

Multiomic Mapping of Acquired Chromosome 1
Copy-Number and Structural Variants to Identify Therapeutic
Vulnerabilities in Multiple Myeloma

Patrick Blaney

July 22, 2025

Contents

Abstract	3
Environment Setup	3
Functions	4
Main Analysis - Hotspot Occurrence	5
Main Analysis - Compartment Quantification in Context of SV Hotspot Regions	20
Main Analysis - Transcribed Genes in Context of Chromosome 1 Arms	22
Main Analysis - Compartment Quantification in Context of Chromosome 1 Arms	26
Main Analysis - Methylation Profile	32
Main Analysis - Quantification of TADs in Context of Chromosome 1 Arms	58
Main Analysis - Compartment Quantification in Regions of Gain and Deletion	60
Main Analysis - Compartment Quantification in Context of Deletion Hotspots	80
Main Analysis - Compartment Quantification in Context of Gain(1q)	81
Main Analysis - Intersection of Super-Enhancers with Hotspots	87
Main Analysis - Histone Signal Enrichment in Hotspots	91

Authors:

Eileen M. Boyle^{1,11,12}, Patrick Blaney^{1,2,11}, James H Stoeckle^{1,11}, Yubao Wang¹, Hussein Ghamlouch^{1,4}, Dylan Gagler^{1,2}, Marc Braunstein¹, Louis Williams^{1,3}, Avital Tenenbaum¹, Ariel Siegel¹, Xiaoyi Chen^{1,5}, Gaurav Varma¹, Jason Avigan¹, Alexander Li¹, Monica Jinsi¹, David Kaminetzsky¹, Arnaldo Arbini¹, Lydia Montes⁴, Jill Corre⁵, Even H Rustad⁶, Ola Landgren⁷, Francesco Maura⁷, Brian A. Walker⁸, Michael Bauer⁹, Benedetto Bruno¹⁰, Aristotelis Tsirigos², Faith E. Davies¹, Gareth J. Morgan^{1,13}

Affiliations:

1. Myeloma Research Program, Perlmutter Cancer Center, NYU Langone Medical Center, New York, NY, USA
2. Applied Bioinformatics Laboratories, NYU Langone Medical Center, New York, NY, USA
3. Myeloma Group, Cleveland Clinic Foundation, Taussig Cancer Center, Cleveland, OH, USA
4. Université d'Amiens-Picardie, Amiens, France
5. Unit for Genomics in Myeloma, Institut Universitaire du Cancer de Toulouse-Oncopole, University Hospital, Toulouse; Centre de Recherche en Cancérologie de Toulouse, Institut National de la Santé et de la Recherche Médicale U1037, Toulouse, France
6. Institute for Cancer Research, Oslo University Hospital Radiumhospitalet, Oslo, Norway
7. Myeloma Service, Sylvester Comprehensive Cancer Center, University of Miami, Miami, FL, USA
8. Melvin and Bren Simon Comprehensive Cancer Center, Department of Hematology Oncology, Indiana University, Indianapolis, IN, USA
9. Department of Biomedical Informatics (DBMI), UAMS, Little-Rock, AR, USA
10. Department of Hematology, Azienda Ospedaliera Città della Salute e della Scienza di Torino, Piemonte, Italy
11. Contributed equally
12. Corresponding for manuscript submission
13. Corresponding author

Abstract

Purpose: Chromosome 1 (chr1) copy number abnormalities (CNAs) and structural variants (SV) are frequent in newly diagnosed multiple myeloma (NDMM) and associate with a heterogeneous impact on outcome the drivers of which are largely unknown. Experimental Design: A multiomic approach comprising CRISPR, gene mapping of CNA and SV, methylation, expression, and mutational analysis was used to document the extent of chr1 molecular variants and their impact on pathway utilisation.

Results: We identified two distinct groups of gain(1q): focal gains associated with limited gene expression changes and a neutral prognosis, and whole-arm gains, which associate with substantial gene expression changes, complex genetics and an adverse prognosis. CRISPR identified a number of dependencies on chr1 but only limited variants associated with acquired CNAs. We identified seven regions of deletion, nine of gain, three of chromothripsis (CT) and two of templated-insertion (TI), which contain a number of potential drivers. An additional mechanism involving hypomethylation of genes at 1q may contribute to the aberrant gene expression of a number of genes. Expression changes associated with whole-arm gains were substantial and gene set enrichment analysis identified metabolic processes, apoptotic resistance, signaling via the MAPK pathway, and upregulation of transcription factors as being key drivers of the adverse prognosis associated with these variants.

Conclusions: Multiple layers of genetic complexity impact the phenotype associated with CNAs on chr1 to generate its associated clinical phenotype. Whole-arm gains of 1q are the critically important prognostic group that deregulate multiple pathways, which may offer therapeutic vulnerabilities.

Environment Setup

```
library(ensemblDb)
library(AnnotationHub)
library(GGally)
library(cowplot)
library(tidyverse)
library(EnhancedVolcano)
library(viridis)
library(ggpubr)
library(ggrepel)
library(ComplexHeatmap)
library(RnBeads)
library(RnBeads.hg19)
library(GenomicRanges)
library(karyoploteR)
library(BSgenome.Hsapiens.UCSC.hg38)
library(TxDb.Hsapiens.UCSC.hg38.knownGene)
# library(Ringo)
library(sjstats)
library(R.utils)
library(Gviz)

options(scipen = 999)

# Set figure base theme
theme_set(
  theme_gray() +
  theme(
    axis.line = element_line(size = 0.5, color = "black"),
    axis.ticks = element_line(size = 0.5, color = "black"),
    panel.grid = element_line(size = 0.5, color = "gray")
  )
)
```

```

        panel.background = element_rect(fill = NA, size = rel(14)),
        panel.grid.minor = element_line(color = NA),
        axis.text = element_text(size = 12, color = "black"),
        axis.title = element_text(size = 14),
        axis.ticks = element_line(size = 0.75),
        title = element_text(size = 16),
        plot.title = element_text(hjust = 0.5)
    )
}

base_data_path <- "../../data/"

```

Functions

```

# From the now archived package Ringo
plotBM <- function(x, boxCol = "darkblue", reorder = FALSE, frame = TRUE, ...) {
  stopifnot(is.matrix(x))
  if (reorder) {
    ## treat them
    whichGroup <- x %*% 2^((ncol(x) - 1):0)
    numTimes <- table(whichGroup)
    ## to avoid breaks in case >=2 categories occur equally often:
    # numTimes <- numTimes + cumsum(rep(0.1, length(numTimes)))
    ord <- order(numTimes[as.character(whichGroup)], whichGroup, decreasing = FALSE)
  } else {
    ord <- nrow(x):1
  }
  x <- x[ord, ]
  blockBorders <- apply(x, 2, function(x) diff(c(FALSE, x, FALSE)))
  plot(y = 0, xlim = c(0, ncol(x)), x = 0, ylim = c(0, nrow(x)),
    type = "n", xaxt = "n", yaxt = "n", xlab = NA, ylab = NA,
    frame.plot = FALSE, ...)
  for (j in 1:ncol(blockBorders)) {
    theseBlocks <- rbind(c(0, 0), cbind(which(blockBorders[, j] == 1) - 1,
                                          which(blockBorders[, j] == -1) - 1))
    for (i in 2:nrow(theseBlocks)) {
      polygon(x = j + c(-1, -1, 0, 0),
              y = theseBlocks[i, c(1, 2, 2, 1)],
              col = boxCol, border = NA)
    }
  }
  ## draw frame around plotted matrix:
  if (frame) {
    arrows(y0 = 0, y1 = 0, x0 = 0, x1 = ncol(x), length = 0)
    arrows(y0 = nrow(x), y1 = nrow(x), x0 = 0, x1 = ncol(x), length = 0)
    arrows(y0 = 0, y1 = nrow(x), x0 = ncol(x), x1 = ncol(x), length = 0)
    arrows(y0 = 0, y1 = nrow(x), x0 = 0, x1 = 0, length = 0)
  } # if (frame)
  ## annotate axes
  if (!is.null(rownames(x))) {
    axis(side = 2, at = seq(nrow(x)) - 0.5, labels = rownames(x),

