FFPE Plots

```
## Fresh vs FFPE - number of variants called
df <- FreshvsFFPE[-c(4)]
df$logFresh <- log(as.numeric(df$Fresh))
df$logFFPE <- log(df$FFPE)
dflog <- df[-c(1,3,4)]
mx <- t(as.matrix(dflog[-1]))
colnames(mx) <- dflog$Tool
mxFresh <- t(as.matrix(Fresh[-c(1)]))
colnames(mxFresh) <- dflog$Tool
##Variants called by fresh frozen tumor and matching fresh frozen normal
barplot(mxFresh, main="Number of variants called by fresh frozen tumor and\nmatching fresh frozen normal
        ylab="Number of variants", col="#b2df8a", las=2, ylim=c(0,20000), mgp=c(3.2, 0.5, 0))

##Variants called FFPE tumor and fresh frozen normal
barplot(mx, main="Number of variants called by frozen fresh vs FFPE",
       ylab="log(Number of variants called)", col=c("#b2df8a", "#1f78b4"),
       las=2, mgp=c(3, 0.5, 0))
legend("topright", legend=c("FFPE", "Fresh"), fill=c("#1f78b4", "#b2df8a"))
```

```
library(ggpubr)
BLGSP_FreshPlot <- ggscatterhist(BLGSP_Fresh, x="Specificity", y="Sensitivity", size = 3, color="Tool",
                                 alpha=0.9, margin.plot = "histogram",
                                 margin.params = list(fill="Tool", color="black", size=0.2),
                                 bins =10, main.plot.size = 1, margin.plot.size = 1, xlim=c(0,1),
                                 title = "BLGSP Fresh with matched normal")
plot(BLGSP_FreshPlot)

BLGSP_MatchedPlot <- ggscatterhist(BLGSP_Matched, x="Specificity", y="Sensitivity", size = 3, color="To
                                 alpha=0.9, margin.plot = "histogram",
                                 margin.params = list(fill="Tool", color="black", size=0.2),
                                 bins =10, main.plot.size = 1, margin.plot.size = 1, xlim=c(0,1),
                                 title = "BLGSP FFPE with matched normal")
plot(BLGSP_MatchedPlot)

BLGSP_TumorPlot <- ggscatterhist(BLGSP_Tumor, x="Specificity", y="Sensitivity", size = 3, color="Tool",
                                 alpha=0.9, margin.plot = "histogram",
                                 margin.params = list(fill="Tool", color="black", size=0.2),
                                 bins =10, main.plot.size = 1, margin.plot.size = 1, xlim=c(0,1),
                                 title = "BLGSP FFPE Tumor Only")
plot(BLGSP_TumorPlot)

HTMCP_FreshPlot <- ggscatterhist(HTMCP_Fresh, x="Specificity", y="Sensitivity", size = 3, color="Tool",
                                 alpha=0.9, margin.plot = "histogram",
                                 margin.params = list(fill="Tool", color="black", size=0.2),
                                 bins =10, main.plot.size = 1, margin.plot.size = 1, xlim=c(0,1),
                                 title = "HTMCP Fresh with matched normal")
plot(HTMCP_FreshPlot)
```

```
HTMCP_MatchedPlot <- ggscatterhist(HTMCP_Matched, x="Specificity", y="Sensitivity", size = 3, color="Too
                                  alpha=0.9, margin.plot = "histogram",
                                  margin.params = list(fill="Tool", color="black", size=0.2),
