_locations.txt",sep = " ")
dfir_telomeres[,4] <- 1
dfir_telomeres_not_ext <- dfir_telomeres
dfir_telomeres$V3[dfir_telomeres$V3 < 1e5] <- dfir_telomeres$V3[dfir_telomeres$V3 < 1e5] + 1e5
dfir_telomeres$V2[dfir_telomeres$V2 > 1e6] <- dfir_telomeres$V2[dfir_telomeres$V2 > 1e6] - 1e5
dfir_telomeres_not_ext$V3[dfir_telomeres_not_ext$V3 < 1e4] <- dfir_telomeres_not_ext$V3[dfir_telomeres_
dfir_telomeres_not_ext$V2[dfir_telomeres_not_ext$V2 > 1e5] <- dfir_telomeres_not_ext$V2[dfir_telomeres_

#Merge Transposable elements within close proximity

overlap <- 50000
rslice <- dfir_dirs_TE[1,]
dfir_dirs_TE_stitched <- data.frame()
for (i in 1:nrow(dfir_dirs_TE)-1) {
  if (all(rslice[c(1,4)]==dfir_dirs_TE[i+1,c(1,4)],
          rslice[9]+overlap>dfir_dirs_TE[i+1,8]))
```

```
    {
      rslice[9] <- max(as.numeric(rslice[9]),dfir_dirs_TE[i+1,9]) #add the data from the next hit
      rslice[8] <- min(as.numeric(rslice[8]),dfir_dirs_TE[i+1,8])
    } else { #the next hit is not within range
      dfir_dirs_TE_stitched <- rbind(dfir_dirs_TE_stitched,rslice)
      rslice <- dfir_dirs_TE[i+1,]
  }
}
dfir_dirs_TE_stitched <- rbind(dfir_dirs_TE_stitched,rslice) #add final row
dfir_dirs_TE_stitched <- cbind(dfir_dirs_TE_stitched[c(1,8,9)],1,dfir_dirs_TE_stitched[4])


#Interpolate read mapping regions

reduced_dfir_coverage <- interpolate(dfir_coverage, interpol = 20)
reduced_8_dfir_coverage <- interpolate(dfir_coverage, interpol = 8)

reduced_dfir_coverage[,4:7] <- log10(1+reduced_dfir_coverage[,4:7])


#Prepare and plot D. firmibasis assembly

firmibasis_main <- firmibasis[1:6,]
firmibasis_main$name <- factor(firmibasis_main$name, levels = firmibasis_main$name)
firmibasis_extra <- firmibasis[7:12,]
firmibasis_extra$name <- factor(firmibasis_extra$name, levels = firmibasis_extra$name)


dfir_gene_locations_plot <- cbind(dfir_gene_locations[1:3],1)
bed_list <- list(dfir_dirs_TE_stitched, dfir_telomeres )

genDens <- genomicDensity(dfir_gene_locations, window.size = 1e5)
genDens <- genDens[genDens$chr %in% firmibasis_main$name,]
mean(genDens$value)
```

```
## [1] 0.7055948
```

```
#svglite::svglite("plots/dfir_main_circos.svg",width = 6.7, height = 6.7)


ylims <- cbind(cbind(c(2,1,6)*-1,c(2,1,6))+round(apply(reduced_dfir_coverage[,4:6], 2, median)),round(ap

circos.clear()
circos.par(start.degree = 90, gap.degree = append(rep(1, nrow(firmibasis_main)-1),30))
circos.genomicInitialize(firmibasis_main[,1:3],
                         sector.names = c(paste0("chr",1:6)))
circos.track(ylim = c(0, 1),
             bg.col = c(brewer.pal(7,"Dark2"),rep("white",7)),
             track.height = 0.05)
circos.genomicTrack(reduced_dfir_coverage[,c(1:3,5,4,6)],
                                      ylim = c(0.5,4),
                                      panel.fun = function(region, value, ...) {
  circos.genomicLines(region, value, col = brewer.pal(4,"Set1")[-3], lwd = 1, ...)
```

```
    circos.lines(CELL_META$cell.xlim, c(1,1), lty = 2, col = "#00000080")
    circos.lines(CELL_META$cell.xlim, c(2,2), lty = 2, col = "#00000080")
    circos.lines(CELL_META$cell.xlim, c(3,3), lty = 2, col = "#00000080")
},track.height = 0.2)
circos.genomicTrack(genDens, ylim = c(0,1), track.height = 0.1,
                    panel.fun = function(region, value, ...) {
  circos.genomicLines(region, value, col = "grey", area = TRUE, border = F, ...)
    #circos.lines(CELL_META$cell.xlim, c(mean(genDens$value),mean(genDens$value)), lty = 2, col = "blac
                    })
circos.genomicTrack(bed_list, ylim = c(0,1),
                                        panel.fun = function(region, value, ...) {
                                          i = getI(...)
                                          circos.genomicRect(region,value, border = NA,  col = c("black
},track.height = 0.1)
```



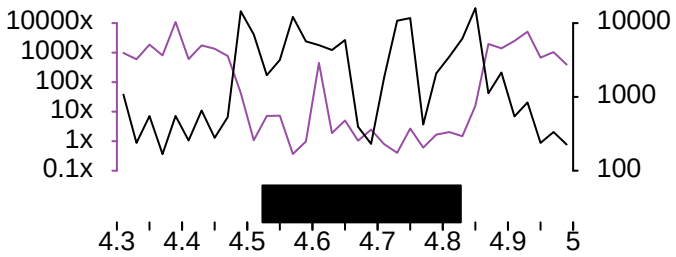