```

```

        tick = FALSE, line = 0, las = 1)
    }
    if (!is.null(colnames(x))) {
      axis(side = 1, at = seq(ncol(x)) - 0.5, labels = colnames(x),
            tick = FALSE, line = 0)
    }
    invisible(x)
} # plotBM

```

Main Analysis - Hotspot Occurrence

Read in BED file that includes the occurrence of gain and deletion hotspot per patient

```

mmrf_set_for_pub <- read_delim(
  file = paste0(base_data_path, "mmrfSampleSetForPub.txt"),
  delim = "\t",
  col_names = T,
  show_col_types = F
)

gain_hotspot_occurrence <- read_delim(
  file = paste0(base_data_path,
                "mmrfChr1Cnvs/gisticRegions.patient.gains.bed"),
  delim = "\t",
  col_select = c(4, 8),
  col_names = F,
  show_col_types = F
)
colnames(gain_hotspot_occurrence) <- c("sample", "hotspot")

gain_hotspot_occurrence <- gain_hotspot_occurrence %>%
  filter(sample %in% mmrf_set_for_pub$sample)

gain_hotspots <- c(
  "G1", "G2", "G3", "G4", "G5",
  "G6", "G7", "G8", "G9"
)
gain_sample_names <- gain_hotspot_occurrence$sample %>%
  unique() %>%
  as_vector()

del_hotspot_occurrence <- read_delim(
  file = paste0(base_data_path,
                "mmrfChr1Cnvs/gisticRegions.patient.deletions.bed"),
  delim = "\t",
  col_select = c(4, 8),
  col_names = F,
  show_col_types = F
)
colnames(del_hotspot_occurrence) <- c("sample", "hotspot")

del_hotspot_occurrence <- del_hotspot_occurrence %>%

```

```

filter(sample %in% mmrf_set_for_pub$sample)

del_hotspots <- c(
  "D1", "D2", "D3", "D4",
  "D5", "D6", "D7"
)
del_sample_names <- del_hotspot_occurrence$sample %>%
  unique() %>%
  as_vector()

```

Create binary matrix of samples by gain hotspots to use for heatmap visualization

```

gain_hotspot_occurrence_matrix <- matrix(
  nrow = length(gain_sample_names),
  ncol = length(gain_hotspots),
  dimnames = list(
    gain_sample_names,
    gain_hotspots
  )
)

for (i in 1:length(gain_sample_names)) {
  gain_sample_specific_occurrence <- gain_hotspot_occurrence %>%
    filter(sample == gain_sample_names[i]) %>%
    dplyr::select(hotspot) %>%
    as_vector()

  for (j in 1:length(gain_hotspots)) {
    gain_hotspot_occurrence_matrix[i, j] <- as.numeric(gain_hotspots[j] %in%
      gain_sample_specific_occurrence)
  }
}

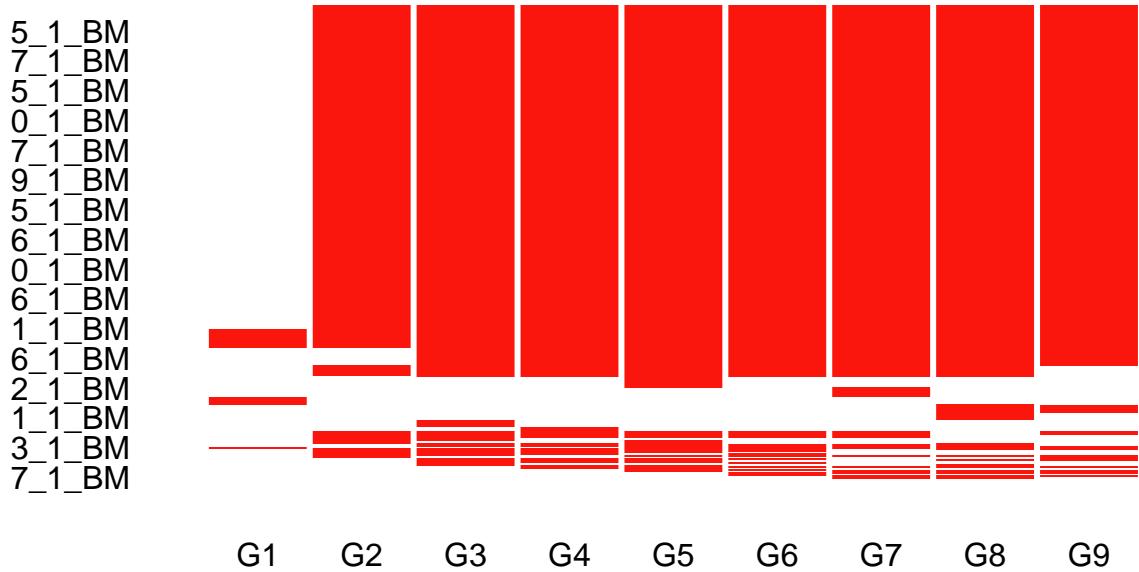
```

Generate visualizations of gain hotspot occurrences

```

plotBM(
  x = gain_hotspot_occurrence_matrix,
  boxCol = "#FA160C",
  reorder = TRUE,
  frame = FALSE
)
abline(
  v = 0:length(gain_hotspots),
  lwd = 3,
  col = "white"
)

```



```

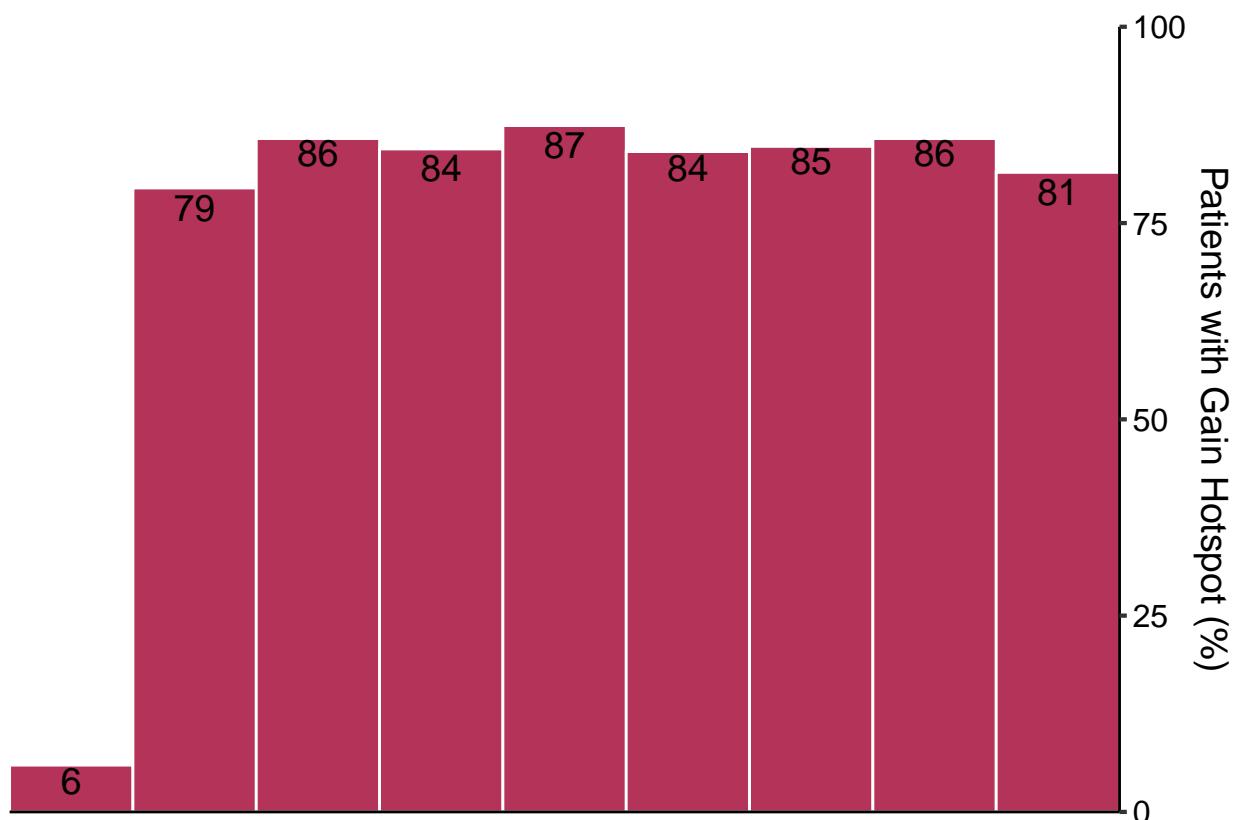
ggplot() +
  geom_col(aes(x = gain_hotspots,
                y = (colSums(x = gain_hotspot_occurrence_matrix) / 302 * 100)),
            width = 1,
            color = "white",
            fill = inferno(n = 30)[15]
  ) +
  theme(
    axis.ticks.x = element_blank(),
    axis.text.x = element_blank()
  ) +
  scale_y_continuous(
    expand = c(0, 0),
    breaks = seq(0, 100, 25),
    limits = c(0, 100),
    position = "right"
  ) +
  scale_x_discrete(expand = c(0, 0)) +
  ylab("Patients with Gain Hotspot (%)") +
  xlab("") +
  annotate(
    geom = "text",
    label = round(
      x = (colSums(x = gain_hotspot_occurrence_matrix) / 302 * 100),
      digits = 0
    ),
  ),

