scTrio-seq2

RNA part

▼ Preprocess

1) merge umi_count.xls of batches into one umi_count.xls

```
count_dir <- "F:/Project/3P/data/RNA/umi_count/raw_data_count/"
save_count_dir <- "F:/Project/3P/data/RNA/umi_count/combined_count/MRT_UMI_count.txt"

merge_count_func(indir = count_dir, outdir=save_count_dir, save_res = T)
```

2) create seurat object with metadata

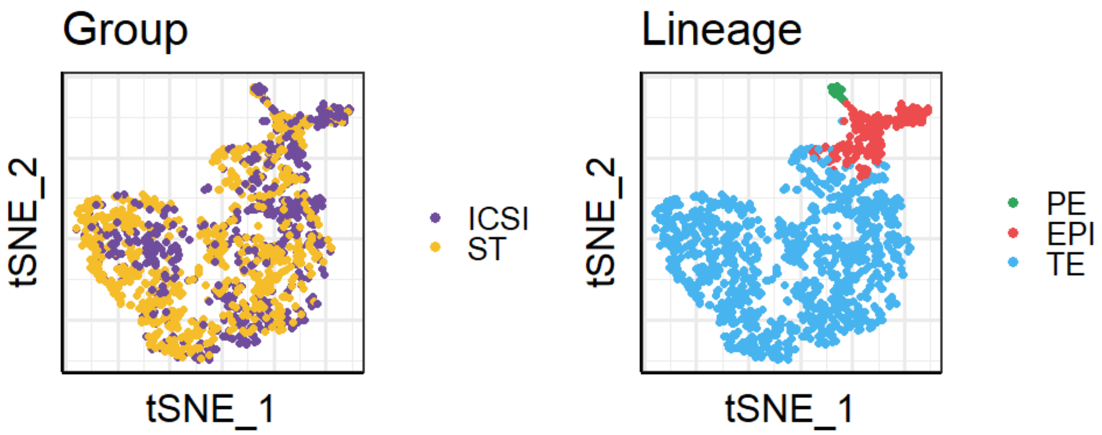
```
save_count_dir <- "F:/Project/3P/data/RNA/umi_count/combined_count/MRT_UMI_count.txt"
save_seurat_dir <- paste0("F:/Project/3P/R_workspace/ST_", Sys.Date(), "_seurat.rda")
infodir <- "F:/Project/3P/data/StatInfo/SampleInfo.txt"

ST.seurat <- seurat_pipeline(
  umi_count_dir = save_count_dir,
  save_path = save_seurat_dir,
  info_dir = infodir,
  min_gene=min_gene_num,min_cell=3
)
ST.seurat <- NormalizeData(ST.seurat, normalization.method = "LogNormalize", scale.factor = 1e+5, verbose = F)
ST.seurat <- FindVariableFeatures(ST.seurat, selection.method = "vst")
ST.seurat <- ScaleData(ST.seurat, features = rownames(ST.seurat))
ST.seurat <- RunPCA(ST.seurat,features = c(PE_marker, EPI_marker,TE_marker))
ST.seurat <- RunTSNE(ST.seurat,dims = 1:5)
```

3）load colorList

```
load("F:/Project/3P/R_workspace/colors.rda")
MRT_colorlist$Group_col["ST"] <- "#f5bd2a"
MRT_colorlist$Group_col["ICSI"] <- "#704d9c"
```

▼ fig.1.b

```
plot_outdir <- "F:/Project/3P/plot/"
pdf(paste0(plot_outdir , "MRT_", min_gene_num, ".reduce_dimension.pdf"))

reduction <- "tsne"
coor_ratio <- calc_coord_ratio(seurat.obj = ST.seurat, reduction_method = reduction)
p1 <- DimPlot(ST.seurat, reduction = "tsne",
        pt.size = 2,
        cols = MRT_colorlist$Group_col,
        group.by = "Group") +
  theme_bw(base_size = 25)+
  theme(axis.line = element_line(size = 1),
        axis.ticks = element_blank(),
        axis.text = element_blank(),
        legend.position = "right")+
  coord_equal(ratio = coor_ratio)

p2 <- DimPlot(ST.seurat, reduction = "tsne",
             pt.size = 2,
             cols = MRT_colorlist$Lineage_col,
             group.by = "Lineage") +
  theme_bw(base_size = 25) +
  theme(axis.line = element_line(size = 1),
        axis.ticks = element_blank(),
        axis.text = element_blank(),
        legend.position = "right")+
  coord_equal(ratio = coor_ratio)
print(p1+p2)

dev.off()
```

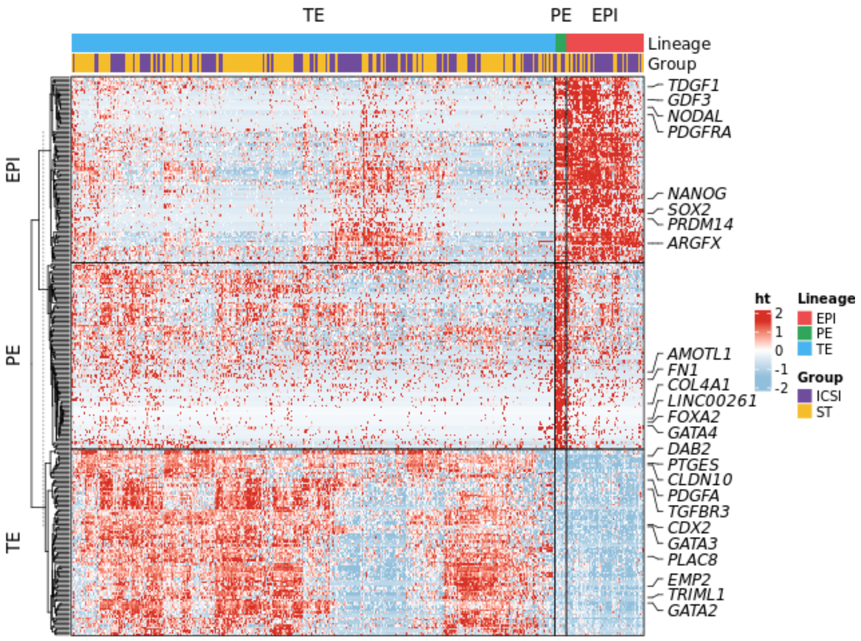


▼ tbl.1

```
## newly defined lineage markers
Idents(ST.seurat) <- ST.seurat$Lineage
levels(ST.seurat)
new_lineage_marker <- FindAllMarkers(ST.seurat, test.use = "wilcox")
new_lineage_marker <- new_lineage_marker[new_lineage_marker$avg_log2FC >0,]
MRT_lineage_marker <- new_lineage_marker[!duplicated(new_lineage_marker$gene),]
write.table(MRT_lineage_marker,file = paste0(plot_outdir, "MRT_", min_gene_num, ".lineage_marker.txt"),
            quote = F, sep = "\t",row.names = F, col.names = T)
```

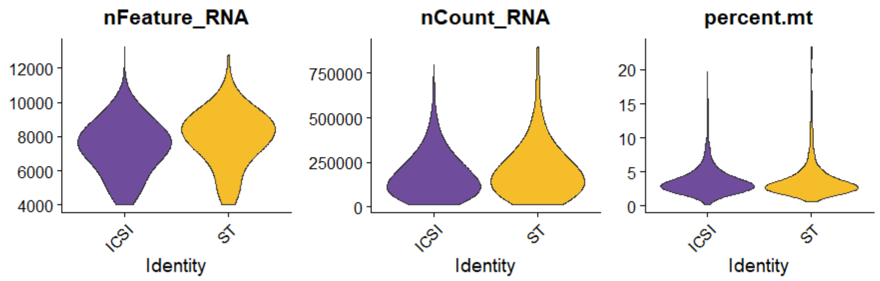
▼ fig.1c

```
MRT_lineage_marker <- read.table(file = paste0(plot_outdir, "MRT_", min_gene_num, ".lineage_marker.txt"),
                                 header = T, stringsAsFactors = F)
library(dplyr)
test <- MRT_lineage_marker %>%
  filter(p_val_adj < 0.05 & pct.1 >.5 ) %>%
  select(avg_log2FC, cluster, gene) %>%
  group_by(cluster) %>%
  slice_max(n = 100, order_by=avg_log2FC) %>%
  as.data.frame()

library(ComplexHeatmap)
library(circlize)
expr_scale <- ST.seurat[["RNA"]]@scale.data[test$gene,]

limit <- 1.5
expr_scale[expr_scale> limit] <- limit
expr_scale[expr_scale< -limit] <- (-limit)

anno_col <- data.frame(Lineage=ST.seurat$Lineage)
rownames(anno_col) <- rownames(ST.seurat@meta.data)

anno_colors <- list(Lineage=MRT_colorlist$Lineage_col)

col_anno <- HeatmapAnnotation(
  Lineage=ST.seurat$Lineage,
  Group=ST.seurat$Group,
  col = list(Lineage=MRT_colorlist$Lineage_col,
             Group=MRT_colorlist$Group_col)
)
row_anno <- rowAnnotation(
  foo = anno_mark(at = which(test$gene %in% c(TE_marker, EPI_marker, PE_marker)),
                  labels = test$gene[test$gene %in% c(TE_marker, EPI_marker, PE_marker)],
                  labels_gp = gpar(fontface="italic"))
)

colors_func <- colorRamp2(
  c(min(expr_scale, na.rm = T),
    mean(expr_scale, na.rm = T),
    max(expr_scale, na.rm = T)),
  c("#91BFDB", "white", "#D73027")
)

ht <- ComplexHeatmap::Heatmap(
  expr_scale,
  show_row_names = F,
  show_column_names = F,
  row_names_gp = gpar(fontface="italic"),
  column_split = factor(anno_col$Lineage,
                        levels = c("TE", "EPI", "PE")),
  column_gap = unit(0, "mm"),
  row_split = test$cluster, row_gap = unit(0, "mm"),
  border = TRUE, border_gp = gpar(lty = 1, lwd = 1),
  cluster_columns = T, cluster_column_slices = T,
  column_dend_height = unit(8, "cm"),
  cluster_row = T,
  cluster_row_slices = T,
  na_col = "lightgrey",
  top_annotation = col_anno,
  right_annotation = row_anno,
  show_row_dend = T,
  show_column_dend = F,
  col = colors_func,
  name = "ht",
  use_raster = T,
  raster_device="CairoPNG",
  raster_quality = 5
)
png(
  paste0(plot_outdir, "MRT_",
         min_gene_num,
         ".lineage_marker_enlarged.png"),
  width = 640, height = 480, units = "px"
)
draw(ht)
dev.off()
```