```
#dev.off()
```

Mapping of mRNA and small RNA on DIRS-1 region on chr1.

sRNA mapping data was analyzed as for the mRNA and gDNA, and mapping is visualised on the DIRS-1 region

```
dfir_gene_locations_dirs <- dfir_gene_locations[dfir_gene_locations[,1]=="contig_31_np1212"&dfir_gene_l

dirs_chr1 <- toGRanges(data.frame(chr="contig_31_np1212", start=4300000, end=5000000))
dirs1_annot_chr1 <- toGRanges(data.frame(dfir_dirs_TE_stitched[dfir_dirs_TE_stitched$TE_ID=="DIRS1",1:3
```

```
#svglite::svglite("plots/dirs1_sRNA_expression.svg",width = 6, height = 2)
pp <- getDefaultPlotParams(plot.type=1)
pp$leftmargin <- 0.2
pp$rightmargin <- 0.2
kp <- plotKaryotype(genome = dirs_chr1, cytobands = dirs1_annot_chr1, labels.plotter = NULL, plot.params
kpAddBaseNumbers(kp, tick.dist = 100000, tick.len = 10, tick.col="black", cex=0.8,
                 minor.tick.dist = 50000, minor.tick.len = 10, minor.tick.col = "black")
kpLines(kp, toGRanges(data.frame(reduced_8_dfir_coverage[905:939,1:3]), y=log10(reduced_8_dfir_coverage$r
kpLines(kp, toGRanges(data.frame(reduced_8_dfir_coverage[905:939,1:3]), y=log10(reduced_8_dfir_coverage$s
#kpRect(kp, chr=dfir_gene_locations_dirs[,1], x0 = dfir_gene_locations_dirs[,2], x1 = dfir_gene_locatio
kpAxis(kp, side = 1, tick.pos = c(0,0.2,0.4,0.6,0.8,1), labels = c("0.1x","1x","10x","100x","1000x","10
kpAxis(kp, side = 2, numticks = 3, tick.pos = c(0,0.5,1), labels = c(100,1000,10000), cex = 0.8, col =
```




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

Mean coverage of different data types

```
round(colMeans(dfir_coverage[,4:7]))
```

```
## Illumina Nanopore     mRNA     sRNA
##      541      200     1576     1154
```

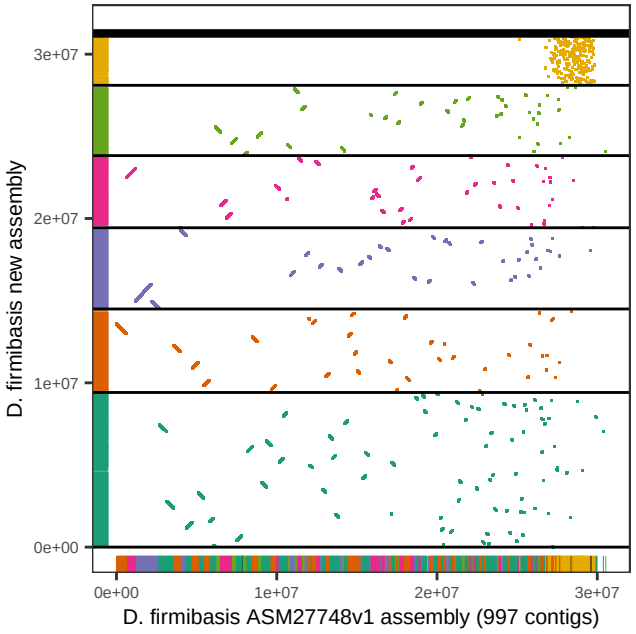
Synteny analysis of new and old D. firimbasis assembly

Synteny calculated with Satsuma2 between the GenBank GCA_000277485.1 assembly and GenBank JAVFKY000000000 assembly. Homologous regions are plotted.

```
dfir_old_sats_in <- read.table("comparative_genomics/dfir_vs_old.out", sep = "\t", header = F)
dfir_old_sats_in <- dfir_old_sats_in[with(dfir_old_sats_in, order(V1,V2)),]


# Merge regions which are in close proximity in both genomes

overlap <- 5000
rslice <- as.character(dfir_old_sats_in[1,])
```

```
dfir_old_sats <- data.frame()
for (i in 1:nrow(dfir_old_sats_in)-1) {
  if (all(rslice[c(1,4,8)]==dfir_old_sats_in[i+1,c(1,4,8)])) {
    if (all(rslice[8]=="-", #the next hit is also reverse and within range
        as.numeric(rslice[5])-overlap<dfir_old_sats_in[i+1,6],
        as.numeric(rslice[3])+overlap>dfir_old_sats_in[i+1,2])) {
      rslice[3] <- max(as.numeric(rslice[3]),dfir_old_sats_in[i+1,3]) #add the data from the next hit
      rslice[5] <- min(as.numeric(rslice[5]),dfir_old_sats_in[i+1,5])
    } else if (all(rslice[8]=="+", #the next hit is also in the same direction and within range
                   as.numeric(rslice[6])+overlap>dfir_old_sats_in[i+1,5],
                   as.numeric(rslice[3])+overlap>dfir_old_sats_in[i+1,2])) {
      rslice[3] <- max(as.numeric(rslice[3]),dfir_old_sats_in[i+1,3]) #add the data from the next hit
      rslice[6] <- max(as.numeric(rslice[6]),dfir_old_sats_in[i+1,6])
    } else { #the next hit is not within range
      dfir_old_sats <- rbind(dfir_old_sats,rslice)
      rslice <- as.character(dfir_old_sats_in[i+1,])
    }