```

```

x = seq(1, 9, 1),
y = round(
  x = (colSums(x = gain_hotspot_occurrence_matrix) / 302 * 100),
  digits = 0
) - 2,
size = 5,
color = "black"
)

```



```

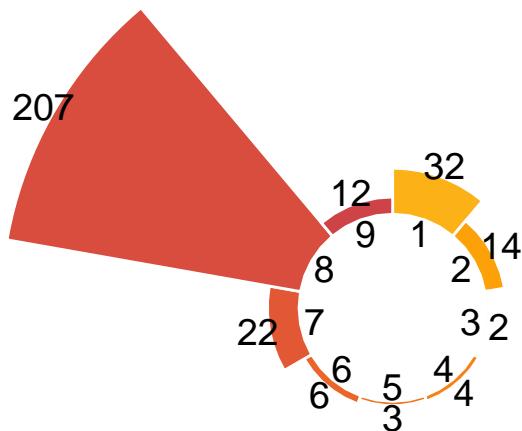
ggplot() +
  geom_col(
    aes(
      x = seq(1, 9, 1),
      y = as_factor(rowSums(gain_hotspot_occurrence_matrix)) %>% summary()
    ),
    width = 1,
    color = "white",
    fill = inferno(n = 30)[25:17]
  ) +
  ylim(-65, 215) +
  theme_minimal() +
  theme(
    axis.text = element_blank(),
    axis.title = element_blank(),
    panel.grid = element_blank(),
    plot.title = element_text(size = 10, color = "black", face = "bold")
  )

```

```

    plot.margin = unit(rep(-1, 4), "cm")
) +
annotate(
  geom = "text",
  label = seq(1, 9, 1),
  x = seq(1, 9, 1),
  y = -10,
  size = 5,
  color = "black"
) +
annotate(
  geom = "text",
  label = as_factor(rowSums(gain_hotspot_occurrence_matrix)) %>% summary(),
  x = seq(1, 9, 1),
  y = as_factor(rowSums(gain_hotspot_occurrence_matrix)) %>% summary() + 8,
  size = 5,
  color = "black"
) +
coord_polar(start = 0)

```



```

# heatmap(hotspot_occurrence_matrix,
#         scale = "none",
#         col = c("white", "mediumseagreen")
#       )

```

```
# ggplot(reshape::melt(hotspot_occurrence_matrix)) +
#   geom_tile(aes(x = X2, y = X1, fill = c("white", "mediumseagreen")[value + 1]),
#             color = "black") +
#   scale_fill_identity() +
#   xlab("Gain Hotspot") +
#   ylab("") +
#   theme(axis.text.y.left = element_text(size = 5))
```

Quantify various metrics of gain hotspots for use in publication

```
# Number of patients that harbor each gain hotspot
colSums(x = gain_hotspot_occurrence_matrix)
```

```
##   G1   G2   G3   G4   G5   G6   G7   G8   G9
##   18  240  259  255  264  254  256  259  246
```

```
# Total number of hotspots per patient
as_factor(rowSums(gain_hotspot_occurrence_matrix)) %>% summary()
```

```
##   1    2    3    4    5    6    7    8    9
##   32   14    2    4    3    6   22  207   12
```

```
# Total number of patients with 7 or more hotspots
# and that as a percentage of all patients
sum(rowSums(gain_hotspot_occurrence_matrix) >= 7)
```

```
## [1] 241
```

```
sum(rowSums(gain_hotspot_occurrence_matrix) >= 7) / 302 * 100
```

```
## [1] 79.80132
```

```
# Number of patients with different combinations of gain hotspots
nrow(as_tibble(gain_hotspot_occurrence_matrix) %>% dplyr::filter(G1 == 0 &
  G2 == 1 &
  G3 == 1 &
  G4 == 1 &
  G5 == 1 &
  G6 == 1 &
  G7 == 1 &
  G8 == 1 &
  G9 == 1))
```

```
## [1] 207
```

Create binary matrix of samples by deletion hotspots to use for heatmap visualization

```

del_hotspot_occurrence_matrix <- matrix(
  nrow = length(del_sample_names),
  ncol = length(del_hotspots),
  dimnames = list(
    del_sample_names,
    del_hotspots
  )
)

for (i in 1:length(del_sample_names)) {
  del_sample_specific_occurrence <- del_hotspot_occurrence %>%
    dplyr::filter(sample == del_sample_names[i]) %>%
    dplyr::select(hotspot) %>%
    as_vector()

  for (j in 1:length(del_hotspots)) {
    del_hotspot_occurrence_matrix[i, j] <- as.numeric(del_hotspots[j] %in%
      del_sample_specific_occurrence)
  }
}

# Write list of patients with any loss
# write_delim(as.data.frame(del_hotspot_occurrence_matrix) %>%
#             rownames_to_column(var = "patient"),
#             file = "~/Downloads/patientDeletionOccurrenceMatrix.txt",
#             delim = "\t",
#             col_names = T)

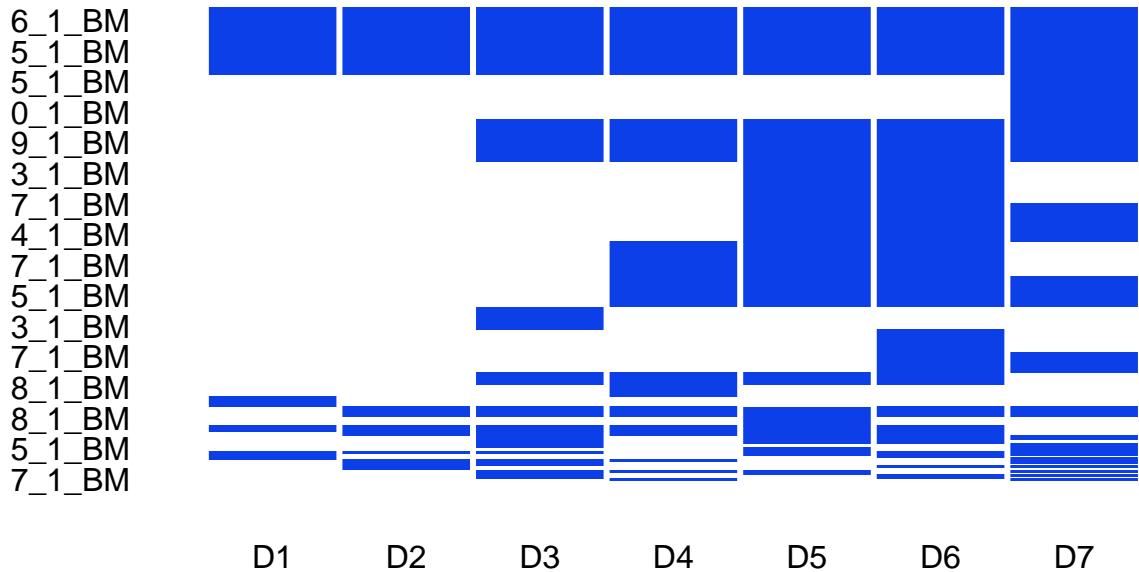
```