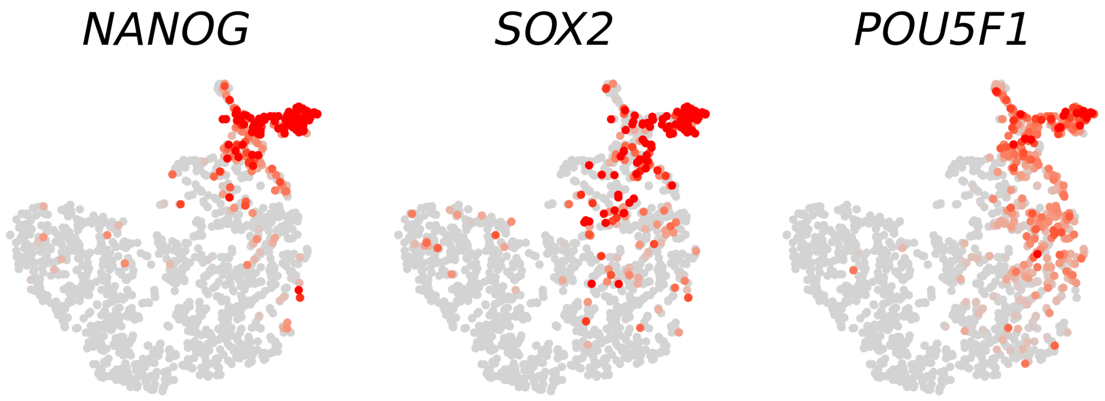
▼ sup.fig.1.b

```
VlnPlot(
    ST.seurat,
    features = c("nFeature_RNA","nCount_RNA","percent.mt"),
    group.by = "Group",
    cols = MRT_colorlist$Group_col,
    adjust = 2,pt.size = 0
)
```

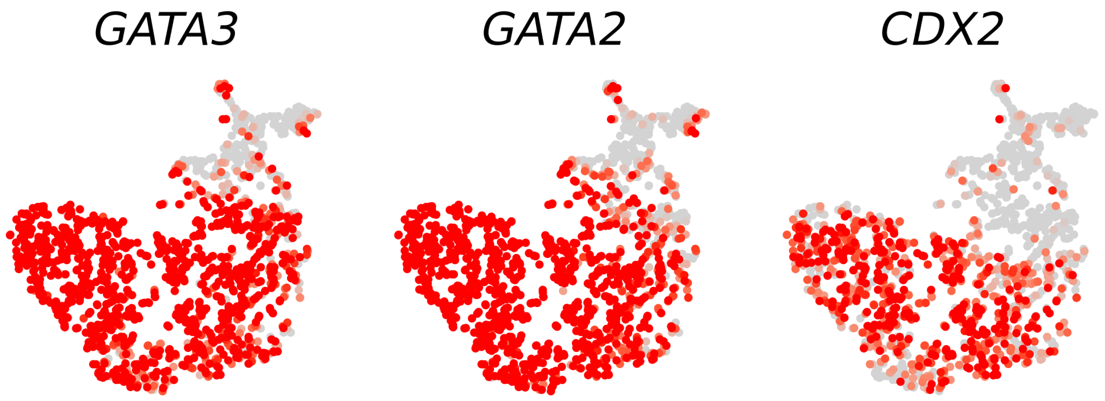


▼ sup.fig.1.c

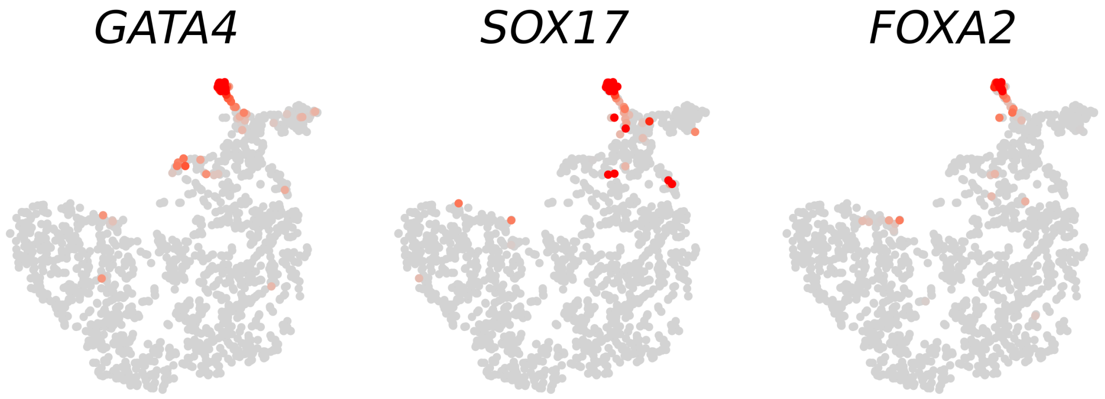
```
## EPI marker
p1 <- myfeature_plot(ST.seurat,markers = "NANOG",min_expr = 1.5,max_expr = 3)
p2 <- myfeature_plot(ST.seurat,markers = "SOX2",min_expr = 1,max_expr = 2)
p3 <- myfeature_plot(ST.seurat,markers = "POU5F1",min_expr = 4,max_expr = 5)
cowplot::plot_grid(p1,p2,p3,ncol = 3)
ggsave(paste0(plot_outdir, "MRT_", min_gene_num, ".EPI_marker_featureplot.pdf"))
```



```
## TE marker
p1 <- myfeature_plot(ST.seurat,markers = "GATA3",min_expr = 2,max_expr = 3)
p2 <- myfeature_plot(ST.seurat,markers = "GATA2",min_expr = 2,max_expr = 3)
p3 <- myfeature_plot(ST.seurat,markers = "CDX2",min_expr = 2,max_expr = 3)
cowplot::plot_grid(p1,p2,p3,ncol = 3)
ggsave(paste0(plot_outdir, "MRT_", min_gene_num, ".TE_marker_featureplot.pdf"))
```

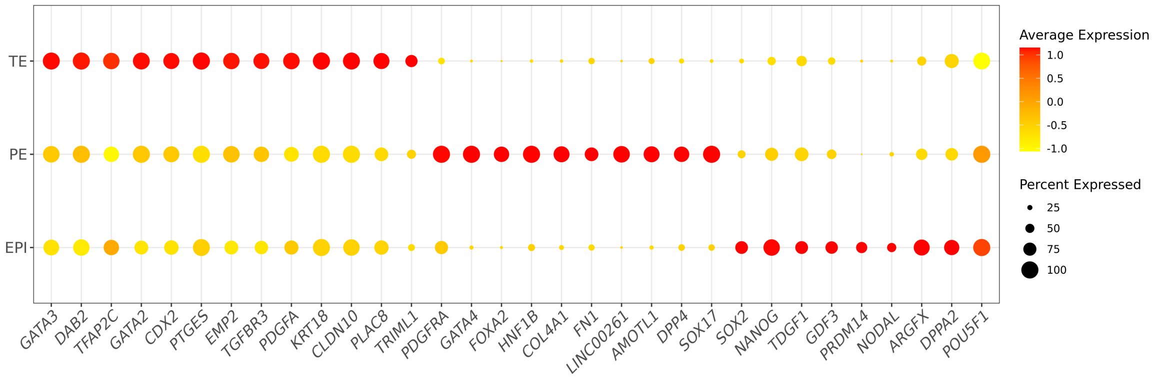


```
## PE marker
p1 <- myfeature_plot(ST.seurat,markers = "GATA4",min_expr = 1,max_expr = 3)
p2 <- myfeature_plot(ST.seurat,markers = "SOX17",min_expr = 2,max_expr = 3)
p3 <- myfeature_plot(ST.seurat,markers = "FOXA2",min_expr = 1,max_expr = 3)
cowplot::plot_grid(p1,p2,p3,ncol = 3)
ggsave(paste0(plot_outdir, "MRT_", min_gene_num, ".PE_marker_featureplot.pdf"))
```



▼ sup.fig.1.e

```
DotPlot(ST.seurat,scale=T,
        cols = c("yellow", "red"),
        features = c(TE_marker,PE_marker, EPI_marker),
        group.by = "Lineage")+
  theme_bw()+
  theme(axis.text.x = element_text(face = "italic", vjust = 1,
                                   hjust = 1, angle = 45),
        axis.title = element_blank(),
        axis.text = element_text(size = 12.5))
```



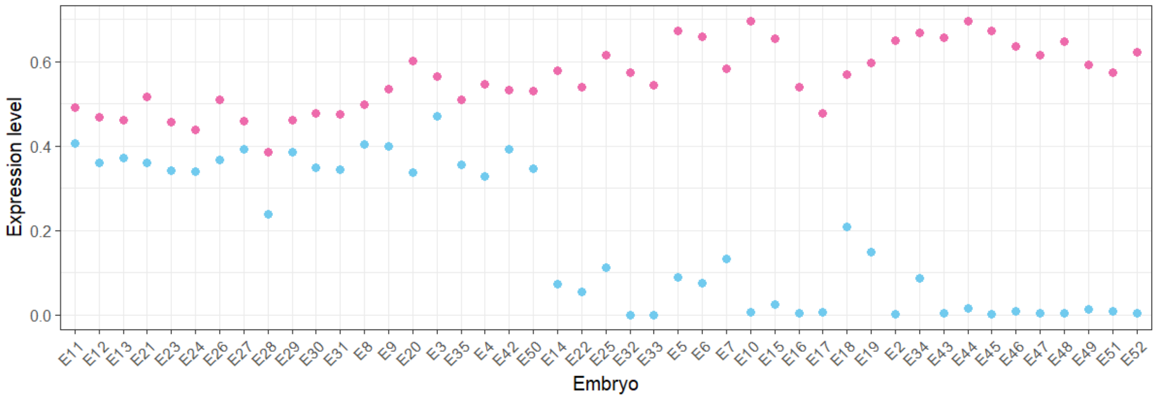
▼ sup.fig.2.b

1) infer embryonic sex by expression of X or Y linked genes

```
X_linked_genelist <- read.table("F:/Project/3P/reference/X_linked_genelist.txt",
                                header = T, stringsAsFactors = F)
Y_linked_genelist <- read.table("F:/Project/3P/reference/Y_linked_genelist.txt",
                                header = T, stringsAsFactors = F)

ST.seurat[["Gender"]] <- "unkown"
ST.seurat[["Mean_X"]] <- 0
ST.seurat[["Mean_Y"]] <- 0
X_gene <- intersect(X_linked_genelist$Gene_id, rownames(ST.seurat))
Y_gene <- intersect(Y_linked_genelist$Gene_id, rownames(ST.seurat))
for (i in names(table(ST.seurat$Embryo))) {
  X_mean <- mean(apply(ST.seurat@assays$RNA@data[X_gene,colnames(ST.seurat)[ST.seurat$Embryo==i]], 2, mean), na.rm = T)
  Y_mean <- mean(apply(ST.seurat@assays$RNA@data[Y_gene,colnames(ST.seurat)[ST.seurat$Embryo==i]], 2, mean), na.rm = T)
  if(X_mean/Y_mean > 2){
    ST.seurat$Gender[ST.seurat$Embryo==i] <- "Female"
  } else {
    ST.seurat$Gender[ST.seurat$Embryo==i] <- "Male"
  }
  ST.seurat$Mean_X[ST.seurat$Embryo==i] <- X_mean
  ST.seurat$Mean_Y[ST.seurat$Embryo==i] <- Y_mean
}
```