  } else { #the next hit is different (chromosome, direction)
    dfir_old_sats <- rbind(dfir_old_sats,rslice)
    rslice <- as.character(dfir_old_sats_in[i+1,])
  }
}

dfir_old_sats <- rbind(dfir_old_sats,rslice) #add final row

dfir_old_sats[,4] <- sapply(strsplit(dfir_old_sats[,4],"_"), `[`, 1)

dfir_old_index <- read.table("genomes/dfir_old_genome.fa.fai", sep = "\t")
firmibasis_old <- data.frame(name = dfir_old_index[,1],
                         start = rep(0,nrow(dfir_old_index)),
                                                 end = dfir_old_index[,2])
firmibasis_old$name <- factor(firmibasis_old$name, levels = firmibasis_old$name)

dfir_old_sats[,2] <- as.numeric(dfir_old_sats[,2])
dfir_old_sats[,3] <- as.numeric(dfir_old_sats[,3])
dfir_old_sats[,5] <- as.numeric(dfir_old_sats[,5])
dfir_old_sats[,6] <- as.numeric(dfir_old_sats[,6])


adj_df <- dfir_old_sats
firmibasis$add <- c(0,cumsum(firmibasis$end)[-length(firmibasis$end)])
adj_df[,c(2,3)] <- adj_df[,c(2,3)]+firmibasis$add[match(adj_df[,1],firmibasis$name)]

firmibasis_old$add <- c(0,cumsum(firmibasis_old$end)[-length(firmibasis_old$end)])
adj_df[,c(5,6)] <- adj_df[,c(5,6)]+firmibasis_old$add[match(adj_df[,4],firmibasis_old$name)]


#Calculate 2D homology in 10000bp intervals

precision <- 10000
correlation_df <- data.frame()
i <- 1
```

```
for (i in 1:nrow(adj_df)) {
  seq_length <- (adj_df[i,3]-adj_df[i,2])/precision
  x <- seq(adj_df[i,2],adj_df[i,3],length.out = seq_length)
    if (adj_df[i,8]=="-") {
      y <- seq(adj_df[i,6],adj_df[i,5],length.out = seq_length)
    } else if (adj_df[i,8]=="+") {
      y <- seq(adj_df[i,5],adj_df[i,6],length.out = seq_length)
    }
  name <- rep(adj_df[i,1],length(x))
  correlation_df <- rbind(correlation_df, cbind(x,y,name))
}
correlation_df$x <- as.numeric(correlation_df$x)
correlation_df$y <- as.numeric(correlation_df$y)
correlation_df$name <- firmibasis$alias[match(correlation_df$name,firmibasis$name)]

ggplot(data = correlation_df, mapping = aes(y=x,x=y, col = name))+
  geom_point(shape = ".")+
  geom_rug(alpha = 1, lwd = 0.1)+xlab("D. firmibasis ASM27748v1 assembly (997 contigs)")+ylab("D. firmi
  geom_hline(yintercept = firmibasis$add)+
  scale_color_manual(values = c("black",brewer.pal(7,"Dark2")))+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(), panel.background = eleme
a
```



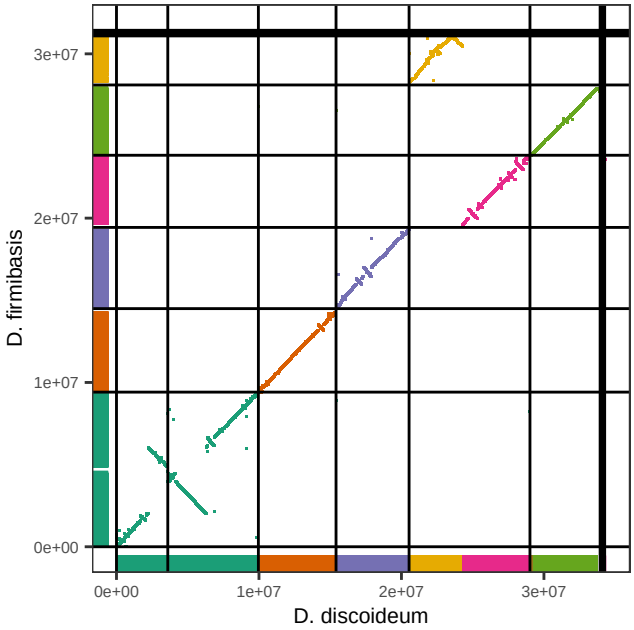

```
#ggsave("plots/dfir_old.png",a,"png", width = 3.8,height = 3.8, dpi = 1200)
```

Synteny analysis of D. discoideum and new D. firimbasis assembly

As above, but synteny calculated between the dictybase D. discoideum assembly and GenBank JAVFKY000000000 assembly.