Generate visualizations of deletion hotspot occurrences

```

plotBM(
  x = del_hotspot_occurrence_matrix,
  boxCol = "#0C3FE8",
  reorder = TRUE,
  frame = FALSE
)
abline(
  v = 0:length(del_hotspots),
  lwd = 3,
  col = "white"
)

```

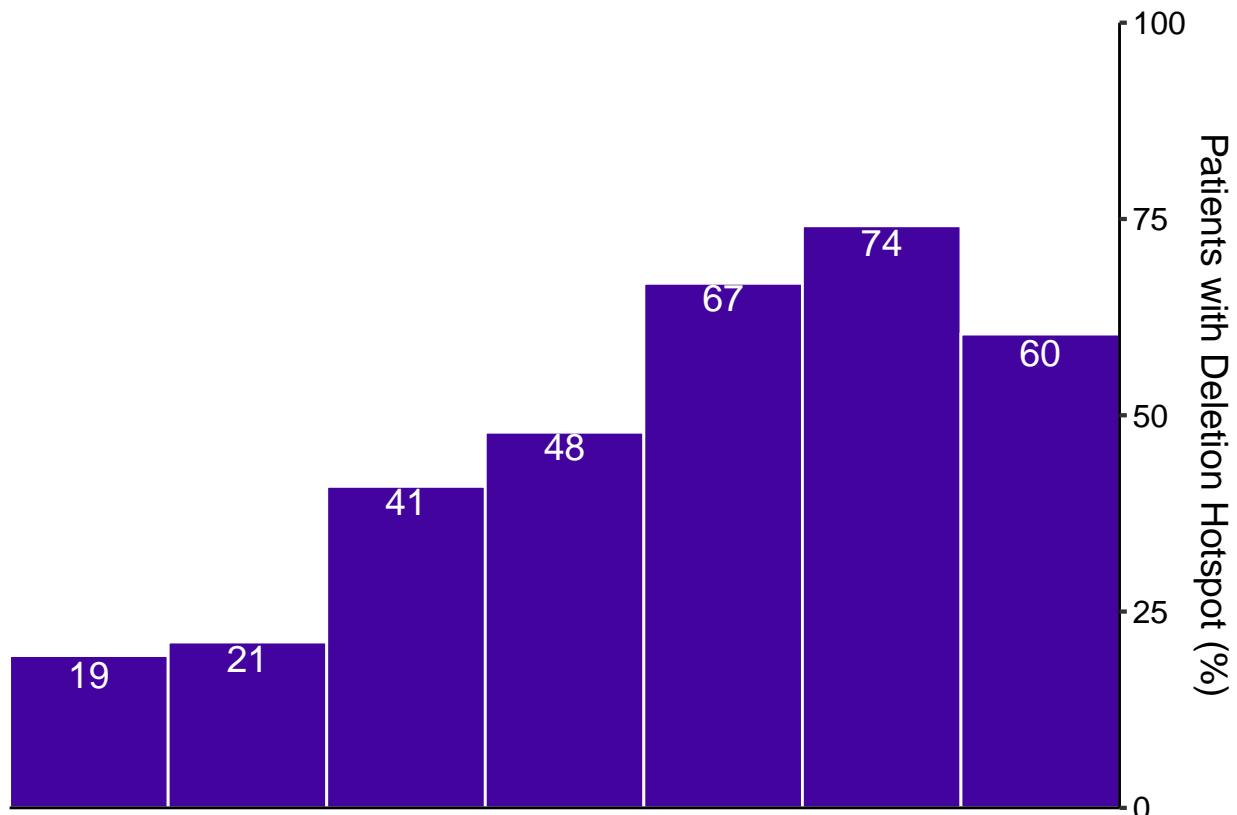


```
ggplot() +
  geom_col(aes(x = del_hotspots,
                y = (colSums(x = del_hotspot_occurrence_matrix) / 232 * 100)),
            width = 1,
            color = "white",
            fill = plasma(n = 30)[4]
  ) +
  theme(
    axis.ticks.x = element_blank(),
    axis.text.x = element_blank()
  ) +
  scale_y_continuous(
    expand = c(0, 0),
    breaks = seq(0, 100, 25),
    limits = c(0, 100),
    position = "right"
  ) +
  scale_x_discrete(expand = c(0, 0)) +
  ylab("Patients with Deletion Hotspot (%)") +
  xlab("") +
  annotate(
    geom = "text",
    label = round(
      x = (colSums(x = del_hotspot_occurrence_matrix) / 232 * 100),
      digits = 0
    ),
    
```

```

    x = seq(1, 7, 1),
    y = round(
      x = (colSums(x = del_hotspot_occurrence_matrix) / 232 * 100),
      digits = 0
    ) - 2,
    size = 5,
    color = "white"
  )
)

```



```

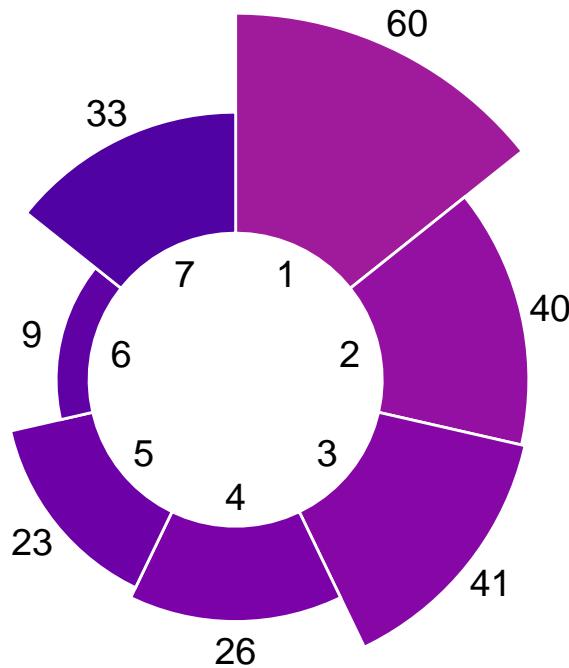
ggplot() +
  geom_col(
    aes(
      x = seq(1, 7, 1),
      y = as_factor(rowSums(del_hotspot_occurrence_matrix)) %>% summary()
    ),
    width = 1,
    color = "white",
    fill = plasma(n = 30)[11:5]
  ) +
  ylim(-40, 70) +
  scale_x_discrete(expand = c(0, 0)) +
  theme_minimal() +
  theme(
    axis.text = element_blank(),
    axis.title = element_blank(),
    plot.title = element_text(size = 14, weight = "bold", color = "#E69138"),
    plot.subtitle = element_text(size = 12, color = "#E69138")
  )
)

```

```

    panel.grid = element_blank(),
    plot.margin = unit(rep(-1, 4), "cm")
) +
  annotate(
    geom = "text",
    label = seq(1, 7, 1),
    x = seq(1, 7, 1),
    y = -8,
    size = 5,
    color = "black"
) +
  annotate(
    geom = "text",
    label = as_factor(rowSums(del_hotspot_occurrence_matrix)) %>% summary(),
    x = seq(1, 7, 1),
    y = as_factor(rowSums(del_hotspot_occurrence_matrix)) %>% summary() + 8,
    size = 5,
    color = "black"
) +
  coord_polar(start = 0)