                                  bins =10, main.plot.size = 1, margin.plot.size = 1, xlim=c(0,1),
                                  title = "HTMCP FFPE with matched normal")
plot(HTMCP_MatchedPlot)

HTMCP_TumorPlot <- ggscatterhist(HTMCP_Tumor, x="Specificity", y="Sensitivity", size = 3, color="Tool",
                                  alpha=0.9, margin.plot = "histogram",
                                  margin.params = list(fill="Tool", color="black", size=0.2),
                                  bins =10, main.plot.size = 1, margin.plot.size = 1, xlim=c(0,1),
                                  title = "HTMCP FFPE Tumor Only")
plot(HTMCP_TumorPlot)
```

```
##Combination Plots
plot(FinalCombi_HTMCP$Sensitivity, FinalCombi_HTMCP$Precision, xlim=c(0,1), ylim=c(0,1), xlab="Median se
     ylab="Median precision", main="FFPE SNV Caller - Combined Results HTMCP",
     cex=ifelse(FinalCombi_HTMCP$Type=='Indi', 3, 1),
     pch=ifelse(FinalCombi_HTMCP$Type=='Indi', 19, 4),
     col=c('#E64B35FF', '#4DBBD5FF', '#00A087FF',
           '#3C5488FF', '#F39B7FFF', '#7E6148FF',
           '#A9A9A9', '#A9A9A9', '#A9A9A9',
           '#A9A9A9', '#A9A9A9', '#A9A9A9',
           '#696969', '#696969', '#696969'))


Groups <- c("LoFreq", "SomVarIUS", "Strelka2",
            "Shimmer", "Virmid", "Mutect2", "Best F1 scores", "Best sensitivity or precision")
legend(x=-0.03, y=0.9, bty="n", legend=Groups,
       fill=c('#E64B35FF', '#4DBBD5FF', '#00A087FF',
              '#3C5488FF', '#F39B7FFF', '#7E6148FF',
              '#696969', '#DCDCDC'), cex=1.2)

plot(Finalcombi$Sensitivity, Finalcombi$Precision, axes=TRUE, xlim=c(0,1), ylim=c(0,1), xlab="Median sen
     ylab="Median precision", main="FFPE SNV Caller - Combined Results BLGSP",
     asp=0, cex=ifelse(Finalcombi$Type=='Indi', 3, 1),
     pch=ifelse(Finalcombi$Type=='Indi', 19, 4),
     col=c('#E64B35FF', '#4DBBD5FF', '#00A087FF',
                   '#3C5488FF', '#F39B7FFF', '#7E6148FF',
                   '#A9A9A9', '#A9A9A9', '#A9A9A9',
                   '#A9A9A9', '#A9A9A9', '#A9A9A9',
                   '#696969', '#696969', '#696969'))

Groups <- c("LoFreq", "SomVarIUS", "Strelka2",
            "Shimmer", "Virmid", "Mutect2", "Best F1 scores", "Best sensitivity or precision")
legend(x=max(Finalcombi$Precision)-1,y=max(Finalcombi$Precision)+0.05, bty="n", legend=Groups,
       fill=c('#E64B35FF', '#4DBBD5FF', '#00A087FF',
              '#3C5488FF', '#F39B7FFF', '#7E6148FF',
              '#696969', '#DCDCDC'), cex=1.2)
```

```
##Alleic fraction vs Insert size
library(ggplot2)
library(ggpubr)
AFvsIS <- ggscatterhist(sample, x="Insert.Size", y="AF", size=3,
```

```
                        color = "Type", palette = c("#1f78b4", "#b2df8a"),
                        alpha=0.2, margin.plot = "histogram",
                        margin.params = list(fill="Type", color="black", size=0.2),
                        bins =25, main.plot.size = 1, margin.plot.size = 1)
plot(AFvsIS)


table_FN$Type <- "FN"
table_TP$Type <- "TP"
table_FP$Type <- "FP"

table_total <- rbind(table_FP, table_FN, table_TP)