```
dfir_sats_in <- read.table("comparative_genomics/ddis_vs_dfir.out", sep = "\t", header = F)
dfir_sats_in <- dfir_sats_in[with(dfir_sats_in, order(V1,V2)),]

# Merge regions which are in close proximity in both genomes

overlap <- 5000
rslice <- as.character(dfir_sats_in[1,])
dfir_sats <- data.frame()
for (i in 1:nrow(dfir_sats_in)-1) {
  if (all(rslice[c(1,4,8)]==dfir_sats_in[i+1,c(1,4,8)])) {
    if (all(rslice[8]=="-", #the next hit is also reverse and within range
        as.numeric(rslice[5])-overlap<dfir_sats_in[i+1,6],
        as.numeric(rslice[3])+overlap>dfir_sats_in[i+1,2])) {
      rslice[3] <- max(as.numeric(rslice[3]),dfir_sats_in[i+1,3]) #add the data from the next hit
      rslice[5] <- min(as.numeric(rslice[5]),dfir_sats_in[i+1,5])
    } else if (all(rslice[8]=="+", #the next hit is also in the same direction and within range
                   as.numeric(rslice[6])+overlap>dfir_sats_in[i+1,5],
                   as.numeric(rslice[3])+overlap>dfir_sats_in[i+1,2])) {
      rslice[3] <- max(as.numeric(rslice[3]),dfir_sats_in[i+1,3]) #add the data from the next hit
      rslice[6] <- max(as.numeric(rslice[6]),dfir_sats_in[i+1,6])
    } else { #the next hit is not within range
      dfir_sats <- rbind(dfir_sats,rslice)
      rslice <- as.character(dfir_sats_in[i+1,])
    }

  } else { #the next hit is different (chromosome, direction)
    dfir_sats <- rbind(dfir_sats,rslice)
    rslice <- as.character(dfir_sats_in[i+1,])
  }
}
dfir_sats <- rbind(dfir_sats,rslice) #add final row

dfir_sats[,1] <-substr(dfir_sats[,1],1,10)

dfir_sats[,2] <- as.numeric(dfir_sats[,2])
dfir_sats[,3] <- as.numeric(dfir_sats[,3])
dfir_sats[,5] <- as.numeric(dfir_sats[,5])
dfir_sats[,6] <- as.numeric(dfir_sats[,6])

adj_df <- dfir_sats
firmibasis$add <- c(0,cumsum(firmibasis$end)[-length(firmibasis$end)])
adj_df[,c(5,6)] <- adj_df[,c(5,6)]+firmibasis$add[match(adj_df[,4],firmibasis$name)]


#Reorder D. discoideum chromosomes for plotting

discoideum$flip <- 0
discoideum$flip[c(1,4,6)] <- discoideum$end[c(1,4,6)]
adj_df[,c(2,3)] <- abs(adj_df[,c(2,3)]-discoideum$flip[match(adj_df[,1],discoideum$name)])

rearrange <- c(6,3,4,5,2,1,7:nrow(discoideum))
discoideum_rearranged <- discoideum[rearrange,]
discoideum_rearranged$add <- c(0,cumsum(discoideum_rearranged$end)[-length(discoideum_rearranged$end)])
```

```
adj_df[,c(2,3)] <- adj_df[,c(2,3)]+discoideum_rearranged$add[match(adj_df[,1],discoideum_rearranged$nam


#Calculate 2D homology in 10000bp intervals

precision <- 1000
correlation_df <- data.frame()
i <- 1
for (i in 1:nrow(adj_df)) {
  seq_length <- (abs(adj_df[i,3]-adj_df[i,2]))/precision
  x <- seq(adj_df[i,2],adj_df[i,3],length.out = seq_length)
    if (adj_df[i,8]=="-") {
      y <- seq(adj_df[i,6],adj_df[i,5],length.out = seq_length)
    } else if (adj_df[i,8]=="+") {
      y <- seq(adj_df[i,5],adj_df[i,6],length.out = seq_length)
    }
  name <- rep(adj_df[i,4],length(x))
  correlation_df <- rbind(correlation_df, cbind(x,y,name))
}
correlation_df$x <- as.numeric(correlation_df$x)
correlation_df$y <- as.numeric(correlation_df$y)
correlation_df$name <- firmibasis$alias[match(correlation_df$name,firmibasis$name)]
correlation_df$name <- factor(correlation_df$name, levels = unique(firmibasis$alias))


ggplot(data = correlation_df, mapping = aes(x,y, col = name))+
  geom_point(shape = ".")+
  geom_rug(alpha = 1, lwd = 0.1)+xlab("D. discoideum")+ylab("D. firmibasis")+
  geom_vline(xintercept = discoideum_rearranged$add)+
  geom_hline(yintercept = firmibasis$add)+
  scale_color_brewer(palette = "Dark2")+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(), panel.background = eleme
a
```