```



Quantify various metrics of deletion hotspots for use in publication

```
# Number of patients that harbor each deletion hotspot
colSums(x = del_hotspot_occurrence_matrix)
```

```

##  D1  D2  D3  D4  D5  D6  D7
##  45  49  95 111 155 172 140

# Total number of deletion hotspots per patient
as_factor(rowSums(del_hotspot_occurrence_matrix)) %>% summary()

##  1  2  3  4  5  6  7
## 60 40 41 26 23  9 33

# Total number of patients with 3 or less hotspots
# and that as a percentage of all patients
sum(rowSums(del_hotspot_occurrence_matrix) <= 3)

## [1] 141

sum(rowSums(del_hotspot_occurrence_matrix) <= 3) / 232 * 100

## [1] 60.77586

# Number of patients with different combinations of deletion hotspots
print("Number of patients with all 7 Deletion hotspots")

## [1] "Number of patients with all 7 Deletion hotspots"

nrow(as_tibble(del_hotspot_occurrence_matrix) %>% dplyr::filter(D1 == 1 &
  D2 == 1 &
  D3 == 1 &
  D4 == 1 &
  D5 == 1 &
  D6 == 1 &
  D7 == 1))

## [1] 33

print("Number of patients with only 1 Deletion hotspots at D7 (TENT5C)")

## [1] "Number of patients with only 1 Deletion hotspots at D7 (TENT5C)"

nrow(as_tibble(del_hotspot_occurrence_matrix) %>% dplyr::filter(D1 == 0 &
  D2 == 0 &
  D3 == 0 &
  D4 == 0 &
  D5 == 0 &
  D6 == 0 &
  D7 == 1))

## [1] 22

```

```

print("Number of patients with 6 of 7 Deletion hotspots, all but p-arm telomere")

## [1] "Number of patients with 6 of 7 Deletion hotspots, all but p-arm telomere"

nrow(as_tibble(del_hotspot_occurrence_matrix) %>% dplyr::filter(D1 == 0 &
  D2 == 0 &
  D3 == 1 &
  D4 == 1 &
  D5 == 1 &
  D6 == 1 &
  D7 == 1))

```

[1] 21

Identify the co-occurrence of both gain and deletion hotspots within the same patient

```

both_hotspot_occurrence <- inner_join(
  x = as_tibble(gain_hotspot_occurrence_matrix) %>%
    mutate(sample = rownames(gain_hotspot_occurrence_matrix)),
  y = as_tibble(del_hotspot_occurrence_matrix) %>%
    mutate(sample = rownames(del_hotspot_occurrence_matrix)),
  by = "sample"
) %>%
  dplyr::select(
    sample,
    D1, D2, D3, D4, D5, D6, D7,
    G1, G2, G3, G4, G5, G6, G7, G8, G9,
  )
)

phi_correlation_coef_matrix <- matrix(
  nrow = length(c(del_hotspots, gain_hotspots)),
  ncol = length(c(del_hotspots, gain_hotspots)),
  dimnames = list(
    c(del_hotspots, gain_hotspots),
    c(del_hotspots, gain_hotspots)
  )
)

phi_corr_p_values <- matrix(
  nrow = length(c(del_hotspots, gain_hotspots)),
  ncol = length(c(del_hotspots, gain_hotspots)),
  dimnames = list(
    c(del_hotspots, gain_hotspots),
    c(del_hotspots, gain_hotspots)
  )
)

for (i in 1:length(c(del_hotspots, gain_hotspots))) {
  for (j in 1:length(c(del_hotspots, gain_hotspots))) {
    phi_correlation_coef_matrix[i, j] <- phicoef(
      x = table(
        column_torownames(

```

```

        .data = both_hotspot_occurrence,
        var = "sample"
    )[, i],
    column_to_rownames(
        .data = both_hotspot_occurrence,
        var = "sample"
    )[, j]
)
)

phi_corr_p_values[i, j] <- crosstable_statistics(
    data = table(
        column_to_rownames(
            .data = both_hotspot_occurrence,
            var = "sample"
        )[, i],
        column_to_rownames(
            .data = both_hotspot_occurrence,
            var = "sample"
        )[, j]
    ),
    statistics = "phi"
)$p.value
}
}

# Adjust each phi correlation p-value for multiple testing by applied FDR adjustment
phi_corr_p_adj_values <- matrix(
    nrow = length(c(del_hotspots, gain_hotspots)),
    ncol = length(c(del_hotspots, gain_hotspots)),
    dimnames = list(
        c(del_hotspots, gain_hotspots),
        c(del_hotspots, gain_hotspots)
    )
)
for (i in 1:nrow(phi_corr_p_values)) {
    phi_corr_p_adj_values[i, ] <- p.adjust(phi_corr_p_values[i, ], method = "fdr")
}

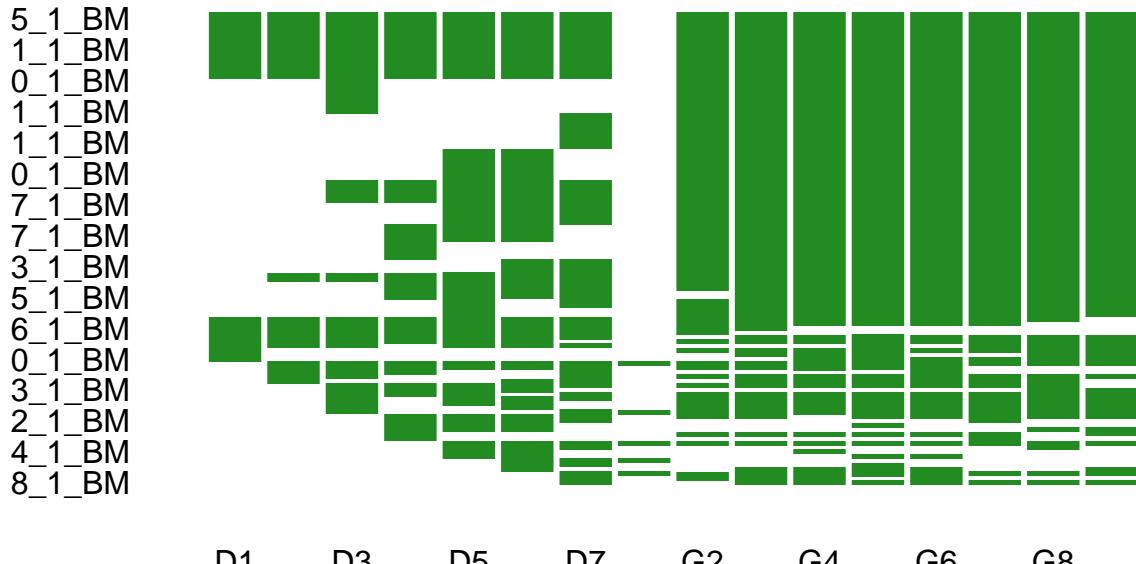
plotBM(
    x = as.matrix(
        column_to_rownames(both_hotspot_occurrence,
                           var = "sample"
        )
    ),
    reorder = TRUE,
    boxCol = "forestgreen",
    frame = FALSE
)
abline(
    v = 0:length(c(del_hotspots, gain_hotspots)),
    lwd = 3,

```

```

    col = "white"
)

```



```

# write table of correlations and p-values
# write_delim(as.data.frame(phi_correlation_coef_matrix) %>%
#             rownames_to_column(var = "region_id"),
#             file = "~/Downloads/gainAndDeletionPhiCorrMatrix.txt",
#             delim = "\t",
#             col_names = T)
# write_delim(as.data.frame(phi_corr_p_adj_values) %>%
#             rownames_to_column(var = "region_id"),
#             file = "~/Downloads/gainAndDeletionPhiCorrFdrPvalues.txt",
#             delim = "\t",
#             col_names = T)