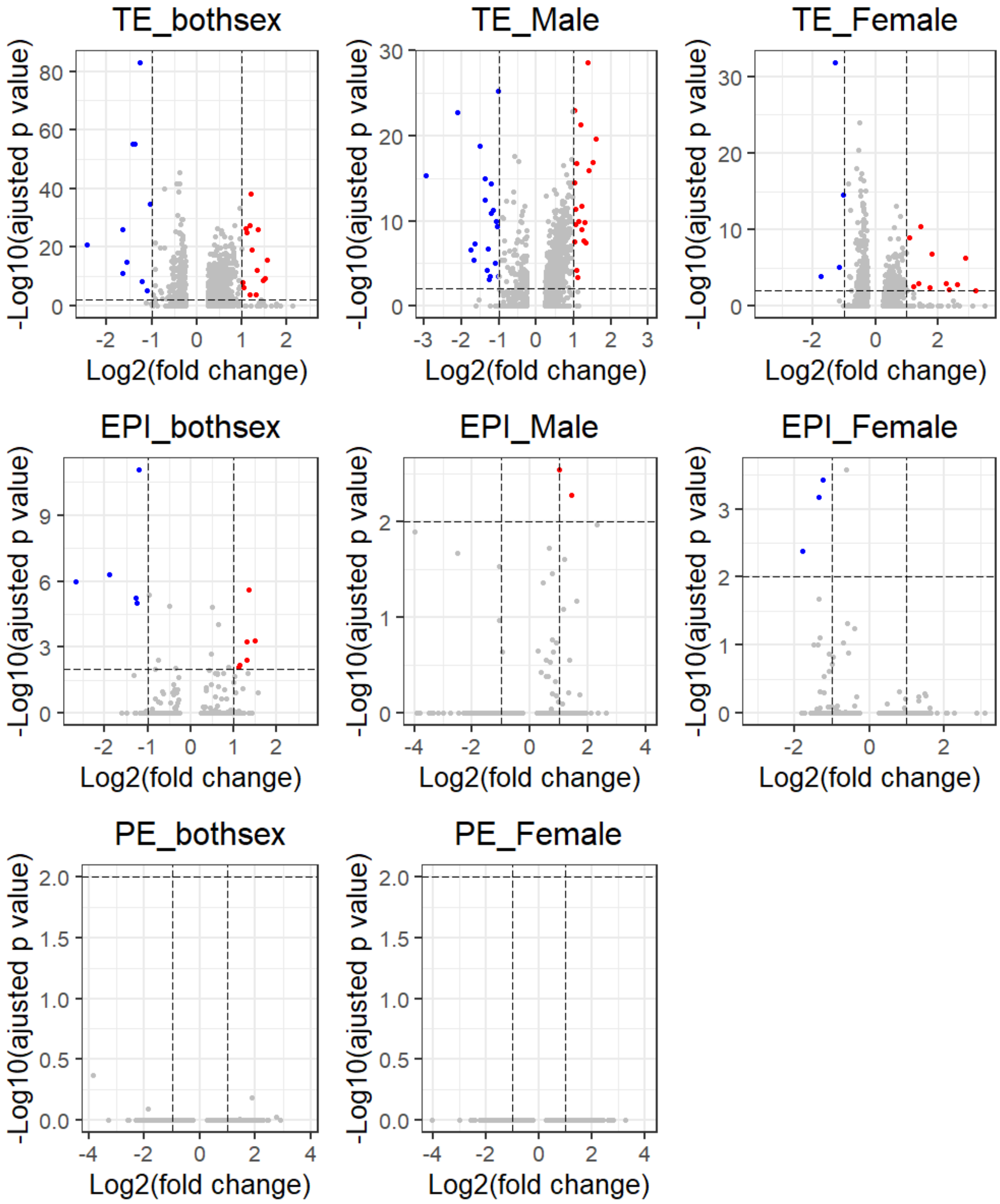
2) plot expression ratio

```
tmp <- unique(ST.seurat@meta.data[, c("Embryo", "Gender","Group", "Mean_X", "Mean_Y")])

library(dplyr)
tmp <- tmp %>% group_by(Gender)
tmp <- tmp %>% arrange(desc(Group),desc(Embryo), .by_group = T) %>% as.data.frame()
tmp$Embryo <- factor(tmp$Embryo, levels = rev(tmp$Embryo))

ggplot(data = tmp, aes(x=Embryo))+
  geom_point(aes(y=Mean_X), color=MRT_colorlist$Gender_col["Female"], size=3)+
  geom_point(aes(y=Mean_Y), color=MRT_colorlist$Gender_col["Male"], size=3)+
  labs(y="Expression level")+
  theme_bw(base_size = 15)+
  theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1))
```



▼ sup.fig.2a.c

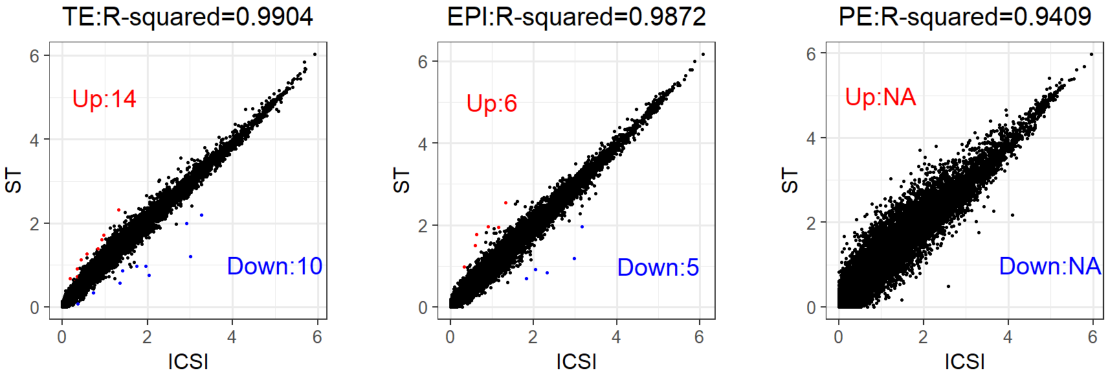
```
library(ggrepel)
# pdf(paste0(plot_outdir, "MRT_", min_gene_num, ".DEG_volcano.pdf"), width = 7,height = 7)
plot_list <- list()
for(lineage in c("TE", "EPI", "PE")){
  #gender <- "Female"
  for(gender in c("", "Female","Male")){
    cell_vector <- ST.seurat$Lineage==lineage &
      ST.seurat$Gender !=gender
    if(gender==""){
      plot_name <- paste0(lineage,"_bothsex")
    } else if(gender=="Female"){
      plot_name <- paste0(lineage, "_Male")
    } else if(gender=="Male"){
      plot_name <- paste0(lineage, "_Female")
    }
    if(length(table((ST.seurat[,cell_vector]$Group)))==1) next
    group_deg <- FindMarkers(
      ST.seurat[,cell_vector],
      group.by = "Group",
      ident.1 = "ST",
      ident.2 = "ICSI",
      min.pct = 0.25,
      test.use = "wilcox"
    )
    write.table(group_deg, file = paste0(plot_outdir, "MRT_", min_gene_num, ".DEG_",plot_name,".txt"),
                quote = F, sep = "\t", col.names = T, row.names = T)
    group_deg$`log10(adj.pvalue)` <- -log10(group_deg$p_val_adj)
    col_vec <- rep("grey", dim(group_deg)[1])
    col_vec[group_deg$avg_log2FC > 1 & group_deg$p_val_adj  < 0.01 ] <- "red"
    col_vec[group_deg$avg_log2FC < -1 & group_deg$p_val_adj  < 0.01 ] <- "blue"
    group_deg$color <- col_vec
    group_deg$label <- rownames(group_deg)
    group_deg[group_deg$color=="grey","label"] <- ""
    # ratio <- diff(range(group_deg$avg_log2FC))/diff(range(group_deg$`log10(adj.pvalue)`))

    ## volcano plot
    plot_list[[plot_name]] <- ggplot()+
      geom_point(data = group_deg, aes(x=avg_log2FC, y=`log10(adj.pvalue)`, color=color))+
      scale_color_manual(values = c(blue="blue", grey="grey", red="red"))+
      geom_hline(yintercept = 2, linetype="longdash")+
      geom_vline(xintercept = c(-1, 1), linetype="longdash")+
      # geom_text_repel(data = group_deg, aes(x=avg_log2FC,
      #                                       y=`log10(adj.pvalue)`,
      #                                       color=color,
      #                                       label=group_deg[,"label"]),
      #                 xlim = c(-max(abs(range(group_deg$avg_log2FC))),
      #                          max(abs(range(group_deg$avg_log2FC)))),
      #                 fontface="italic",
      #                 color="black",
      #                 max.overlaps = 10)+
      labs(x="Log2(fold change)",
           title = plot_name,
           y="-Log10(ajusted p value)")+
      xlim(-max(abs(range(group_deg$avg_log2FC))),
           max(abs(range(group_deg$avg_log2FC))))+
      theme_bw(base_size = 20)+
      theme(#axis.text = element_text(size = 15),
        #axis.title = element_text(size = 15),
        legend.position = "none",
        plot.title = element_text(vjust = 1, hjust = .5))
    # print(p)
    plot_outdir <- "F:/Project/3P/plot/MRT"
    # ggsave(paste(plot_outdir, min_gene_num,plot_name, "DEG_volcano.png", sep = "."), width = 6, height = 7)
    # n <- n+1

    out_deg <- group_deg[group_deg$color!="grey",]
    if(nrow(out_deg)==0) next
    else out_deg$gene <- rownames(out_deg)
    # write.table(out_deg, file = paste0(plot_outdir,"_", min_gene_num, ".DEG_",plot_name,".txt"),
    #              quote = F, sep = "\t", col.names = T, row.names = F)
  }
}
# dev.off()

library(cowplot)
do.call("plot_grid", plot_list)
```