```
#ggsave("plots/dfir_ddis.png",a,"png", width = 2.4,height = 2.4, dpi = 1200)
```

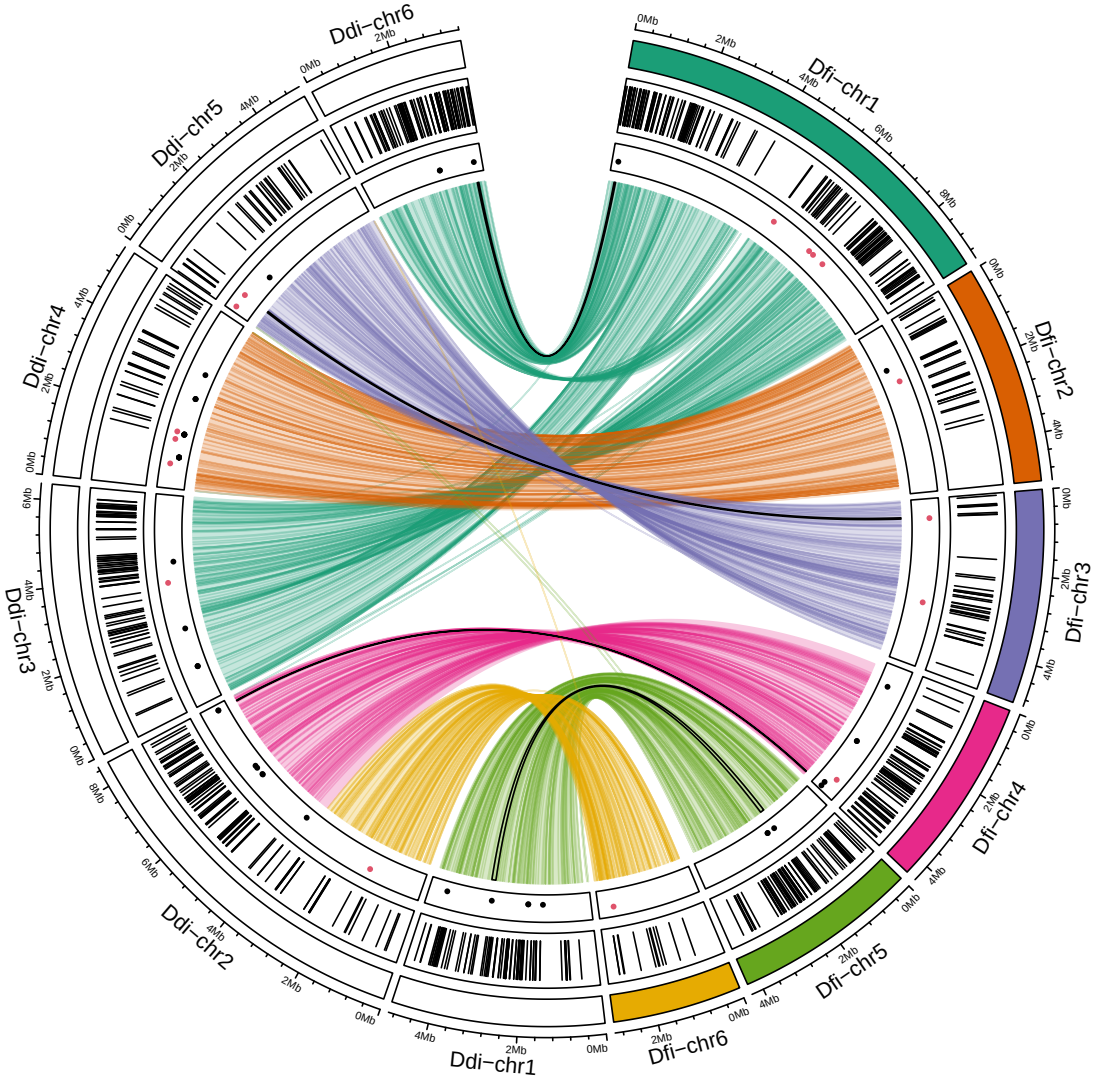
Detailed commparison between D. discoideum and D. firmibasis.

ncRNA locations generated and located in *genomes* folder.

```
links_df <- dfir_sats

fir_use_contigs <- c("contig_31_np1212","contig_57_np1212","contig_25_np1212",
                     "contig_29_np1212","contig_24_np1212","contig_32_np1212")
dis_use_contigs <- c("DDB0232428","DDB0232429","DDB0232430",
                     "DDB0232431","DDB0232432","DDB0232433")
combined <- rbind(firmibasis[firmibasis$name%in%fir_use_contigs,1:3],
                  discoideum[discoideum$name%in%dis_use_contigs,1:3])
combined$name <- factor(combined$name, levels = c(fir_use_contigs,dis_use_contigs))
links_df <- links_df[links_df[,1]%in%combined$name&links_df[,4]%in%combined$name,]


dfir_tRNAs <- read.table("genomes/dfir_tRNAs_location.txt",sep = " ")
ddis_tRNAs <- read.table("genomes/ddis_tRNAs_location.txt",sep = " ")
tRNA_comb <- rbind.data.frame(dfir_tRNAs,ddis_tRNAs)
tRNA_lines <- data.frame(chr = tRNA_comb$V1, start = tRNA_comb$V2, end = tRNA_comb$V3, value1 = 1.5, ty
bed_list <- list()
for (i in unique(tRNA_lines$type)) {
  bed_list  <- append(bed_list, list(tRNA_lines[tRNA_lines$type==i,1:4]))
}

classI <- read.table("genomes/curated_classI.bed",sep = "\t")
classI$V5 <- "classI"
miRNAs <- read.table("genomes/combined_miRNA_annotations.bed",sep = "\t")
miRNAs <- miRNAs[miRNAs$V8=="miRNA_primary_transcript",c(1:3,10,5,6)]
```