```

```

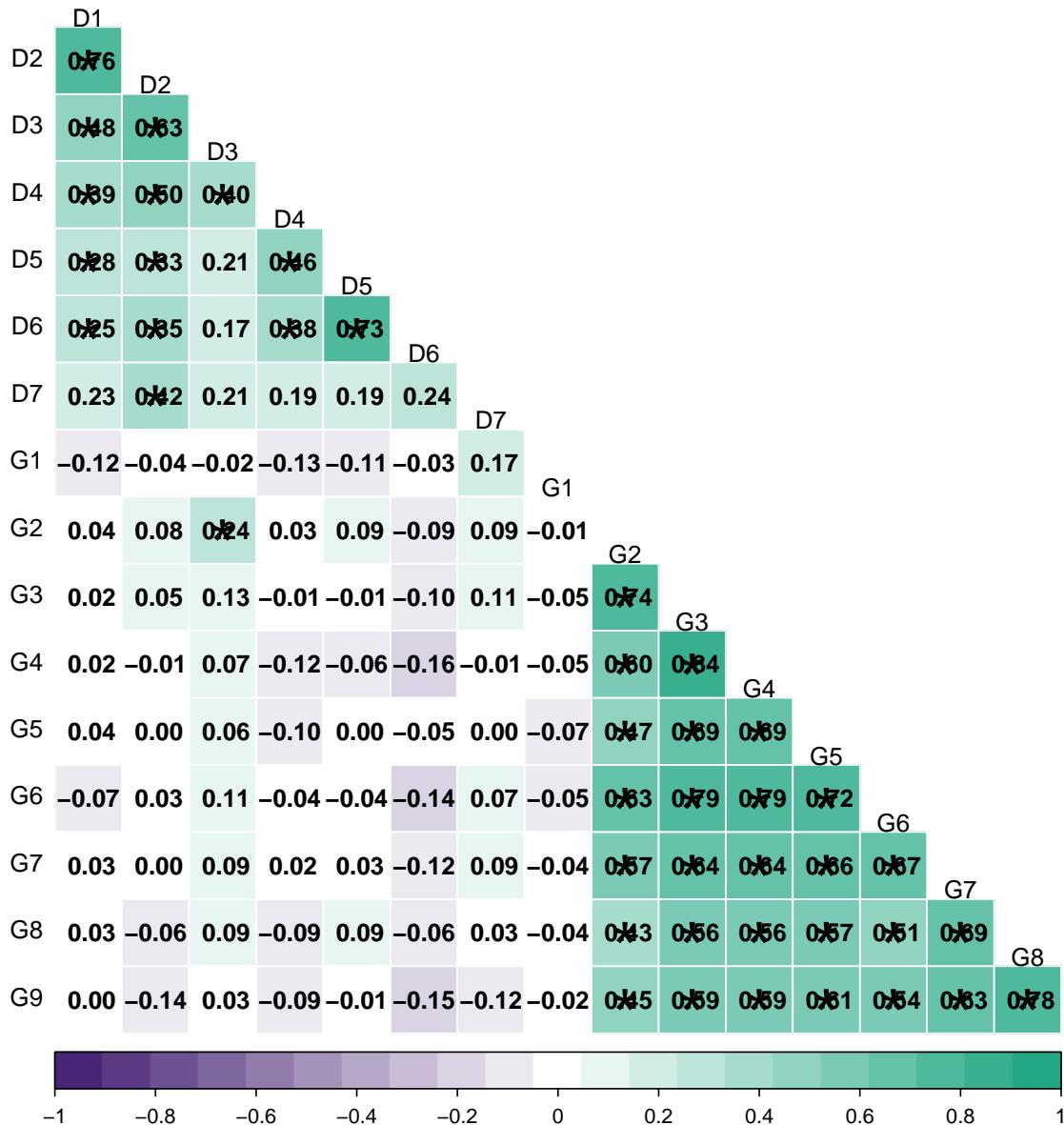
corrplot_colors <- colorRampPalette(c(viridis(21)[3], "white", viridis(21)[13]))
corrplot::corrplot(
  corr = phi_correlation_coef_matrix,
  method = "color",
  type = "lower",
  tl.col = "black",
  tl.srt = 0,
  diag = FALSE,
  addgrid.col = "white",
  addCoef.col = "black",

```

```

    col = corrplot_colors(21),
    p.mat = phi_corr_p_adj_values,
    sig.level = 0.05,
    insig = "label_sig",
    pch.cex = 3
)

```



Quantify number of patients with common pattern of gain and deletion hotspots

```
print("Number of patients with all hotspots but Gain on p-arm telomere")
```

```
## [1] "Number of patients with all hotspots but Gain on p-arm telomere"
```

```

nrow(as_tibble(both_hotspot_occurrence) %>% filter(D1 == 1 &
D2 == 1 &
D3 == 1 &
D4 == 1 &
D5 == 1 &
D6 == 1 &
D7 == 1 &
G1 == 0 &
G2 == 1 &
G3 == 1 &
G4 == 1 &
G5 == 1 &
G6 == 1 &
G7 == 1 &
G8 == 1 &
G9 == 1))

## [1] 15

```

Main Analysis - Compartment Quantification in Context of SV Hotspot Regions

Read in BED files for templated insertions and chromothripsis hotspot regions

```

temp_ins_1 <- read_delim(
  file = paste0(base_data_path, "compartmentQuantification/TI1.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "section", "score"),
  show_col_types = F
) %>%
  mutate(length = end - start)

temp_ins_2 <- read_delim(
  file = paste0(base_data_path, "compartmentQuantification/TI2.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "section", "score"),
  show_col_types = F
) %>%
  mutate(length = end - start)

chromothripsis_1 <- read_delim(
  file = paste0(base_data_path, "compartmentQuantification/CT1.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "section"),
  show_col_types = F
) %>%
  mutate(length = end - start)

chromothripsis_2 <- read_delim(
  file = paste0(base_data_path, "compartmentQuantification/CT2.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "section"),
  show_col_types = F
)

```

```

) %>%
  mutate(length = end - start)

chromothripsis_3 <- read_delim(
  file = paste0(base_data_path, "compartmentQuantification/CT3.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "section"),
  show_col_types = F
) %>%
  mutate(length = end - start)

```

Calculate the percentage of hotspot length that is within A compartment

```

# TI1 A compartment percentage
temp_ins_1 %>%
  dplyr::filter(section == "a_compartment") %>%
  dplyr::select(length) %>%
  sum() /
  temp_ins_1$length[1] * 100

```

```
## [1] 69.27944
```

```

# TI2 A compartment percentage
temp_ins_2 %>%
  dplyr::filter(section == "a_compartment") %>%
  dplyr::select(length) %>%
  sum() /
  temp_ins_2$length[1] * 100

```

```
## [1] 87.36806
```

```

# CT1 A compartment percentage
chromothripsis_1 %>%
  dplyr::filter(section == "a_compartment") %>%
  dplyr::select(length) %>%
  sum() /
  chromothripsis_1$length[1] * 100

```

```
## [1] 87.7004
```

```

# CT2 A compartment percentage
chromothripsis_2 %>%
  dplyr::filter(section == "a_compartment") %>%
  dplyr::select(length) %>%
  sum() /
  chromothripsis_2$length[1] * 100

```

```
## [1] 58.64679
```

```
# CT3 A compartment percentage
chromothripsis_3 %>%
  dplyr::filter(section == "a_compartment") %>%
  dplyr::select(length) %>%
  sum() /
  chromothripsis_3$length[1] * 100
```

```
## [1] 83.54079
```

Main Analysis - Transcribed Genes in Context of Chromosome 1 Arms

Read in Salmon quantification TPM file

```
salmon_nonzero_tpm <- read_delim(
  file = paste0(base_data_path, "U266_salmon_rna.tpm"),
  delim = "\t",
  skip = 1,
  col_names = c("GENEID", "tpm"),
  show_col_types = F
) %>%
  dplyr::filter(tpm > 0)
salmon_nonzero_tpm$GENEID <- str_remove(salmon_nonzero_tpm$GENEID,
  pattern = "\\..*"
)
```