▼ fig.2.a

```
data <- as.data.frame(t(FetchData(object = ST.seurat, slot = "data",vars = rownames(ST.seurat))))

# pdf(file = paste0(plot_outdir, "MRT_", min_gene_num, ".regression_pearson.pdf"))

plot_list <- list()
```

```
for(lineage in c("TE", "EPI", "PE")){
  cor_df <- data.frame(ST=apply(data[,ST.seurat$Lineage==lineage &
                                       ST.seurat$Group=="ST"], 1, mean),
                       ICSI=apply(data[,ST.seurat$Lineage==lineage &
                                         ST.seurat$Group=="ICSI"], 1, mean))
  model.lm <- lm(ICSI ~ ST, data = cor_df)
  summary(model.lm)

  cor_df$Gene_id <- rownames(cor_df)
  deg_df <- read.table(
    paste0(plot_outdir, "_", min_gene_num, ".DEG_",lineage,"_bothsex.txt"),
    header = T,
    row.names = 1,
    stringsAsFactors = F
  )
  deg_df$Gene_id <- rownames(deg_df)
  # colnames(deg_df) <- sub("gene", "Gene_id", colnames(deg_df))
  cor_df <- merge(cor_df, deg_df, by = "Gene_id", all.x = T)

  col_vec <- rep("grey", dim(cor_df)[1])
  col_vec[cor_df$avg_log2FC > 1 & cor_df$p_val_adj  < 0.01 ] <- "red"
  col_vec[cor_df$avg_log2FC < -1 & cor_df$p_val_adj  < 0.01 ] <- "blue"
  cor_df$color <- col_vec
  cor_df$label <- cor_df$Gene_id
  cor_df[cor_df$color=="grey","label"] <- ""

  cor_ratio <- diff(range(cor_df$ICSI, na.rm = T))/diff(range(cor_df$ST, na.rm = T))
  plot_list[[lineage]] <- ggplot(data = cor_df, aes(x=ICSI, y=ST, color=color))+
    geom_point()+
    scale_colour_manual(values = c("grey"="black", "red"="red", "blue"="blue"))+
    annotate("text", label = paste0("Up:",as.numeric(table(cor_df$color)["red"])),
             x = 1, y = 5, size = 10, colour = "red")+
    annotate("text", label = paste0("Down:",as.numeric(table(cor_df$color)["blue"])),
             x = 5, y = 1, size = 10, colour = "blue")+
    coord_equal(ratio = cor_ratio)+
    labs(title = paste0(lineage, ":","R-squared=",
                        format(summary(model.lm)$r.squared, digits = 4)))+
    theme_bw(base_size = 25)+
    theme(panel.border = element_rect(size = 1),
          #panel.grid = element_blank(),
          plot.title = element_text(hjust = .5),
          axis.ticks = element_line(size = 1),
          legend.key.size=unit(1,'cm'),
          legend.position = "none")
  # print(p)
  # ggsave(file = paste0(plot_outdir, "_",min_gene_num, ".",lineage,".regression_pearson.png"))
}
# dev.off()

library(cowplot)
do.call("plot_grid", c(plot_list, ncol=3))
```



▼ fig.2.b

1) prepare input.csv of SCENIC in R

```
write.csv(t(as.matrix(ST.seurat@assays$RNA@counts)),
          file = paste0(plot_outdir, "MRT_", min_gene_num, ".rawcount.csv"))
```

2) prepare input.loom of SCENIC in python

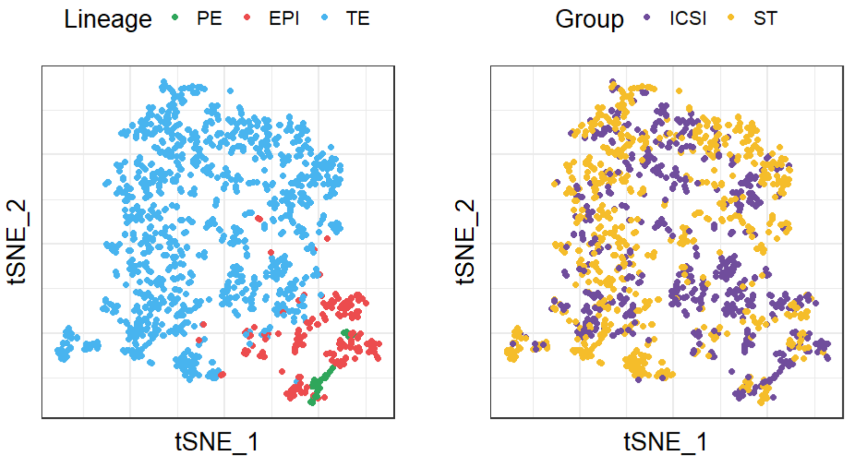
```
import loompy as lp
import numpy as np
import scanpy as sc
x=sc.read_csv("../../code/sample.csv")
row_attrs = {
  "Gene": np.array(x.var_names),
}
col_attrs = {
  "CellID": np.array(x.obs_names)
}
lp.create("MRT_4000.loom",x.X.transpose(),row_attrs,
          col_attrs)
```

3) run pySCENIC (0.11.0) in the CLS

```
#!/bin/bash
home=/gpfs1/tangfuchou_pkuhpc/tangfuchou_coe/xuexiaohui
script=$home/script/pipeline/SCENIC_xxh/pyscenic_pipeline.sh
dir=$home/project/MRT/Trio_RNA/SCENIC
prefix=MRT_4000
source $script $dir $prefix

do_01_grn
do_02_ctx
do_03_aucell
```

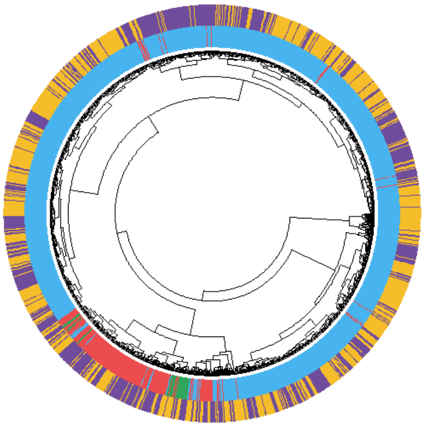
4) plot ouptut of pySCENIC in R

```
scenicLoomPath <- paste0(plot_outdir, "MRT_", min_gene_num, ".auc_mtx.loom")
loom <- open_loom(scenicLoomPath)
regulonsAUC <- get_regulons_AUC(loom,column.attr.name='RegulonsAUC')
regulon_res <- getAUC(regulonsAUC)

set.seed(0)
tsne_out <- Rtsne(t(regulon_res),
                  dims = 2,pca =T,
                  perplexity = 10,
                  theta = 0,
                  check_duplicates = FALSE)
rownames(tsne_out$Y) <- colnames(regulon_res)
ST.seurat[["scenic_tSNE_1"]] <- tsne_out$Y[,1]
ST.seurat[["scenic_tSNE_2"]] <- tsne_out$Y[,2]

ratio <- diff(range(ST.seurat$scenic_tSNE_1))/diff(range(ST.seurat$scenic_tSNE_2))

## lineage
p1 <- ggplot(data = ST.seurat@meta.data,
       aes(x=scenic_tSNE_1, y=scenic_tSNE_2,color=Lineage))+
  geom_point(size=2)+
  scale_color_manual(values = MRT_colorlist$Lineage_col)+
  coord_equal(ratio = ratio)+
  labs(x="tSNE_1", y="tSNE_2")+
  theme_bw(base_size = 20)+
  theme(legend.position = "top",
        axis.ticks = element_blank(),
        axis.text = element_blank())

p2 <- ggplot(data = ST.seurat@meta.data,
       aes(x=scenic_tSNE_1, y=scenic_tSNE_2,color=Group))+
  geom_point(size=2)+
  scale_color_manual(values = MRT_colorlist$Group_col)+
  coord_equal(ratio = ratio)+
  labs(x="tSNE_1", y="tSNE_2")+
  theme_bw(base_size = 20)+
  theme(legend.position = "top",
        axis.ticks = element_blank(),
        axis.text = element_blank())

cowplot::plot_grid(p1,p2, ncol = 2)
```



▼ fig.2.c

```
anno_col <- data.frame(Group=ST.seurat$Group,
                       Lineage=ST.seurat$Lineage)
rownames(anno_col) <- rownames(ST.seurat@meta.data)

anno_colors <- list(
  Lineage=MRT_colorlist$Lineage_col,
  Group=MRT_colorlist$Group_col
)

library(pheatmap)
out.hclust <-pheatmap::pheatmap(regulon_res,
                                cluster_rows = F, cluster_cols = T,
                                annotation_col = anno_col,
                                show_colnames = F,show_rownames = F,
                                annotation_colors = anno_colors,
                                annotation_names_row=F,
                                annotation_names_col = T,
                                scale = "none",silent = F,
                                clustering_method = "ward.D2")
circos_hclust(info = ST.seurat@meta.data,
              hclust.out = out.hclust,
              col_list = MRT_colorlist,
              levels= c("Group", "Lineage"))
```