```
colnames(miRNAs) <- paste0("V",1:6)
miRNAs$V5 <- "miRNA"
miRNAs$V4 <- substr(miRNAs$V4, 6,30)
sRNAs <- rbind(classI,miRNAs)

dfir_sRNA_links <- data.frame()
for (i in 1:nrow(sRNAs)) {
  test <- c(which(links_df[,5]<sRNAs[i,2]&links_df[,6]>sRNAs[i,3]&sRNAs[i,1]==links_df[,4]))
  if (length(test) > 0) {
    dfir_sRNA_links <- rbind(dfir_sRNA_links,
                             data.frame(links_df[test,],"type" = sRNAs[i,5], "name" = sRNAs[i,4]))
  }
}
ddis_sRNA_links <- data.frame()
for (i in 1:nrow(sRNAs)) {
  test <- c(which(links_df[,2]<sRNAs[i,2]&links_df[,3]>sRNAs[i,3]&sRNAs[i,1]==links_df[,1]))
  if (length(test) > 0) {
    ddis_sRNA_links <- rbind(ddis_sRNA_links,
                             data.frame(links_df[test,],"type" = sRNAs[i,5], "name" = sRNAs[i,4]))
  }
}
sRNA_links <- dplyr::intersect(dfir_sRNA_links[,-10],ddis_sRNA_links[,-10])

bed_list <- list(data.frame(sRNAs[sRNAs$V5=="classI",1:3],value=1),
                 data.frame(sRNAs[sRNAs$V5=="miRNA",1:3],value=1))

#svglite::svglite("plots/discoideum_firmibasis_circos.svg",width = 6.7, height = 6.7)
circos.clear()
circos.par(start.degree = 80, gap.degree = append(rep(1, nrow(combined)-1),20))
circos.genomicInitialize(combined,
                         sector.names = c(paste0("Dfi-chr",1:6),
                                          paste0("Ddi-chr",1:6)))
circos.track(ylim = c(0, 1),
             bg.col = c(brewer.pal(6,"Dark2"),rep("white",6)),
             track.height = 0.05)
#circos.genomicDensity(tRNA_comb,col = c("#40404040"),track.height = 0.10, border = "#40404080", window
#circos.genomicRainfall(tRNA_comb,col = c("black"),track.height = 0.10, pch = 16, cex = 0.3)
circos.genomicTrack(tRNA_lines, stack = TRUE, track.height = 0.10,
    panel.fun = function(region, value, ...) {
        circos.genomicLines(region, value, type = "h", baseline = "bottom")
})
circos.genomicTrack(bed_list, stack = TRUE, track.height = 0.05,
    panel.fun = function(region, value, ...) {
        i = getI(...)
        circos.genomicPoints(region, value, pch = 16, cex = 0.5, col = i, ...)
})
for (i in 1:7) {
  circos.genomicLink(links_df[links_df[,4]==combined$name[i],1:3],
                     links_df[links_df[,4]==combined$name[i],4:6],
                     col = paste0(brewer.pal(7, "Dark2")[i],"40"))
}
for (i in 1:7) {
  circos.genomicLink(sRNA_links[sRNA_links[,4]==combined$name[i],1:3],
```

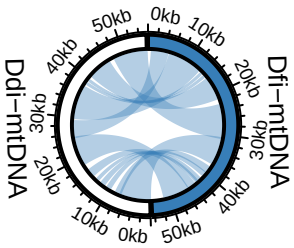
```
                    sRNA_links[sRNA_links[,4]==combined$name[i],4:6],
                    col = paste0(brewer.pal(7, "Dark2")[i],"40"), border = "black")
}
```




```
#dev.off()

circos.clear()
```

Compare mtDNA

```
use_contigs <- c("contig_65_np1212","DDB0169550")
combined <- rbind(firmibasis[firmibasis$name%in%use_contigs,1:3],
                  discoideum[discoideum$name%in%use_contigs,1:3])
combined$name <- factor(combined$name, levels = c(use_contigs))

mtDNA_sats_in <- dfir_sats_in
mtDNA_sats_in$V1 <- substr(mtDNA_sats_in$V1, 1, 10)
mtDNA_sats_in <- mtDNA_sats_in[mtDNA_sats_in[,1]%in%combined$name&mtDNA_sats_in[,4]%in%combined$name,]

overlap <- 500
rslice <- mtDNA_sats_in[1,]
mtDNA_sats <- data.frame()
for (i in 2:nrow(mtDNA_sats_in)) {
  if (all(rslice[c(1,4)]==mtDNA_sats_in[i,c(1,4)],
          abs(rslice[3]-mtDNA_sats_in[i,2])<overlap,
          abs(rslice[6]-mtDNA_sats_in[i,5])<overlap))
    {
      rslice[3] <- max(as.numeric(rslice[3]),mtDNA_sats_in[i,3]) #add the data from the next hit
      rslice[6] <- max(as.numeric(rslice[6]),mtDNA_sats_in[i,6])
    } else { #the next hit is not within range
      mtDNA_sats <- rbind(mtDNA_sats,rslice)
      rslice <- mtDNA_sats_in[i,]
  }
}
mtDNA_sats <- rbind(mtDNA_sats,rslice) #add final row