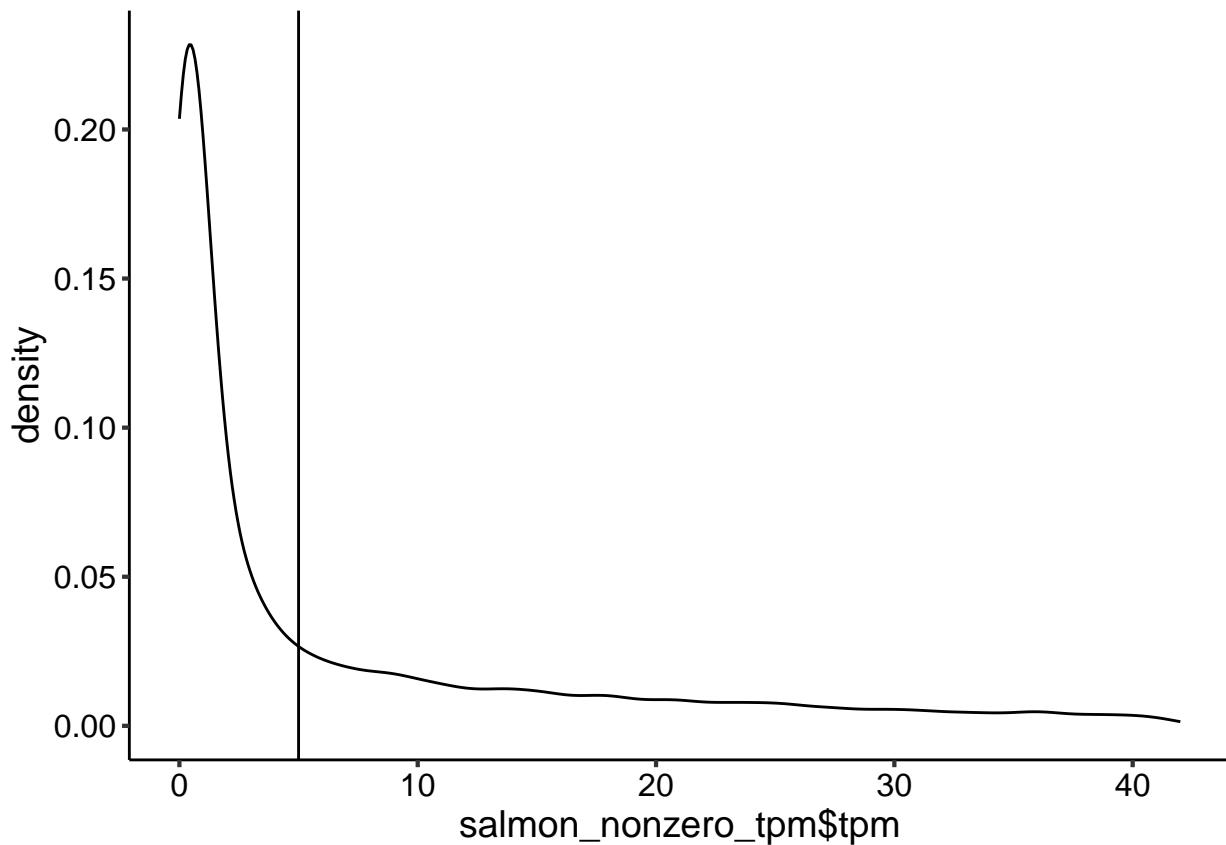
Determine minimum expression threshold via density plot of TMP values

```
summary(salmon_nonzero_tpm$tpm)
```

```
##      Min.    1st Qu.     Median      Mean    3rd Qu.      Max.
## 0.002    0.384    2.540   41.701  18.070 55079.489
```

```
ggplot() +
  geom_vline(xintercept = 5) +
  geom_density(aes(x = salmon_nonzero_tpm$tpm)) +
  xlim(0, 42)
```

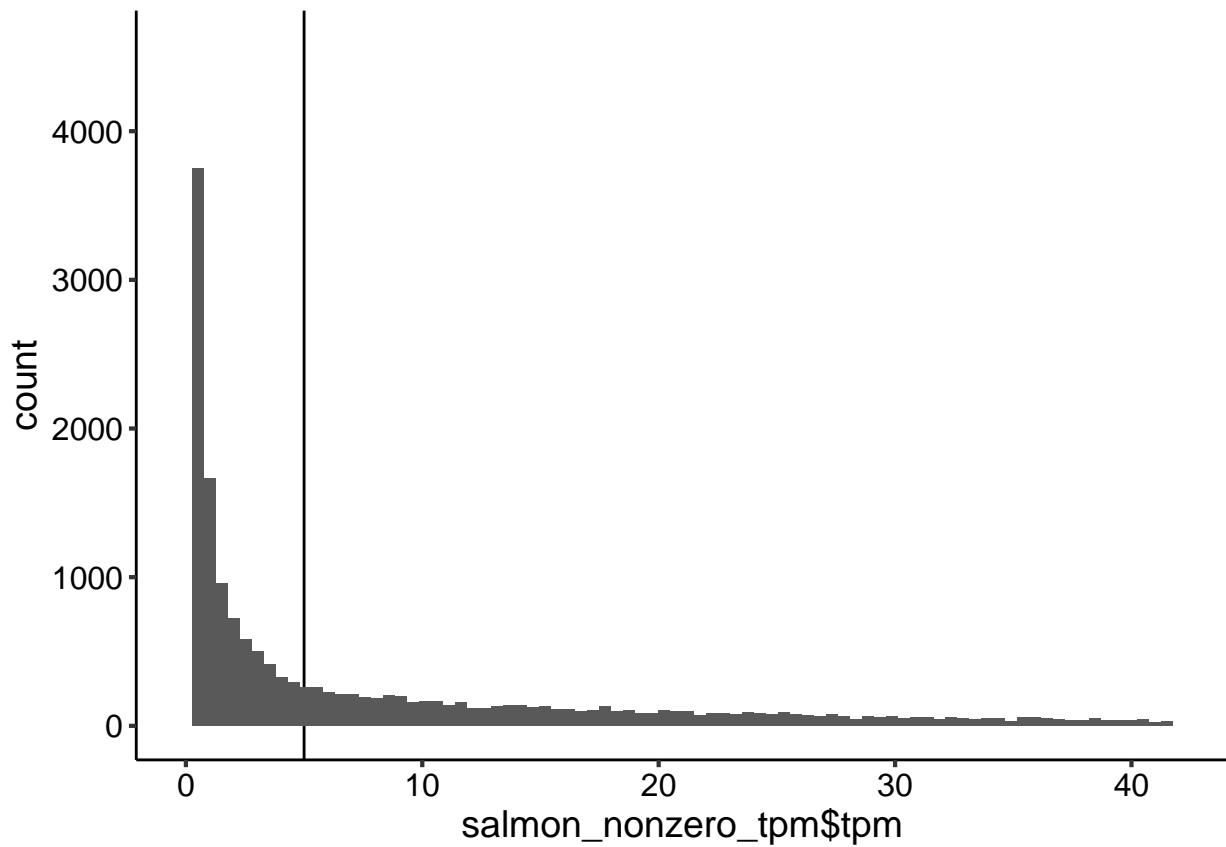
```
## Warning: Removed 3067 rows containing non-finite outside the scale range
## ('stat_density()'').
```



```
ggplot() +
  geom_vline(xintercept = 5) +
  geom_histogram(aes(x = salmon_nonzero_tpm$tpm), bins = 84) +
  xlim(0, 42)
```

```
## Warning: Removed 3067 rows containing non-finite outside the scale range
## ('stat_bin()').
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## ('geom_bar()').
```



```
salmon_threshold_tpm <- salmon_nonzero_tpm %>%
  dplyr::filter(tpm > 5)
```

