plots_inversions.Rmd

Andrea Estandia

17/10/2022

Unzip beagle.gz window file if unzipped file does not exist already

```
#if selecting individual windows
beagle_file <- file.path(
  reports_path,
    "localPCA",
    "beagle_by_window",
    params$chr,
    params$beagle)

#if selecting a set of combined outlier windows
beagle_file <- file.path(
  reports_path,
    "localPCA",
    "outlier_mds",
    "beagle",
    params$beagle)

if (!file.exists(beagle_file)) {
gunzip(paste0(beagle_file, ".gz"), remove=FALSE)
}
```

Read beagle file, remove allele columns, rename columns and remove extra numbers after sample_name

```
outlier <-
  read.table(file.path(reports_path,
                       "localPCA",
                        "outlier_mds",
                        "beagle",
                       params$beagle), header=T) %>%
  select(-c("allele1", "allele2")) %>%
  as_tibble() %>%
  pivot_longer(starts_with("Ind")) %>%
  rename(sample_name=name) %>%
  rename(genotype=value) %>%
  mutate(sample_name=str_remove(sample_name, "\\.[^.]*$"))
```

Read phenotype file with body size (PC1), pop info and genetic sex

```
pheno <-
  read_tsv(file.path(data_path,
                     "wgs",
                     "lists",
                     "body_PC1.tsv"),
```

```
           col_names = FALSE) %>%
  rename(sample_name = X1) %>%
  rename(body_PC1 = X2)

pop_info <-
  read_csv(file.path(data_path,
                     "phenotypes",
                     "phenotypes+SMC2+AE.csv")) %>%
  rename(x = sample_name) %>%
  rename(sample_name = id) %>%
  select(sample_name, pop, region, latitude, longitude)

geneticsex <-
  read_csv(file.path(data_path,
                     paste0("phenotypes/geneticsex.csv"))) %>%
  rename(sample_name = Blood_Number)
```

Generate a vector repeating the genotypes AA, AB, and BB Paste vector to main beagle dataframe

```
geno <-
  rep(c("AA", "AB", "BB"),
      times = length(outlier$sample_name) / 3)

df <-
  cbind(outlier, geno)

df <- df %>%
  filter(genotype>0.34) %>%
  group_by(sample_name, marker) %>%
  top_n(1, genotype)
```

Merge all datasets and separate column marker into chr and position

```
df2 <-
  df %>%
  left_join(pop_info, by = "sample_name") %>%
  left_join(pheno, by = "sample_name") %>%
  separate(marker,into = c("chr", "position")) %>%
  mutate(position=as.numeric(position)) %>%
  unite("marker", chr:position, remove=FALSE)
```

Read covariance matrix for the window and population labels to give it row names

```
label <-
  read.table(file.path(data_path,"localPCA/pop_label"))

cov_mat <-
  as.matrix(read.table(file.path(
    reports_path,
      "localPCA",
      "outlier_mds",
      "cov",
      paste0(gsub("\\..*","",params$beagle), ".cov")
  )))
```

Decompose covariance matrix into its eigenvalues

```r
#Do MDS on cov matrix
mds.cor <- (1 - cov_mat) %>%
  cmdscale(k=3, eig = TRUE)

colnames(mds.cor$points) <- c("Dim.1", "Dim.2", "Dim.3")
rownames(mds.cor$points) <-
  label$V3

#Do PCA on cov matrix
pca<-eigen(cov_mat)

pca.mat <-
  as.matrix(pca$vectors %*% (diag(pca$values))^0.5)

nPC <-
  dim(pca$vectors)[2]

col_PC <-
  vector(length=nPC)

for (i in 1 : nPC) {col_PC[i] <-
  paste0("PC",i)}

#add column names
colnames(pca.mat) <-
  c(col_PC)

#add row names
rownames(pca.mat) <-
  label$V3

for (x in 1:4) {
  nam <-
    as.character(paste0("var",x))
    assign(nam, round(pca$values[x]*100/sum(pca$values[pca$values>=0]),2))
}

kmeans_res<-
  kmeans(as.matrix(mds.cor$points[,1]),
                   c(min(mds.cor$points[,1]),
                     median(mds.cor$points[,1]),
                     max(mds.cor$points[,1])))
k_ss<-
  round(kmeans_res$betweenss/kmeans_res$totss,3)

k <- as.data.frame(kmeans_res$cluster)
colnames(k) <- "k"

pca.mat <-
  as.data.frame(pca.mat)

pca.mat$pop <-
  label$V3
```

```
pca.mat$sample_name <-
  label$V1

pca.out <-
  pca.mat[,c(1:4)]

clusters <-
  cbind(label, pca.out) %>%
  select(-V3, -V4) %>%
  cbind(mds.cor$points) %>%
  cbind(k) %>%
  rename(id=V1) %>%
  rename(sample_name=V2) %>%
  rename(subregion=V5) %>%
  left_join(geneticsex) %>%
  left_join(df2, by="sample_name")
```