▼ fig.4.a

```
infer_result_ref <- read.table("F:/Project/3P/plot/infercnv.references.txt",header = T, row.names = 1, stringsAsFactors = F)
infer_result_obs <- read.table("F:/Project/3P/plot/infercnv.observations.txt", header = T, row.names = 1, stringsAsFactors = F)
infer_result_total <- cbind(infer_result_obs, infer_result_ref)
rm(infer_result_ref)
rm(infer_result_obs)

## gene_metadata
setwd(dir)
gene_bed <- read.table("F:/Project/3P/plot/infercnv.gene_order.txt", header = F, stringsAsFactors = F)
colnames(gene_bed) <- c("gene_symbol", "chr", "start", "end")
rownames(gene_bed) <- gene_bed$gene_symbol
gene_bed <- gene_bed[rownames(infer_result_total),]
gene_bed <- gene_bed %>% group_by(chr)
gene_bed$chr <- factor(gene_bed$chr, levels = c(paste0("chr", 1:22), "chrX", "chrY", "chrMT"))
gene_bed <- gene_bed %>% arrange(start, .by_group=T)
gene_bed <- as.data.frame(gene_bed)
gene_bed <- gene_bed[!gene_bed$chr %in% c("chrX", "chrY"),]
infer_result <- infer_result_total[gene_bed$gene_symbol,]

## cell_metadata
embryo_list <- unique(ST.seurat$Embryo)

inferCNV_stat <- stat_CNV(
  infer_result_total,
  embryo_list = embryo_list,
  cell_info = ST.seurat@meta.data
)
# inferCNV_stat$embryo["E11","chr8"]<- -1
# inferCNV_stat$embryo["E45",c("chr6", "chr8", "chr15")]<- 1
# inferCNV_stat$sc[info$Cell_id[info$Embryo=="E11"],"chr8"] <- -1
# inferCNV_stat$sc[info$Cell_id[info$Embryo=="E45"],c("chr6", "chr8", "chr15")] <- 1
inferCNV_stat$sc[info$Cell_id[info$Embryo=="E6"],"chr4"] <- -1
inferCNV_stat$sc[info$Cell_id[info$Embryo=="E6"],"chr9"] <- 1

## arrange chromosome by cell
inferCNV_sc <- inferCNV_stat$sc

inferCNV_sc[inferCNV_sc==0] <- 2
inferCNV_sc[inferCNV_sc==-1] <- 1.5
inferCNV_sc$Order <- apply(inferCNV_sc, 1, function(x){min(c(1:22)[which(x!=2)])})
inferCNV_sc <- inferCNV_sc %>% arrange(Order)
info_tmp <- unique(ST.seurat@meta.data[,c("Cell_id", "Lineage", "Group")])
inferCNV_sc$Cell_id <- rownames(inferCNV_sc)
inferCNV_sc <- merge(inferCNV_sc, info_tmp, by = "Cell_id")
inferCNV_sc <- inferCNV_sc %>% group_by(Group, Lineage) %>% arrange(Order, .by_group=T) %>% as.data.frame()
inferCNV_sc$Lineage <- factor(inferCNV_sc$Lineage, levels = c("EPI", "PE", "TE"))
inferCNV_sc$Group <- factor(inferCNV_sc$Group, levels = c("ICSI", "ST"))


## plot heatmap
chr_col <- rep(c("black", "white"),11)
names(chr_col) <- paste0("chr", 1:22)
col_anno <- HeatmapAnnotation(
  chr=paste0("chr", 1:22),
  col = list(chr=chr_col),
  show_annotation_name = F,
  border = T,
  show_legend = F
  )
row_anno <- rowAnnotation(
  Lineage=inferCNV_sc$Lineage,
  Group=inferCNV_sc$Group,
  col = list(Lineage=MRT_colorlist$Lineage_col,
             Group=MRT_colorlist$Group_col),
  show_annotation_name=F
)
ht <- Heatmap(
  as.matrix(inferCNV_sc[,2:23]),
  name="CNV",
  show_row_names = F,
  show_column_names = F,
  left_annotation = row_anno,
  top_annotation = col_anno,
  row_split = inferCNV_sc[,c("Lineage", "Group")],
  row_gap = unit(0, "mm"),
  row_title_rot = 0,
  cluster_rows=F,
  border = T,
  show_row_dend = F,
  cluster_columns=FALSE,
  column_split = factor(paste0("chr", 1:22), levels = paste0("chr", 1:22)),
  column_gap = unit(0, "mm"),
  column_title_rot = 90,
  cluster_row_slices = F,
  col=c("1"="red","2"="white", "1.5"="blue")
)
draw(ht)
```

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/aa29e0a6-2d2c-47c1-a535-5d4506d0db08/MRT_4000.inferCNV_sc_heatmap(corrected).pdf

▼ fig.4.b

```
ST.seurat[["inferCNV_CNV_count"]] <- 0
ST.seurat@meta.data[rownames(inferCNV_stat$sc),"inferCNV_CNV_count"] <- rowSums(abs(inferCNV_stat$sc))
ST.seurat[["inferCNV_ploidy_sc"]] <- "euploidy"
ST.seurat$inferCNV_ploidy_sc[ST.seurat$inferCNV_CNV_count!=0] <- "aneuploidy"

ploidy_count_embryo_per <- ST.seurat@meta.data %>%
  dplyr::select(Group, Embryo,inferCNV_ploidy_sc) %>%
  dplyr::group_by(Group, Embryo, inferCNV_ploidy_sc) %>%
  dplyr::summarise(Ploidy=n()) %>%
  dplyr::mutate(total_cell=sum(Ploidy),CNV_freq=round(Ploidy/sum(Ploidy)*100,2)) %>%
  dplyr::filter(total_cell >=30) %>%
  as.data.frame()

write.table(ploidy_count_embryo_per,
            file = paste0(outdir, "MRT_", min_gene_num, "ploidy_count_embryo_per.txt"),
            quote = F, sep = "\t",
            col.names = T, row.names = F)

ggplot(data = ploidy_count_embryo_per %>%
         filter(inferCNV_ploidy_sc=="aneuploidy"),
       aes(x=Group, y=CNV_freq, fill=Group))+
  geom_boxplot(width=.6, outlier.shape = NA)+
  labs(y="CNV frequency")+
  scale_fill_manual(values = MRT_colorlist$Group_col)+
  geom_signif(comparisons = list(c("ST", "ICSI")),
              test = "t.test",
              textsize = 10,
              y_position = 65)+
  theme_bw(base_size = 25)+
  theme(axis.title.x = element_blank())
```

DNA part

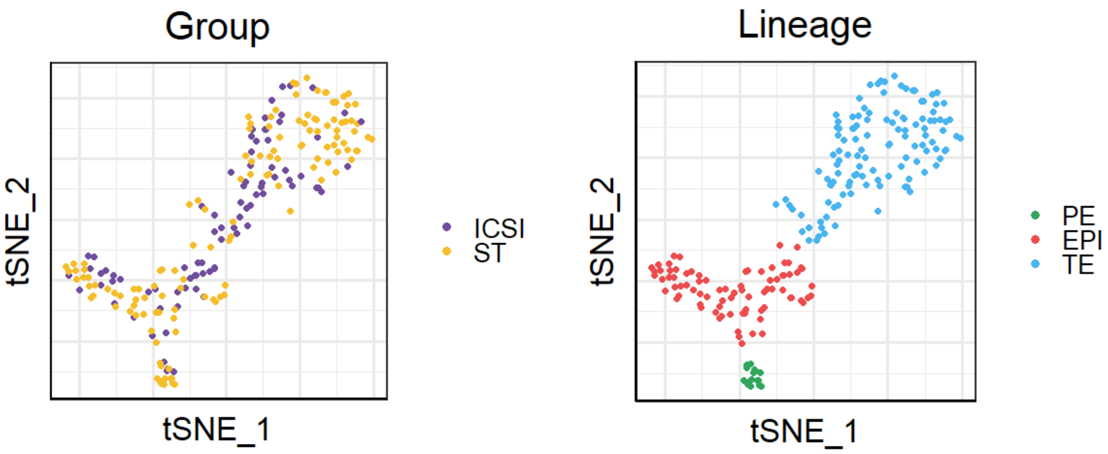
▼ Preprocess

```
plot_outdir <- "F:/Project/3P/plot/"
info_dir <- paste0(plot_outdir, "Meth_Total_SampleInfo.txt")

info_df <- read.table(info_dir, header = T, stringsAsFactors = F)
attach(info_df)
info_df<- info_df[
  CpG_TotalCpG.1X. > 2e+6 &
  Lambda_percent < 25 &
  Conversion_ratio > 99 &
    lambda_percent==10,]
detach()
```

▼ fig.3.a

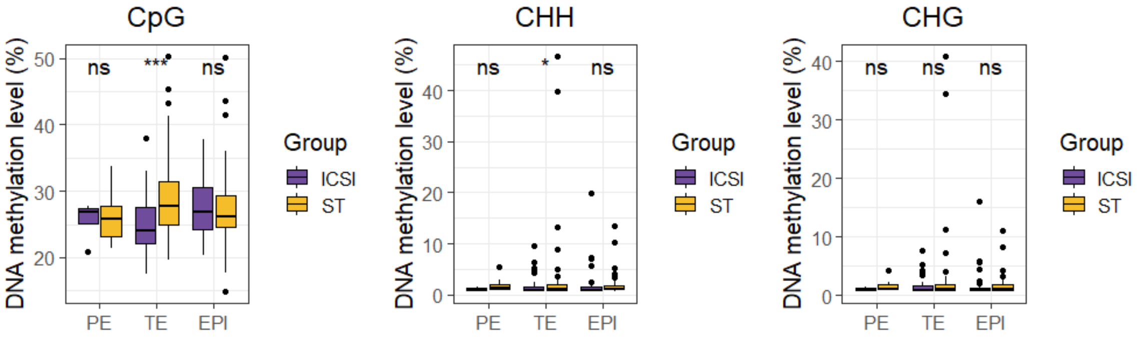
```
seurat_dir <- paste0(plot_outdir, "_", min_gene_num, ".seurat.rda")

load(seurat_dir)

meth.seurat <- subset(ST.seurat, subset = ID %in% info_df$Sample)
meth.seurat <-NormalizeData(meth.seurat, normalization.method = "LogNormalize", scale.factor = 1e+5, verbose = F)
meth.seurat <-FindVariableFeatures(meth.seurat, selection.method = "vst")
meth.seurat <- ScaleData(meth.seurat, features = rownames(meth.seurat), verbose = F)
meth.seurat <- RunPCA(meth.seurat,features = c(PE_marker, EPI_marker,TE_marker))
# DimHeatmap(meth.seurat, dims = 1:10, balanced = TRUE)
# ElbowPlot(meth.seurat)
```

```
meth.seurat <- RunTSNE(meth.seurat, dims = 1:5)

reduction <- "tsne"
plot_list <- list()
for (i in c("Group", "Lineage")) {
  p <- DimPlot(meth.seurat, reduction = reduction,
                pt.size = 2,
                # cols = MRT_colorlist[paste0(i, "_col")],
                group.by = i) +
    scale_color_manual(values = MRT_colorlist[[paste0(i, "_col")]])+
    theme_bw(base_size = 25) +
    theme(axis.line = element_line(size = 1),
          axis.ticks = element_blank(),
          axis.text = element_blank(),
          plot.title = element_text(hjust = .5),
          legend.position = "right")
  plot_list[[i]] <- p + coord_equal(ratio = diff(range(p$data$tSNE_1))/diff(range(p$data$tSNE_2)))
}

library(cowplot)
do.call("plot_grid", plot_list)
```