table_total$Depth <- table_total$REF_COUNT + table_total$ALT_COUNT
table_total$FractOfAlt <- ((table_total$ALT_COUNT)/table_total$Depth)
table_total <- table_total[table_total$Depth < 200, ]

results <- ggscatterhist(table_total, x="Depth", y="FractOfAlt", color="Type", alpha=0.5,
                    palette=c("#fdbf6f","#1f78b4","#b2df8a"), margin.plot = "histogram", ylab = "Allel
                    margin.params = list(fill="Type", color="black"), xlim=c(0, 200),
                    title = "Virmid - A66063")

#plot(results)
```

```
##ChipSeq
library(VariantAnnotation)
library(GenomicRanges)
library(ChIPpeakAnno)
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
library(dplyr)

finall_values <- matrix(ncol = 6, nrow = 21)
colnames(final_values) <- c('H3K4Me1', 'H3K4Me3', 'H3K9Me3', 'H3K27Me3', 'H3K36Me3', 'H3K27ac')
rownames(final_values) <- c(levels(droplevels(input_table$PID)))


values <- function(chipSeq_data, VcfRanges_data, FFPE_VCF_data, FalsePositivesVCF_data, TruePositivesVC
  chipSeq_range <- GRanges(chipSeq_data$chr, IRanges(chipSeq_data$start, chipSeq_data$end))

  Overlaps <- findOverlaps(VcfRanges_data, chipSeq_range)
  variantsHits <- FFPE_VCF_data[queryHits(Overlaps)]
  NumberOfHits <- NROW(as.data.frame(variantsHits@rowRanges@ranges@start))

  FP_InSample <- intersect.Vector(variantsHits@rowRanges@ranges, FalsePositivesVCF_data@rowRanges@ranges
  CountOfFPInSample <- NROW(as.data.frame(FP_InSample))

  TP_InSample <- intersect.Vector(variantsHits@rowRanges@ranges, TruePositivesVCF_data@rowRanges@ranges
  CountOfTPInSample <- NROW(as.data.frame(TP_InSample))

  FP_depletion <- phyper(CountOfFPInSample, CountOfFPInVCF, CountOfTPInVCF, NumberOfHits, lower.tail = T
  TP_depletion <- phyper(CountOfTPInSample, CountOfTPInVCF, CountOfFPInVCF, NumberOfHits, lower.tail = T
  FP_enrichment <- phyper(CountOfFPInSample-1, CountOfFPInVCF, CountOfTPInVCF, NumberOfHits, lower.tail
  TP_enrichment <- phyper(CountOfTPInSample-1, CountOfTPInVCF, CountOfFPInVCF, NumberOfHits, lower.tail
```

```
  result <- list(FPInSample_count=CountOfFPInSample, TPInSample_count=CountOfTPInSample, FPInVCF_count=C
  return(result)
}


for(i in rownames(input_table)){
  row <- input_table[i, ]
  PID <- levels(droplevels(row$PID))
  H3K4Me1 <- levels(droplevels(row$H3K4Me1))
  H3K4Me3 <- levels(droplevels(row$H3K4Me3))
  H3K9Me3 <- levels(droplevels(row$H3K9Me3))
  H3K27Me3 <- levels(droplevels(row$H3K27Me3))
  H3K36Me3 <- levels(droplevels(row$H3K36Me3))
  H3K27ac <- levels(droplevels(row$H3K27ac))
  H3K4Me1_pid <- levels(droplevels(row$H3K4Me1_PID))
  H3K4Me3_pid <- levels(droplevels(row$H3K4Me3_PID))
  H3K9Me3_pid <- levels(droplevels(row$H3K9Me3_PID))
  H3K27Me3_pid <- levels(droplevels(row$H3K27Me3_PID))
  H3K36Me3_pid <- levels(droplevels(row$H3K36Me3_PID))
  H3K27ac_pid <- levels(droplevels(row$H3K27ac_PID))


  Fresh_VCF <- readVcf(paste("Path_to_LoFreq_HTMCP_Results",PID,"_Freshsomatic_final.snvs.vcf.gz", sep =
  FFPE_VCF <- readVcf(paste("Path_to_LoFreq_HTMCP_Results",PID,"somatic_final.snvs.vcf.gz", sep =""), "H