Plot PC2 vs PC1 and visually determine how many clusters

```
mds_plot <- clusters %>%
  distinct(id, .keep_all = T) %>%
  ggplot(aes(y=Dim.2, x=Dim.1, col=as.factor(k)))+
  geom_point()+
  theme_minimal() +
  scale_color_manual(values = c("#264653", "#2a9d8f", "#e9c46a"))+
  theme(
    panel.border = element_blank(),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.text.x = element_text(hjust=1, size=text_size),
    axis.text.y = element_text(size = text_size),
    axis.title = element_text(size = text_size),
    legend.position = "right",
    legend.text = element_text(size=11),
    legend.title = element_text())+
  labs(x="\nMDS1", y="MDS2\n",
       title=paste0("Outlier window in ", gsub("\\_.*","",params$beagle), "\n"),
       subtitle="Multidimensional Scaling\n")+
  guides(color=guide_legend(title="Cluster"))

pca_plot <- clusters %>%
  distinct(id, .keep_all = T) %>%
  ggplot(aes(y=PC2, x=PC1, col=as.factor(k)))+
  geom_point()+
  theme_minimal() +
  scale_color_manual(values = c("#264653", "#2a9d8f", "#e9c46a"))+
  theme(
    panel.border = element_blank(),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.text.x = element_text(hjust=1, size=text_size),
    axis.text.y = element_text(size = text_size),
    axis.title = element_text(size = text_size),
    legend.position = "right",
```

```
    legend.text = element_text(size=11),
    legend.title = element_text())+
  labs(x="\nPC1", y="PC2\n",
       subtitle="PCA\n")+
  guides(color=guide_legend(title="Cluster"))

plot_mds_pca <-
  mds_plot+pca_plot+plot_layout(guides = "collect")

ggsave(
  plot_mds_pca,
  filename = file.path(reports_path,
                       "plots",
                       paste0("mds_vs_pca_",params$beagle,".pdf")),
  device = "pdf",width = 10, height=5
)
```

```
plot_mds_pca
```

```
df <-
  clusters %>%
  group_by(pop, geno, k, region) %>%
  summarise(n=n())

regions <-
  df %>%
  distinct(region) %>%
  drop_na()

for (i in as.vector(regions$region)) {
  #sum all genotypes by region
  total_geno <- df %>%
    group_by(region) %>%
    drop_na() %>%
    summarise(total_n = sum(n))
  #sum all genotypes containing allele B
  total_b <- df %>%
    filter(grepl("B", geno)) %>%
    group_by(region) %>%
    drop_na() %>%
    summarise(total_b = sum(n))
  #merge dataset
  summary_alleles_b <-
    full_join(total_geno, total_b)

}

summary_alleles_b$percentage_b <- summary_alleles_b[,3]/summary_alleles_b[,2]
summary_alleles_b

regions <-
  df %>%
  distinct(pop) %>%
  drop_na()
```

```
for (i in as.vector(regions$pop)) {
  #sum all genotypes by region
  total_geno <- df %>%
    group_by(pop) %>%
    drop_na() %>%
    summarise(total_n = sum(n))
  #sum all genotypes containing allele B
  total_cluster1 <- df %>%
    filter(grepl("1", as.factor(k))) %>%
    group_by(pop) %>%
    drop_na() %>%
    summarise(total_cluster1 = sum(n))
  total_cluster2 <- df %>%
    filter(grepl("2", as.factor(k))) %>%
    group_by(pop) %>%
    drop_na() %>%
    summarise(total_cluster2 = sum(n))
  total_cluster3 <- df %>%
    filter(grepl("3", as.factor(k))) %>%
    group_by(pop) %>%
    drop_na() %>%
    summarise(total_cluster3 = sum(n))
  #merge dataset
  summary_alleles_cluster <-
    full_join(total_geno, total_cluster1) %>%
    full_join(total_cluster2) %>%
    full_join(total_cluster3)

}

summary_alleles_cluster$percentage_cluster1 <-
  summary_alleles_cluster[,3]/summary_alleles_cluster[,2]

summary_alleles_cluster$percentage_cluster2 <-
  summary_alleles_cluster[,4]/summary_alleles_cluster[,2]

summary_alleles_cluster$percentage_cluster3 <-
  summary_alleles_cluster[,5]/summary_alleles_cluster[,2]

summary_alleles_cluster

df_map <-
  summary_alleles_cluster %>%
  left_join(pop_info, by="pop") %>%
  distinct(pop, .keep_all = T) %>%
  select(-sample_name)

df_map$percentage_cluster1 <-
  as.numeric(unlist(df_map$percentage_cluster1))
df_map$percentage_cluster2 <-
  as.numeric(unlist(df_map$percentage_cluster2))
df_map$percentage_cluster3 <-
  as.numeric(unlist(df_map$percentage_cluster3))
```

```r
write_csv(df_map,
          file.path(reports_path,
                    "localPCA",
                    "outlier_mds",
                    paste0(params$beagle,"_summary_alleles_cluster.csv")))
```

```r
df_map[is.na(df_map)] <- 0

oceania <-
  rnaturalearth::ne_countries(scale = "large",