▼ fig.3.b

```
library(ggpubr)

group_var <- "Lineage"
output_type <- "Ratio"

plot_list <- list()
for(j in c("CpG","CHH", "CHG")){
  for(i in c(1)){
    if(output_type=="Ratio"){
      y <- paste0(j, "_MethRatio.", i, "X.")
      ylab <- "DNA methylation level (%)"
    } else if(output_type=="Total"){
      y <- paste0(j, "_Total",j,".",i,"X.")
      ylab <- "CpG site covered (1X)"
    }
    plot_list[[j]] <- ggboxplot(
       info_df,
       x = group_var,y = y,
       fill = "Group",
       # width = .8,
       # outlier.shape = NA,
       palette = MRT_colorlist$Group_col
      )+
      stat_compare_means(
         aes(group = Group),
         label="p.signif",
         size=5,vjust = 1
        )+
      labs(y=ylab,title=j)+
      theme_bw(base_size = 15)+
      theme(axis.title.x = element_blank(),
            plot.title = element_text(hjust = .5),
            strip.background = element_blank())
    # print(p)
  }
}

do.call("plot_grid", c(plot_list, ncol=3))
```



▼ fig.3.e

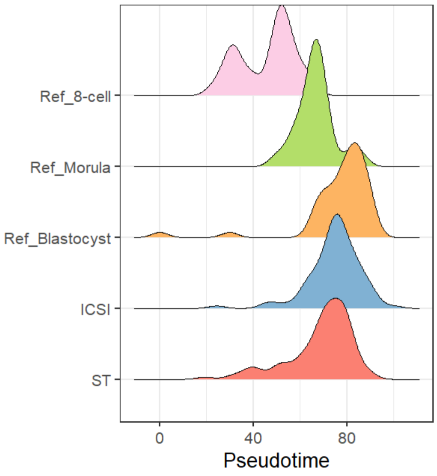
1) Analyze cells of 2017_NG the same way as this project, except that the length of random primer is 9bp

2) Perform 300bp-tiling on both CpG.singleC files of MRT and 2017_NG, and calculate the methylation level of tiles by C/(C+T), where C stands for the total number of Cs in the tile

3) Select cells passing quality control and merge their tile files into one matrix. Filter the tile position with less than 30% NAs among all the merged cells. Do PCA upon the filtered matrix.

4) Select PC1 as the pseudotime of CpG methylation.

```
library(scales)
library(ggridges)

## load ref info
ref_info <- read.table("F:/Project/3P/data/REF/2017_NG/2017_NG_StatInfo.txt",
                       header = T, stringsAsFactors = F)
ref_info$Group <- "Ref"
ref_info$Embryo <- "Ref"
ref_info$Stage <- ""
var_cols <- c("Sample", "Lineage", "Group", "Embryo","Stage")
merge_info <- rbind(merge_df[,var_cols], ref_info[,var_cols])

## load pca input
pca_input <- read.table("F:/Project/3P/Meth/GC_Merge_PCA_Result_300bp_per0.3.txt",
                        header = T, stringsAsFactors = F, row.names = 1)
pca_input$Sample <- rownames(pca_input)
pca_input <- merge(pca_input, merge_info, by = "Sample")
pca_input <- pca_input[pca_input$Lineage!="Blastocyst",]
pca_input <- pca_input[pca_input$Embryo !="E15",]
pca_input$Pseudotime <- rescale(pca_input$PC1, to=c(0,100)) ## rescale PC1 to 0~100

## rename group to Ref_8-cell, Ref_Morula, Ref_Blastcyst, ICSI, ST
pca_input$Group_1 <- pca_input$Group
selector <- pca_input$Lineage %in% c("8-cell", "Morula")
pca_input$Group_1[selector] <- paste0(pca_input$Group[selector], "_", pca_input$Lineage[selector])
pca_input$Group_1[pca_input$Group_1=="Ref"] <- paste0(pca_input$Group[pca_input$Group_1=="Ref"],"_Blastocyst")
ref_group <- paste0("Ref_", c("8-cell", "Morula", "Blastocyst"))
pca_input$Group_1 <- factor(pca_input$Group_1, levels = rev(c(ref_group, "ICSI", "ST")))

## ridge plot
ggplot(data = pca_input[pca_input$Embryo !="E15",],
            aes(x=Pseudotime, y=Group_1, fill=Group_1))+
  geom_density_ridges()+
  scale_fill_manual(values = brewer.pal(8,"Set3")[4:8])+
  theme_bw(base_size = 20)+
  theme(axis.title.y = element_blank(),
        legend.position = "none")

#1 cells from all the stages of MRT were used

#2 embryo E15 were excluded for its abnormal dimensional pattern
```



▼ sup.fig.3.4

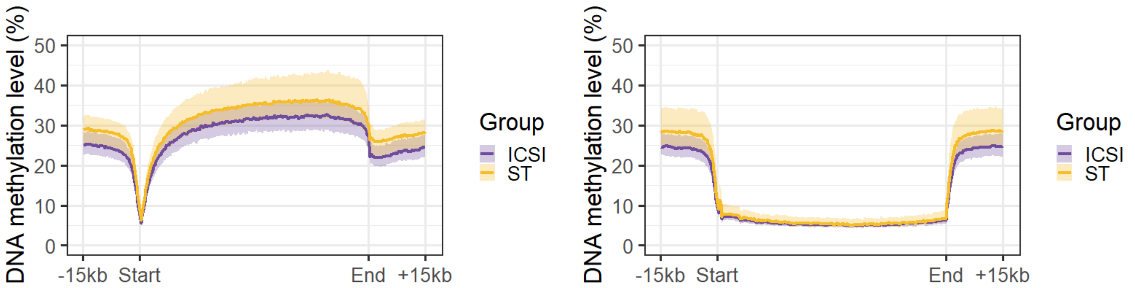
```
for (region in c("genebody", "CGI")) {
  if(region=="CGI"){
    x_label <- c("-15kb","Start", "End", "+15kb")
  }
  if(region=="genebody"){
    x_label <- c("-15kb","TSS", "TES", "+15kb")
  }
  # genebody_meth_list <- list()
  # indir <- paste0("F:/Project/3P/data/DNA/",region, "_profile_mtx/")
  # for(i in dir(indir)){
  #   fileName <- paste0(indir, i)
  #   genebody_meth_list[[i]] <- read.table(fileName,header = T, stringsAsFactors = F)
  # }
  # genebody_meth_merge_df<- do.call("rbind", genebody_meth_list)
  # rownames(genebody_meth_merge_df) <- 1:nrow(genebody_meth_merge_df)
  # write.table(
  #   genebody_meth_merge_df,
  #   file = paste0("F:/Project/3P/plot/MRT_4000.",region, "_meth_merge.txt"),
  #   col.names = T, row.names = F, quote = F, sep = "\t"
  # )

  genebody_meth_merge_df <- read.table(
    paste0("F:/Project/3P/plot/MRT_4000.",region, "_meth_merge.txt"),
    header = T, stringsAsFactors = F
  )
  info_tmp <- info_df[,c("Sample", "Group", "Lineage", "Embryo", "Stage", "lambda_percent")]
  genebody_meth_merge_df<- merge(genebody_meth_merge_df,info_tmp, by="Sample")

  ## plot for each embryo, quality control of embryo
  for(i in c("ICSI", "ST")){
    # pdf(paste0(region,"Profile_Embryo_", i, ".pdf"), height = 6, width = 8)
    p <- ggplot(data = genebody_meth_merge_df[genebody_meth_merge_df$Group==i,],
                aes(x=Coord, y=data, group=Sample, color=Group))+
      geom_line()+scale_x_continuous(breaks = c(0,50, 250,300),
                                     labels = x_label)+
      labs(y="DNA methylation level (%)")+
      facet_wrap(.~Embryo, ncol = 4)+
      coord_fixed(ratio = 4)+
      #facet_grid(Lineage~Stage)+
      scale_color_manual(values =  MRT_colorlist$Group_col[i])+
      theme_bw(base_size = 15)+
      theme(panel.grid.minor.x = element_blank(),
            axis.title.x = element_blank(),
            legend.position = "right",
            strip.background = element_blank(),
            axis.text.x = element_text(angle = 45, hjust = 1,vjust = 1),
            #strip.text = element_text(size = 15),
            plot.title = element_text(vjust = 1, hjust = .5))
    print(p)
    # dev.off()
  }
}
```

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/94c71fce-4bf6-4500-a9f5-81fa78024da6/sup.fig3.pdf

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/640720ab-80fa-4385-a9c1-30b969245b10/sup.fig4.pdf

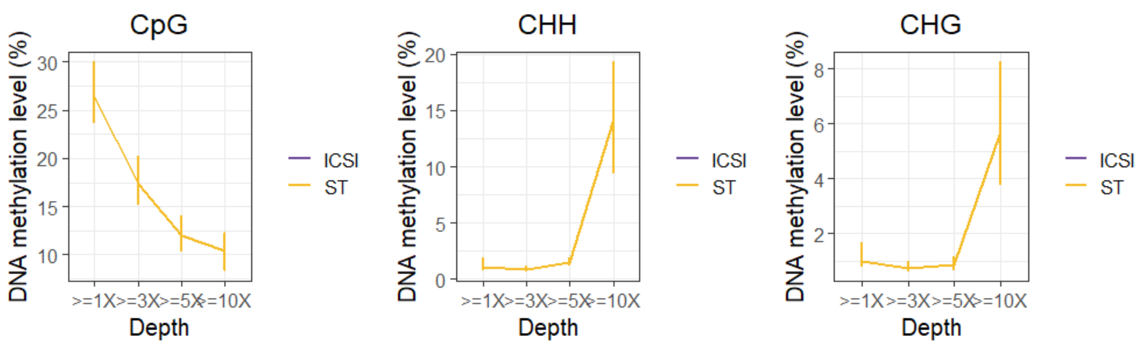
▼ fig.3.f

```
library(dplyr)

genebody_meth_group <- genebody_meth_merge_df %>%
  select(c("data", "Coord", "Group")) %>%
  group_by(Group, Coord) %>%
  dplyr::summarise(
    Median=median(data),
    Q1=quantile(data,0.25),
    Q3=quantile(data,0.75)
  ) %>%
  mutate(Pos=Coord)

ggplot(data = genebody_meth_group,aes(x=Coord))+
  geom_ribbon(aes(ymin=Q1, ymax=Q3, fill=Group),
              alpha=.3)+
  geom_line(aes(y=Median, group=Group, color=Group),
            size=1.5)+
  scale_fill_manual(values = MRT_colorlist$Group_col)+
  scale_color_manual(values = MRT_colorlist$Group_col)+
  scale_x_continuous(breaks = c(0,50, 250,300), labels = x_label)+
  labs(y="DNA methylation level (%)")+
  ylim(c(0,50))+
  coord_fixed(ratio = 3.5)+
  theme_bw(base_size = 25)+
  theme(panel.grid.minor.x = element_blank(),
        axis.title.x = element_blank(),
        legend.position = "right",
        strip.background = element_blank(),
        axis.text.x = element_text(angle = 0, hjust = .5,vjust = 1),
        #strip.text = element_text(size = 15),
        plot.title = element_text(vjust = 1, hjust = .5))
```