  Fresh_VCF_table <- read.table(paste("Path_to_LoFreq_HTMCP_Results",PID,"_Freshsomatic_final.snvs.vcf.g
  FFPE_VCF_table <- read.table(paste("Path_to_LoFreq_HTMCP_Results",PID,"somatic_final.snvs.vcf.gz", sep

  TruePositivesVCF <- readVcf(paste("Path_to_LoFreq_HTMCP_Results",PID,"0_1.vcf.gz", sep = ""), "Hg19")
  FalsePositivesVCF <- readVcf(paste("Path_to_LoFreq_HTMCP_Results",PID,"1.vcf.gz", sep= ""), "Hg19")

  TruePositivesVCF_Table <- read.table(paste("Path_to_LoFreq_HTMCP_Results",PID,"0_1.vcf.gz", sep = ""))
  FalsePositivesVCF_Table <- read.table(paste("Path_to_LoFreq_HTMCP_Results",PID,"1.vcf.gz", sep = ""))

  VcfRanges <-GRanges(FFPE_VCF_table$V1, IRanges(FFPE_VCF_table$V2, FFPE_VCF_table$V2))

  CountOfTPInVCF <- NROW(TruePositivesVCF_Table)
  CountOfFPInVCF <- NROW(FalsePositivesVCF_Table)

  H3K4Me1_table <- read.table(paste("Path_to_Macs",H3K4Me1_pid,"/",H3K4Me1, sep = ""), header = TRUE)
  H3K4Me3_table <- read.table(paste("Path_to_Macs",H3K4Me3_pid,"/",H3K4Me3, sep = ""), header = TRUE)
  H3K9Me3_table <- read.table(paste("Path_to_Macs",H3K9Me3_pid,"/",H3K9Me3, sep = ""), header = TRUE)
  H3K27Me3_table <- read.table(paste("Path_to_Macs",H3K27Me3_pid,"/",H3K27Me3, sep = ""), header = TRUE
  H3K36Me3_table <- read.table(paste("Path_to_Macs",H3K36Me3_pid,"/",H3K36Me3, sep=""), header = TRUE)
  H3K27ac_table <- read.table(paste("Path_to_Macs", H3K27ac_pid,"/",H3K27ac, sep = ""), header = TRUE)


  H3K4Me1_results <- values(H3K4Me1_table, VcfRanges, FFPE_VCF, FalsePositivesVCF, TruePositivesVCF)
  H3K4Me3_results <- values(H3K4Me3_table, VcfRanges, FFPE_VCF, FalsePositivesVCF, TruePositivesVCF)
  H3K9Me3_results <- values(H3K9Me3_table, VcfRanges, FFPE_VCF, FalsePositivesVCF, TruePositivesVCF)
  H3K27Me3_results <- values(H3K27Me3_table, VcfRanges, FFPE_VCF, FalsePositivesVCF, TruePositivesVCF)
  H3K36Me3_results <- values(H3K36Me3_table, VcfRanges, FFPE_VCF, FalsePositivesVCF, TruePositivesVCF)
  H3K27ac_results <- values(H3K27ac_table, VcfRanges, FFPE_VCF, FalsePositivesVCF, TruePositivesVCF)
```

```
  finall_values[PID, "H3K4Me1"] <- paste(H3K4Me1_results, collapse = ',')
  finall_values[PID, "H3K4Me3"] <- paste(H3K4Me3_results, collapse = ',')
  finall_values[PID, "H3K9Me3"] <- paste(H3K9Me3_results, collapse = ',')
  finall_values[PID, "H3K27Me3"] <- paste(H3K27Me3_results, collapse = ',')
  finall_values[PID, "H3K36Me3"] <- paste(H3K36Me3_results, collapse = ',')
  finall_values[PID, "H3K27ac"] <- paste(H3K27ac_results, collapse = ',')
}


#P value adjustments
marks <- colnames(final_values)
sample <- rownames(final_values)
marks.sample <- paste(sample[row(final_values)], marks[col(final_values)], sep = ".")
i <-upper.tri(final_values, diag = TRUE)
dat <- data.frame(marks.sample[i], p.value=final_values[i])
dat$BH <- p.adjust(dat$p.value, method = "BH")
dat$Ber <- p.adjust(dat$p.value, method = "bonferroni")
```

```
##Computational Resources
a <- ggplot(ComputationalResources, aes(fill=Metric, y=log(Value), x=Tool)) +
  geom_bar(position="dodge", stat="identity") + ggtitle("Computational Resorces")
plot(a)
```