                returnclass = "sf",
                continent = "oceania")

anzo <- ggplot(data = oceania) +
    geom_sf(fill = "darkgrey", color = NA) +
    coord_sf(xlim = c(140, 187),
             ylim = c(-50,-12),
             expand = FALSE) +
    theme(
      text=element_text(),
      panel.background = element_rect(fill = background_colour),
      panel.border = element_blank(),
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      axis.text.x = element_blank(),
      axis.text.y = element_blank(),
      axis.title.x = element_blank(),
      axis.title.y = element_blank(),
      axis.ticks = element_blank(),
      # legend.position = "none",
      plot.margin = margin(0.2, 0.2, 0.2, 0.2, "cm"))+
    geom_scatterpie(aes(x=longitude+0.5, y=latitude+0.5, group = pop, r = 1),
                    data = df_map, cols=c("percentage_cluster1",
                                          "percentage_cluster2",
                                          "percentage_cluster3"), color=NA) +
    scale_fill_manual(values = c("#264653", "#2a9d8f", "#e9c46a"))

ggplot(data = oceania) +
    geom_sf(fill = "darkgrey", color = NA) +
    coord_sf(xlim = c(-180, -173),
             ylim = c(-47,-40),
             expand = FALSE) +
    theme(
      text=element_text(),
      panel.background = element_rect(fill = background_colour),
      panel.border = element_blank(),
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      axis.text.x = element_blank(),
      axis.text.y = element_blank(),
      axis.title.x = element_blank(),
      axis.title.y = element_blank(),
      axis.ticks = element_blank(),
```

```
      # legend.position = "none",
      plot.margin = margin(0.2, 0.2, 0.2, 0.2, "cm"))+
    geom_scatterpie(aes(x=longitude+0.5, y=latitude+0.5, group = pop, r = 1),
                    data = df_map, cols=c("percentage_cluster1",
                                          "percentage_cluster2",
                                          "percentage_cluster3"), color=NA) +
    scale_fill_manual(values = c("#264653", "#2a9d8f", "#e9c46a"))

ggplot(data = oceania) +
    geom_sf(fill = "darkgrey", color = NA) +
    coord_sf(xlim = c(-155, -143),
             ylim = c(-20,-14),
             expand = FALSE) +
    theme(
      text=element_text(),
      panel.background = element_rect(fill = background_colour),
      panel.border = element_blank(),
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      axis.text.x = element_blank(),
      axis.text.y = element_blank(),
      axis.title.x = element_blank(),
      axis.title.y = element_blank(),
      axis.ticks = element_blank(),
      # legend.position = "none",
      plot.margin = margin(0.2, 0.2, 0.2, 0.2, "cm"))+
    geom_scatterpie(aes(x=longitude+0.5, y=latitude+0.5, group = pop, r = 1),
                    data = df_map, cols=c("percentage_cluster1",
                                          "percentage_cluster2",
                                          "percentage_cluster3"), color=NA) +
    scale_fill_manual(values = c("#264653", "#2a9d8f", "#e9c46a"))

van_sm <- ggplot(data = oceania) +
    geom_sf(fill = "darkgrey", color = NA) +
    coord_sf(xlim = c(160, 172),
             ylim = c(-25,-12),
             expand = FALSE) +
    theme(
      text=element_text(),
      panel.background = element_rect(fill = background_colour),
      panel.border = element_blank(),
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      axis.text.x = element_blank(),
      axis.text.y = element_blank(),
      axis.title.x = element_blank(),
      axis.title.y = element_blank(),
      axis.ticks = element_blank(),
      # legend.position = "none",
      plot.margin = margin(0.2, 0.2, 0.2, 0.2, "cm"))+
    geom_scatterpie(aes(x=longitude+0.5, y=latitude+0.5, group = pop, r = 0.3),
                    data = df_map, cols=c("percentage_cluster1",
                                          "percentage_cluster2",
```

```
                                          "percentage_cluster3"), color=NA) +
    scale_fill_manual(values = c("#264653", "#2a9d8f", "#e9c46a"))

chartpie <-
  anzo+van_sm+plot_layout(guides="collect")

chartpie

ggsave(
  chartpie,
  filename = file.path(reports_path,
                       "plots",
                       paste0("chartpie_",params$beagle,".pdf")),
  device = "pdf",width = 10, height=5
)
```