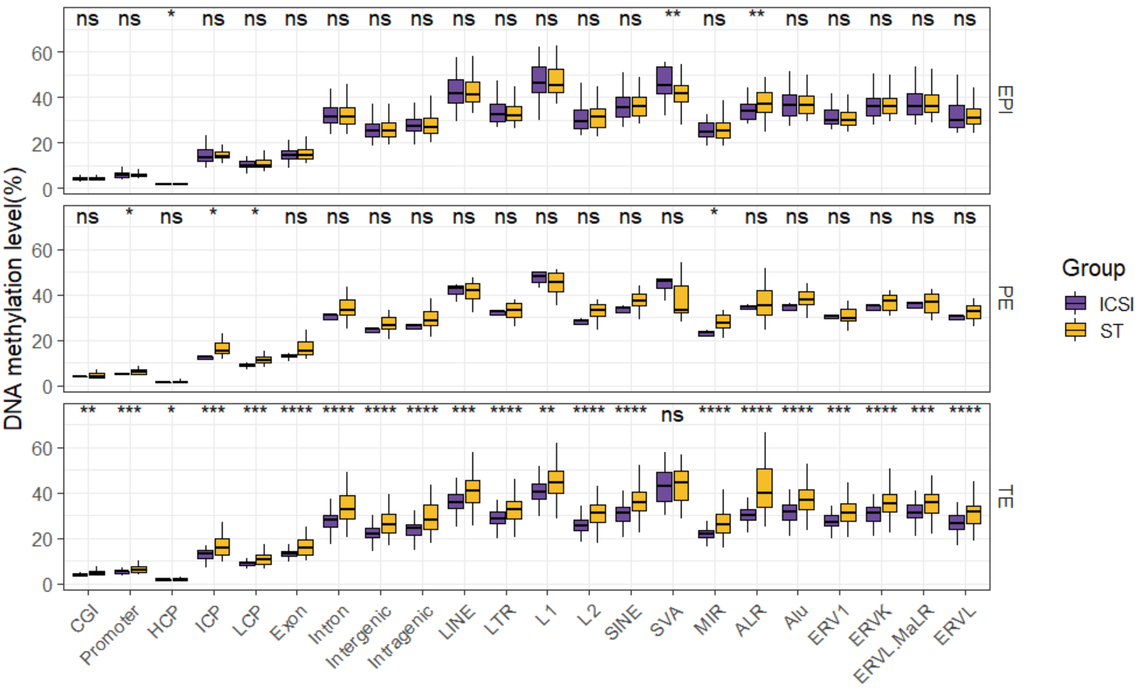
▼ sup.fig.2.b

```
meth_summary <- function(x, cells,group, type="ratio", C_type="CpG"){
  if (type=="site") {
    colume <- switch(C_type,
                     CpG=c(1,seq(3,18,5)),
                     CHH=c(1,seq(23,38,5)),
                     CHG=c(1,seq(43,58,5)))
  } else if (type=="ratio"){
    colume <- switch(C_type,
                     CpG=c(1,seq(5,20,5)),
                     CHH=c(1,seq(25,40,5)),
                     CHG=c(1,seq(45,60,5)))
  } else {
    stop("Unrecognized type!")
  }
  x <- na.omit(x)
  rownames(x) <- x$Sample
  tmp <- x[cells,colume]
  tmp <- na.omit(tmp)
  #if(class(tmp[,2])=="integer"){
  if(type=="site"){
    tmp[,2:5] <- round(tmp[, 2:5]/1e+06,2)
  }
  tmp_summary <- as.data.frame(t(apply(tmp[,2:5], 2, summary)))
  tmp_summary$Depth <- sapply(strsplit(rownames(tmp_summary), "[.]"), "[[", 2)
  rownames(tmp_summary) <- 1:dim(tmp_summary)[1]
  tmp_summary$Var <- group
  return(tmp_summary)
}

output_type <- "ratio"
# pdf(paste0("Meth_", output_type, "_Group_Depth_lambdaPercent.pdf"),width = 8, height = 7)
for (i in c("CpG","CHH","CHG")) {
  depth_summary_list <- list()
  for (j in c("ICSI", "ST")) {
    for (k in c(10)){
      # selector <- merge_df$Group==j &
      #   merge_df$Lambda_percent<50 &
      #   merge_df$lambda_percent==k
      # cells <- merge_df$Sample[selector]
      cells <- merge_df$Sample
      depth_summary_list[[paste0(j,"_",k)]] <- meth_summary(
        stat_meth,
        cells = cells,
        group = j,
        type=output_type,
        C_type = i
      )
      depth_summary_list[[paste0(j,"_",k)]]$Var <- paste0(depth_summary_list[[paste0(j,"_",k)]]$Var, "_", k)
    }
  }

  depth_summary_df<- do.call("rbind", depth_summary_list)
  depth_summary_df$lambda_percent <- sapply(strsplit(depth_summary_df$Var, "_"), "[[", 2)
  depth_summary_df$Var <- sapply(strsplit(depth_summary_df$Var, "_"), "[[", 1)
  depth_summary_df$Depth <- factor(depth_summary_df$Depth, levels = c("1X", "3X", "5X", "10X"))

  if(output_type=="ratio"){
    y_title <- "DNA methylation level (%)"
  } else if(output_type=="site"){
    y_title <- "Median of covered sites(M)"
  }
  p<- ggplot(data = depth_summary_df,
             aes_string(x="Depth",y="Median",color="Var", group="Var"))+
    geom_errorbar(
      aes(ymin=`1st Qu.`, ymax=`3rd Qu.`),
      width=.05, size=1)+
    geom_line(size=1)+
    scale_color_manual(values = MRT_colorlist$Group_col)+
    scale_x_discrete(label=c(">=1X", ">=3X", ">=5X", ">=10X"))+
    labs(y=y_title,title = i)+
    theme_bw(base_size = 25)+
    theme(legend.title = element_blank(),
          strip.background = element_blank(),
          plot.title = element_text(vjust = 1, hjust = .5)
    )
  print(p)
}
# dev.off()
```



▼ sup.fig.2.f

```
library(reshape2)

anno_list <- list()
indir <- "F:/Project/3P/data/DNA/CpG_site_anno/"
for(i in dir(indir)){
  fileName <- paste0(indir, i)
  anno_list[[i]] <- read.table(fileName,header = T, stringsAsFactors = F)
}
# anno_df<- do.call("rbind", anno_list)
# rownames(anno_df) <- 1:nrow(anno_df)
# write.table(
#   anno_df,
#   file = paste0("F:/Project/3P/plot/MRT_4000.CpG_site_anno.txt"),
#   col.names = T, row.names = F, quote = F, sep = "\t"
# )

info_tmp <- merge_df[,c("Sample", "Group", "Lineage", "Embryo", "Stage", "lambda_percent")]
anno_df<- merge(anno_df, info_tmp, by.x = "sample", by.y="Sample")

anno_long <- reshape2::melt(
  anno_df,
  id.vars = c("sample", "Group","Lineage",
              "Embryo", "Stage", "lambda_percent")
)
anno_long$variable <- factor(
  anno_long$variable,
  levels = c("CGI","Promoter", "HCP", "ICP", "LCP",
             "Exon", "Intron","Intergenic", "Intragenic",
             "LINE", "LTR", "L1", "L2",
             "SINE", "SVA", "MIR", "ALR", "Alu",
             "ERV1", "ERVK", "ERVL.MaLR","ERVL")
)

library(ggpubr)
library(RColorBrewer)
# pdf("DNA_element_Group_Lineage.pdf", height = 10, width = 15)
ggboxplot(
  anno_long,
  x = "variable",
  y = "value",
  fill = "Group",
  outlier.shape = NA,
  #add = "jitter",
  palette = MRT_colorlist$Group_col
  )+
  facet_grid(Lineage~.)+
  stat_compare_means(
    aes(group = Group),
       label="p.signif",
       size=5, vjust = .5
    )+
  labs(y="DNA methylation level(%)")+
  theme_bw(base_size = 15)+
  theme(strip.background = element_blank(),
        axis.title.x = element_blank(),
        axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1))
# dev.off()
```



▼ fig.3.d

1) Merge the bam files from the same lineage of ICSI and ST group, extract the methylation levels and divide the methylation files into 300-bp tiles.