#svglite::svglite("plots/discoideum_firmibasis_mtDNA_circos.svg",width = 2.5, height = 2.5)
circos.clear()
circos.par(start.degree = 90)
circos.genomicInitialize(combined, major.by = 10000, axis.labels.cex = 0.6,
                         sector.names = c("Dfi-mtDNA","Ddi-mtDNA"))
circos.track(ylim = c(0, 1),
             bg.col = c("#377EB8", "white"),
             bg.border = "#000000", bg.lwd = 2, track.height = 0.1)
circos.genomicLink(mtDNA_sats[,1:3],
                   mtDNA_sats[,4:6],
                   col = "#377EB860")
```

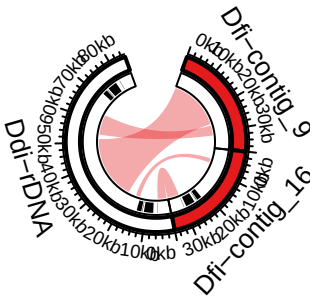



```
#dev.off()

circos.clear()
```

Comparison of the D. discoideum extrachromosomal DNA, with homologous contigs in D. firmibasis assembly JAVFKY000000000

```
use_contigs <- c("contig_9_np1212","contig_16_np1212","DDB0237465")
combined <- rbind(firmibasis[firmibasis$name%in%use_contigs,1:3],
                  discoideum[discoideum$name%in%use_contigs,1:3])
combined$name <- factor(combined$name, levels = c(use_contigs))

rDNA_sats_in <- dfir_sats_in
rDNA_sats_in$V1 <- substr(rDNA_sats_in$V1, 1, 10)
rDNA_sats_in <- rDNA_sats_in[rDNA_sats_in[,1]%in%combined$name&rDNA_sats_in[,4]%in%combined$name,]

overlap <- 5000
rslice <- rDNA_sats_in[1,]
rDNA_sats <- data.frame()
for (i in 2:nrow(rDNA_sats_in)) {
  if (all(rslice[c(1,4,8)]==rDNA_sats_in[i,c(1,4,8)],
          abs(rslice[3]-rDNA_sats_in[i,2])<overlap,
          abs(rslice[6]-rDNA_sats_in[i,5])<overlap))
    {
      rslice[3] <- max(as.numeric(rslice[3]),rDNA_sats_in[i,3]) #add the data from the next hit
      rslice[6] <- max(as.numeric(rslice[6]),rDNA_sats_in[i,6])
      rslice[2] <- min(as.numeric(rslice[2]),rDNA_sats_in[i,2]) #add the data from the next hit
      rslice[5] <- min(as.numeric(rslice[5]),rDNA_sats_in[i,5])
    } else { #the next hit is not within range
      rDNA_sats <- rbind(rDNA_sats,rslice)
      rslice <- rDNA_sats_in[i,]
  }
}
rDNA_sats <- rbind(rDNA_sats,rslice) #add final row


rRNAs_plot <- list(data.frame(rRNAs[,1:3],"value" = 1))
```

```
#svglite::svglite("plots/discoideum_firmibasis_rDNA_circos.svg",width = 2.5, height = 2.5)
circos.clear()
circos.par(start.degree = 70, gap.degree = c(0,0,40))
circos.genomicInitialize(combined, major.by = 10000, axis.labels.cex = 0.6,
                         sector.names = c("Dfi-contig_9","Dfi-contig_16","Ddi-rDNA"))
circos.track(ylim = c(0, 1),
             bg.col = c("#E41A1C","#E41A1C", "white"),
             bg.border = "#000000", bg.lwd = 2, track.height = 0.1)
circos.genomicTrack(rRNAs_plot, ylim = c(0,1),
                                        panel.fun = function(region, value, ...) {
                                          i = getI(...)
                                          circos.genomicRect(region,value, border = NA, col = "black",
},track.height = 0.1)
circos.genomicLink(rDNA_sats[,1:3],
                   rDNA_sats[,4:6],
                   col = "#E41A1C60")
```




```
#dev.off()

circos.clear()
```

Differential gene expression analysis in D. firmibasis and D. discoideum multicellular development

Gene counts are located in *transcriptomics* folder. Differentially expressed genes identified by Likelihood Ratio Test with DESeq2

```
#read data
pval <- 0.001

#Mean mapping percentage D. firmibasis
mean(94.66,94.78,90.59,93.01,94.72,92.79,92.95,94.79,93.37)
```

```
## [1] 94.66
```

```r
dfir_fc_summary <- read.table("transcriptomics/counts_dfir.txt.summary", header = T,row.names = 1)
rowMeans(dfir_fc_summary["Assigned",]/colSums(dfir_fc_summary))
```

```
##  Assigned
## 0.9395271
```

```r
#Mean mapping percentage D. discoideum
mean(87.86,89.96,89.39,89.99,93.08,91.41,86.99,90.96,87.91)
```

```
## [1] 87.86
```

```r
ddis_fc_summary <- read.table("transcriptomics/counts_ddis.txt.summary", header = T,row.names = 1)
rowMeans(ddis_fc_summary["Assigned",]/colSums(ddis_fc_summary))
```

```
##  Assigned
## 0.9767054
```

```r
DDBtable <- read.table("genomes/DDB-GeneID-UniProt.txt", sep = "\t", header = T)
rownames(DDBtable) <- DDBtable$DDB.ID


ddis_counts <- read.table("transcriptomics/counts_ddis.txt",sep = "\t", header = T)
rownames(ddis_counts) <- ddis_counts$Geneid
ddis_counts <- round(ddis_counts[,-c(1:6)])

dfir_counts <- read.table("transcriptomics/counts_dfir.txt",sep = "\t", header = T)
dfir_ID2name <- read.table("genomes/dfir_ID2name.txt",sep = "\t", header = F)
sum(grepl("Similar to ",dfir_ID2name$V2))
```