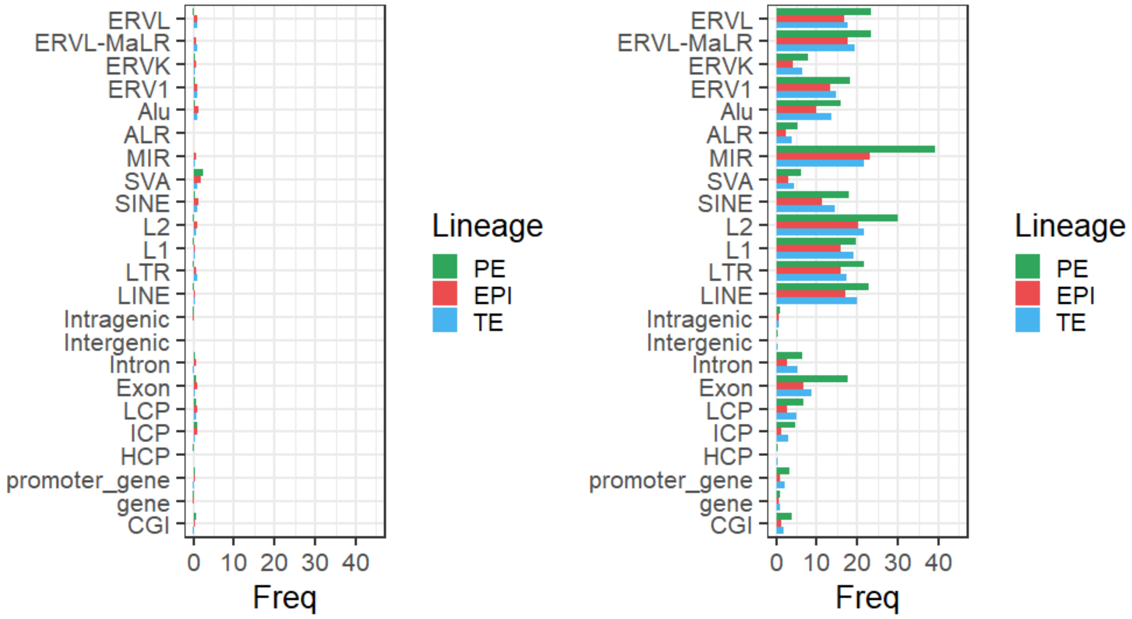
2) Tiles with DNA methylation less than 10% in one group and greater than 40% in the other group were defined as DMR and further annotated to genetic elements.

```
#!/bin/sh

############################################
### PBAT Pipeline                        ###
### Author: Xiaohui Xue                  ###
### Last Modification: 2021-04-02        ###
############################################

dir=$1
sp=$2
prefix=$3

echo "workdir=$dir"
echo "sample=$sp"

### Software
bin_dir=~/tangfuchou_coe/xuexiaohui/software
conda_PBAT_dir=$bin_dir/miniconda3/envs/PBAT/bin
samtools_exe=$conda_PBAT_dir/samtools
methylDackel_exe=$conda_PBAT_dir/MethylDackel
perl_exe=$conda_PBAT_dir/perl
bedtools_exe=$conda_PBAT_dir/bedtools

### Database
db_dir=~/tangfuchou_coe/xuexiaohui/database
ref=$db_dir/hg38/Bismark/hg38.genome_lambda.fa
annodir=$db_dir/hg38/Annotation/sub_group
genebody_ref=$db_dir/hg38/Annotation/hg38.gencode.p5.allGene.bed
promoter_ref=$db_dir/hg38/Annotation/hg38.gencode.p5.allGene_promoter.bed
element='ALR Alu ERV1 ERVK ERVL-MaLR ERVL Exon HCP ICP Intergenic Intragenic Intron L1 L2 LCP LINE LTR MIR SINE SVA'

### Script
code_dir=~/tangfuchou_coe/xuexiaohui/script/pipeline/PBAT_xxh
tile_pl=$code_dir/Tile_Meth_Methylkit_v2.pl

### Merge group bams
outdir=$dir/07.dmr/${prefix}
mkdir -p $outdir
bamlist=$outdir/${prefix}_bamlist.txt
merged_bam=$outdir/${prefix}_merged.bam

cat $sp | awk '{print "'$dir'""/02.bam/"$1"/"$1".sort.rmdup.bam"}' > $bamlist
```

```
function do_01_MergeGroupBam(){
  $samtools_exe merge -@ 3 -b $bamlist $merged_bam &&\
  $samtools_exe index $merged_bam
}

### Convert bam to methylkit
function do_02_Bam2Methylkit(){
  methylkit_prefix=$outdir/${prefix}_merged

  $methylDackel_exe extract \
  $ref -@ 1 --methylKit \
  --CHG --CHH \
  $merged_bam \
  -o $methylkit_prefix.tmp

  grep "Lambda" $methylkit_prefix.tmp* >\
  $methylkit_prefix.Lambda.methylkit

  for tp in CpG CHH CHG
  do
    #grep -v "Lambda" $methylkit_prefix.tmp_${tp}.methylKit  |\
    #grep -v "chrBase" > ${methylkit_prefix}.${tp}.methylkit
    cat $methylkit_prefix.tmp_${tp}.methylKit |\
    awk '{if(($1!="chrBase" && $1!~/^Lambda/) && ($6 <10 || $6 >90))print $0}' >\
    ${methylkit_prefix}.${tp}.methylkit
    rm $methylkit_prefix.tmp_${tp}.methylKit
  done
}

### Split tile (no use)
function do_03_SplitTile(){
  methylkit_prefix=$outdir/${prefix}_merged
  tile=${methylkit_prefix}.300bp_1X_CpG.txt
  $perl_exe $tile_pl ${methylkit_prefix}.CpG.methylkit 300 1 CpG $tile
}

### Annotation
function do_03_AnnoSite(){
  methylkit_prefix=$outdir/${prefix}_merged
  tmp=$outdir/${prefix}_merged_CpG.tmp.bed

  cat $methylkit_prefix.CpG.methylkit |\
  awk '{print $2"\t"$3"\t"$3"\t"$4"\t"$5"\t"$6}' > $tmp

  for i in $element
  do
    output=$methylkit_prefix.${i}_CpG.bed
    $bedtools_exe intersect \
    -b $tmp -a $annodir/hg38.$i.xls -wa -wb |\
    $bedtools_exe groupby -i - -g 1-4 -c 11 -o mean > $output
  done

  output=$methylkit_prefix.gene_CpG.bed
  $bedtools_exe intersect \
  -b $tmp -a $genebody_ref -wa -wb |\
  $bedtools_exe groupby -i - -g 1-4 -c 11 -o mean > $output

  output=$methylkit_prefix.CGI_CpG.bed
  $bedtools_exe intersect \
  -b $tmp -a $annodir/hg38.CGI.xls -wa -wb |\
  $bedtools_exe groupby -i - -g 1-4 -c 10 -o mean > $output

  output=$methylkit_prefix.promoter_gene_CpG.bed
  $bedtools_exe intersect \
  -b $tmp -a $promoter_ref -wa -wb |\
  $bedtools_exe groupby -i - -g 1-4 -c 11 -o mean > $output
  rm -rf $tmp
}
```

```
element='ALR Alu CGI ERV1 ERVK ERVL-MaLR ERVL Exon HCP ICP Intergenic Intragenic Intron L1 L2 LCP LINE LTR MIR SINE SVA gene promo

  for lineage in TE PE EPI
  do
      indir=$dir/07.dmr/$lineage
      outdir=$dir/07.dmr/DMR/$lineage
      mkdir -p $outdir
      for i in $element
      do
        echo "#!/bin/bash
        awk 'NR==FNR{a[\$1,\$2,\$3]=\$5}NR!=FNR && \
        a[\$1,\$2,\$3]{
          print \$1\"\\t\"\$2\"\\t\"\$3\"\\t\"\$4\"\\t\"a[\$1,\$2,\$3]\"\\t\"\$5
        }' $indir/ST/ST_merged.${i}_CpG.bed $indir/NC/NC_merged.${i}_CpG.bed > $outdir/DMR_merged.${i}_CpG.bed
        " > $script_prefix.$lineage.$i.tmp.sh

      done
  done
```

```
DMR_dir <- "F:/Project//3P/data/DNA/Meth/DMR/data/noCNV/"
regions <- c("CGI","HCP", "ICP", "LCP",
             "Exon", "Intron","Intergenic", "Intragenic",
             "LINE", "LTR", "L1", "L2",
             "SINE", "SVA", "MIR", "ALR", "Alu",
             "ERV1", "ERVK", "ERVL-MaLR","ERVL",
             "gene", "promoter_gene")
file.list <- list()
hyper_dmr_count_mtx <- matrix(ncol = 2, nrow = length(regions))
rownames(hyper_dmr_count_mtx) <- regions
hypo_dmr_count_mtx <- matrix(ncol = 2, nrow = length(regions))
rownames(hypo_dmr_count_mtx) <- regions
hyper_DMR_summary <- list()
hypo_DMR_summary <- list()
for (lineage in c("TE", "PE", "EPI")) {
  #lineage <- "PE"
  count_DMR <- TRUE
  # pdf(paste0(DMR_dir, "/plot/",lineage, "_merged.","CpG_correlation.pdf"), width = 7.73, height = 6.41)
  for(i in regions){
    fileName <- paste0(DMR_dir, lineage,"/DMR_merged.", i,"_CpG.bed")# TE_merged.gene_CpG
    file.list[[i]] <- as.data.frame(fread(fileName, header = F, stringsAsFactors = F))
    colnames(file.list[[i]]) <- c("Chr", "Start", "End", "Element", "ST_Meth", "NC_Meth")

    mtx <- file.list[[i]]
    total <- nrow(mtx)
    hypo_dmr <- nrow(mtx[(mtx$ST_Meth<10 & mtx$NC_Meth>40),])
    hyper_dmr <- nrow(mtx[(mtx$ST_Meth>40 & mtx$NC_Meth<10),])
    hyper_dmr_count_mtx[i,] <- c(total, hyper_dmr/total*100)
    hypo_dmr_count_mtx[i,] <- c(total, hypo_dmr/total*100)

    hyper_DMR_summary[[lineage]] <- as.data.frame(hyper_dmr_count_mtx)
    colnames(hyper_DMR_summary[[lineage]]) <- c("Count", "Freq")
    hypo_DMR_summary[[lineage]] <- as.data.frame(hypo_dmr_count_mtx)
    colnames(hypo_DMR_summary[[lineage]]) <- c("Count", "Freq")
}

## count hyper or hypo DMR
DMR_summary <- hypo_DMR_summary
DMR_summary <- do.call("rbind", DMR_summary)
DMR_summary <- na.omit(DMR_summary)
DMR_summary$Lineage <- sapply(strsplit(rownames(DMR_summary), "[.]"), "[[", 1)
DMR_summary$Lineage <- factor(DMR_summary$Lineage, levels = c("TE", "EPI", "PE"))
DMR_summary$Element <- sapply(strsplit(rownames(DMR_summary), "[.]"), "[[", 2)
DMR_summary$Element <- factor(DMR_summary$Element,
                              levels = c("CGI","gene","promoter_gene", "HCP", "ICP", "LCP",
                                          "Exon", "Intron","Intergenic", "Intragenic",
                                          "LINE", "LTR", "L1", "L2",
                                          "SINE", "SVA", "MIR", "ALR", "Alu",
                                          "ERV1", "ERVK", "ERVL-MaLR","ERVL"
                                         )
                              )

ggplot(data = DMR_summary, aes(x=Element, y=Freq, fill=Lineage))+
  geom_bar(stat = "identity", position = "dodge")+
  coord_flip()+
  ylim(c(0,45))+
  #scale_y_continuous(breaks = seq(0,50,10),labels = seq(0,50,10), limits = seq(0,50,10))+
  scale_fill_manual(values = MRT_colorlist$Lineage_col)+
  theme_bw(base_size = 20)+
  theme(axis.title.y = element_blank(),
        axis.text.y = element_text(hjust = 1))
}
```


ERVL
ERVL-MaLR
ERVK
ERV1
Alu
ALR
MIR
SVA
SINE
L2
L1
LTR
LINE
Intragenic
Intergenic
Intron
Exon
LCP
ICP
HCP
promoter_gene
gene
CGI
0
10
20
30
40
Freq
Lineage
PE
EPI
TE
ERVL
ERVL-MaLR
ERVK
ERV1
Alu
ALR
MIR
SVA
SINE
L2
L1
LTR
LINE
Intragenic
Intergenic
Intron
Exon
LCP
ICP
HCP
promoter_gene
gene
CGI
0
10
20
30
40
Freq
Lineage
PE
EPI
TE