```
## [1] 10196
```

```r
length(dplyr::intersect(dfir_counts$Geneid[rowSums(dfir_counts[,7:15])<100],dfir_ID2name$V1[dfir_ID2nam
```

```
## [1] 577
```

```r
sessionInfo()
```

```
## R version 4.3.1 (2023-06-16)
## Platform: x86_64-apple-darwin20 (64-bit)
## Running under: macOS Ventura 13.6.7
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRlapack.dylib;  LAPACK
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: Europe/Stockholm
```

```
## tzcode source: internal
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] karyoploteR_1.28.0   regioneR_1.34.0      GenomicRanges_1.54.1
##  [4] GenomeInfoDb_1.38.5  IRanges_2.36.0       S4Vectors_0.40.2
##  [7] BiocGenerics_0.48.1  lubridate_1.9.3      forcats_1.0.0
## [10] stringr_1.5.1        dplyr_1.1.4          purrr_1.0.2
## [13] readr_2.1.5          tidyr_1.3.0          tibble_3.2.1
## [16] ggplot2_3.4.4        tidyverse_2.0.0      RColorBrewer_1.1-3
## [19] circlize_0.4.15
##
## loaded via a namespace (and not attached):
##   [1] rstudioapi_0.15.0           shape_1.4.6
##   [3] magrittr_2.0.3              GenomicFeatures_1.54.1
##   [5] farver_2.1.1                rmarkdown_2.25
##   [7] GlobalOptions_0.1.2         BiocIO_1.12.0
##   [9] zlibbioc_1.48.0             vctrs_0.6.5
##  [11] memoise_2.0.1               Rsamtools_2.18.0
##  [13] RCurl_1.98-1.14             base64enc_0.1-3
##  [15] htmltools_0.5.7             S4Arrays_1.2.0
##  [17] progress_1.2.3              curl_5.2.0
##  [19] SparseArray_1.2.3           Formula_1.2-5
##  [21] htmlwidgets_1.6.4           cachem_1.0.8
##  [23] GenomicAlignments_1.38.1    lifecycle_1.0.4
##  [25] pkgconfig_2.0.3             Matrix_1.6-4
##  [27] R6_2.5.1                    fastmap_1.1.1
##  [29] GenomeInfoDbData_1.2.11     MatrixGenerics_1.14.0
##  [31] digest_0.6.33               colorspace_2.1-0
##  [33] AnnotationDbi_1.64.1        bezier_1.1.2
##  [35] Hmisc_5.1-1                 RSQLite_2.3.4
##  [37] labeling_0.4.3              filelock_1.0.3
##  [39] fansi_1.0.6                 timechange_0.2.0
##  [41] httr_1.4.7                  abind_1.4-5
##  [43] compiler_4.3.1              bit64_4.0.5
##  [45] withr_2.5.2                 htmlTable_2.4.2
##  [47] backports_1.4.1             BiocParallel_1.36.0
##  [49] DBI_1.2.0                   highr_0.10
##  [51] biomaRt_2.58.0              rappdirs_0.3.3
##  [53] DelayedArray_0.28.0         rjson_0.2.21
##  [55] tools_4.3.1                 foreign_0.8-86
##  [57] nnet_7.3-19                 glue_1.7.0
##  [59] restfulr_0.0.15             grid_4.3.1
##  [61] checkmate_2.3.1             cluster_2.1.6
##  [63] generics_0.1.3              gtable_0.3.4
##  [65] BSgenome_1.70.1             tzdb_0.4.0
##  [67] ensembldb_2.26.0            data.table_1.14.10
##  [69] hms_1.1.3                   xml2_1.3.6
##  [71] utf8_1.2.4                  XVector_0.42.0
##  [73] pillar_1.9.0                BiocFileCache_2.10.1
##  [75] lattice_0.22-5              rtracklayer_1.62.0
```

```
##  [77] bit_4.0.5                  biovizBase_1.50.0
##  [79] tidyselect_1.2.0           Biostrings_2.70.1
##  [81] knitr_1.45                 gridExtra_2.3
##  [83] ProtGenerics_1.34.0        SummarizedExperiment_1.32.0
##  [85] xfun_0.41                  Biobase_2.62.0
##  [87] matrixStats_1.2.0          stringi_1.8.3
##  [89] lazyeval_0.2.2             yaml_2.3.8
##  [91] evaluate_0.23              codetools_0.2-19
##  [93] cli_3.6.2                  rpart_4.1.23
##  [95] munsell_0.5.0              dichromat_2.0-0.1
##  [97] Rcpp_1.0.12                dbplyr_2.4.0
##  [99] png_0.1-8                  XML_3.99-0.16
## [101] parallel_4.3.1             blob_1.2.4
## [103] prettyunits_1.2.0          AnnotationFilter_1.26.0
## [105] bitops_1.0-7               VariantAnnotation_1.48.1
## [107] scales_1.3.0               crayon_1.5.2
## [109] bamsignals_1.34.0          rlang_1.1.3
## [111] KEGGREST_1.42.0
```