Convert Salmon quantification TMP file to include gene symbols and start/end locus

```
# anno_hub <- AnnotationHub()
# esdb_101 <- query(x = anno_hub,
#                      pattern = c("EnsDb", "sapiens", "101"))[[1]]
#
# chr1_gene_base_df <- genes(x = esdb_101,
#                               columns = c("symbol", "gene_seq_start", "gene_seq_end", "gene_biotype"),
#                               filter = c(SeqNameFilter("1"),
#                                         GeneBiotypeFilter(c("miRNA", "protein_coding", "snRNA",
#                                         "snoRNA", "rRNA", "lncRNA",
#                                         "transcribed_processed_pseudogene",
#                                         "scRNA")))))
#
# chr1_gene_names <- tibble(gene = chr1_gene_base_df$symbol,
#                            names = chr1_gene_base_df$gene_id)
#
# chr1_gene_info <- as.data.frame(chr1_gene_base_df@ranges)
#
# chr1_gene_bed <- left_join(x = chr1_gene_names,
#                             y = chr1_gene_info,
#                             by = "names") %>%
```

```

#           mutate(chrom = "chr1") %>%
#           select(chrom, start, end, gene, names)
# chr1_gene_bed <- chr1_gene_bed[c(-774, -1144, -1657, -2549, -2776, -3788),]

# write_delim(x = chr1_gene_bed,
#             file = "../data/chr1_genes.bed",
#             delim = "\t",
#             col_names = FALSE)
chr1_gene_bed <- read_delim(
  file = paste0(base_data_path, "chr1_genes.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "gene", "names"),
  show_col_types = F
)

salmon_threshold_tpm_chr1 <- salmon_threshold_tpm %>%
  dplyr::filter(GENEID %in% chr1_gene_bed$names) %>%
  dplyr::rename("names" = GENEID)

transcribed_genes_chr1_bed <- left_join(
  x = chr1_gene_bed[chr1_gene_bed$names %in% salmon_threshold_tpm_chr1$names, ],
  y = salmon_threshold_tpm_chr1,
  by = "names"
)

# write_delim(x = transcribed_genes_chr1_bed,
#             file = "../data/U266_salmon_rna_threshold.chr1.tpm.bed",
#             delim = "\t",
#             col_names = FALSE)

```

Read in transcribed gene BED file per arm to determine number of unique genes and percentage of those genes in A compartments

```

transcribed_genes_chr1p <- read_delim(
  file = paste0(base_data_path, "U266_transcribed_genes.chr1p.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "gene", "ensembl_id", "tpm"),
  show_col_types = F
)

length(unique(transcribed_genes_chr1p$ensembl_id))

```

```
## [1] 538
```

```

transcribed_genes_chr1q <- read_delim(
  file = paste0(base_data_path, "U266_transcribed_genes.chr1q.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "gene", "ensembl_id", "tpm"),
  show_col_types = F
)

length(unique(transcribed_genes_chr1q$ensembl_id))

```

```
## [1] 470
```

Main Analysis - Compartment Quantification in Context of Chromosome 1 Arms

Read in BED files for chromosome 1 A/B compartments

```
a_compartments_1p <- read_delim(  
  file = paste0(base_data_path,  
    "compartmentQuantification/U266-MboI_A_compartments.chr1p.bed"),  
  delim = "\t",  
  col_names = c("chrom", "start", "end"),  
  show_col_types = F  
) %>%  
  mutate(length = end - start)  
  
a_compartments_1q <- read_delim(  
  file = paste0(base_data_path,  
    "compartmentQuantification/U266-MboI_A_compartments.chr1q.bed"),  
  delim = "\t",  
  col_names = c("chrom", "start", "end"),  
  show_col_types = F  
) %>%  
  mutate(length = end - start)
```

Calculate metrics for A compartments across chromosome 1 arms

```
# Median/mean length of 1p A compartments  
median(a_compartments_1p$length)  
  
## [1] 500000  
  
range(a_compartments_1p$length)  
  
## [1] 100000 15300000  
  
# Percentage of 1p arm that is A compartment  
sum(a_compartments_1p$length) / (122026459 - 10000) * 100  
  
## [1] 51.79629  
  
# Median/mean length of 1q A compartments  
median(a_compartments_1q$length)  
  
## [1] 400000  
  
range(a_compartments_1q$length)  
  
## [1] 100000 5700000
```

```
# Percentage of 1q arm that is A compartment
sum(a_compartments_1q$length) / (248946422 - 124932724) * 100
```

```
## [1] 40.12171
```

Create lollipop plot of transcribed genes per A compartment segment per arm

```
lollipop_all_genes_chr1p <- read_delim(
  file = paste0(base_data_path, "chr1p_genes_A_compartment.lollipop"),
  delim = "\t",
  col_names = c("num_of_genes", "start", "end"),
  show_col_types = F
) %>%
  mutate(order = seq(1, 33, 1))

lollipop_chr1p <- read_delim(
  file = paste0(base_data_path,
    "U266_A_compartment_transcribed_genes.chr1p.lollipop"),
  delim = "\t",
  col_names = c("num_of_genes", "start", "end"),
  show_col_types = F
) %>%
  mutate(order = seq(1, 33, 1)) %>%
  mutate(segment = str_c(start, end, sep = " - ")) %>%
  mutate(hotspots = c(
    2, 0, 0, 0, 0, 0, 0, 0, 0, 2, 1,
    0, 2, 0, 0, 0, 0, 0, 0, 1, 1,
    0, 0, 0, 1, 0, 0, 0, 1, 3, 2, 2
  ))

lollipop_chr1p_plot <- ggplot(lollipop_chr1p) +
  geom_hline(
    yintercept = c(0, 10, 50, 100, 150),
    color = "lightgrey"
  ) +
  geom_hline(
    yintercept = 16,
    color = "lightgrey",
    linetype = "dashed"
  ) +
  geom_segment(
    aes(
      x = order, xend = order,
      y = 0, yend = num_of_genes,
      color = as_factor(hotspots)
    ),
    size = 1.25
  ) +
  geom_point(aes(
    x = order,
    y = num_of_genes,
    size = num_of_genes,
    color = as_factor(hotspots)
```

```

)) +
  scale_color_viridis_d(option = "plasma") +
  scale_size(
    range = c(1, 8),
    breaks = c(1, 10, 50, 100),
    labels = c("1-10", "11-50", "51-100", "101-150")
) +
  theme(
    panel.border = element_blank(),
    axis.title.x = element_blank(),
    axis.text.x = element_text(angle = 80, size = 6, vjust = 0.5),
    legend.title = element_blank()
) +
  scale_x_continuous(breaks = seq(1, 33, 1), labels = lollipop_chr1p$segment) +
  scale_y_continuous(breaks = c(0, 10, 50, 100, 150)) +
  annotate(
    geom = "text",
    x = lollipop_chr1p$order,
    y = -5,
    label = replace_na(
      as.double(
        round(lollipop_chr1p$num_of_genes /
              lollipop_all_genes_chr1p$num_of_genes * 100,
              digits = 2
        )
      ),
      replace = 0
    ),
    angle = 80,
    size = 3
) +
  ylab("")

```

```

## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

```

```

lollipop_all_genes_chr1q <- read_delim(
  file = paste0(base_data_path, "chr1q_genes_A_compartment.lollipop"),
  delim = "\t",
  col_names = c("num_of_genes", "start", "end"),
  show_col_types = F
)

lollipop_chr1q <- read_delim(
  file = paste0(base_data_path,
                "U266_A_compartment_transcribed_genes.chr1q.lollipop"),
  delim = "\t",
  col_names = c("num_of_genes", "start", "end"),
  show_col_types = F
) %>%

```

```

    mutate(order = seq(1, 54, 1)) %>%
    mutate(segment = str_c(start, end, sep = " - ")) %>%
    mutate(hotspots = c(
      1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0,
      0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 2, 1, 1, 1, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 1, 1
    ))
  )

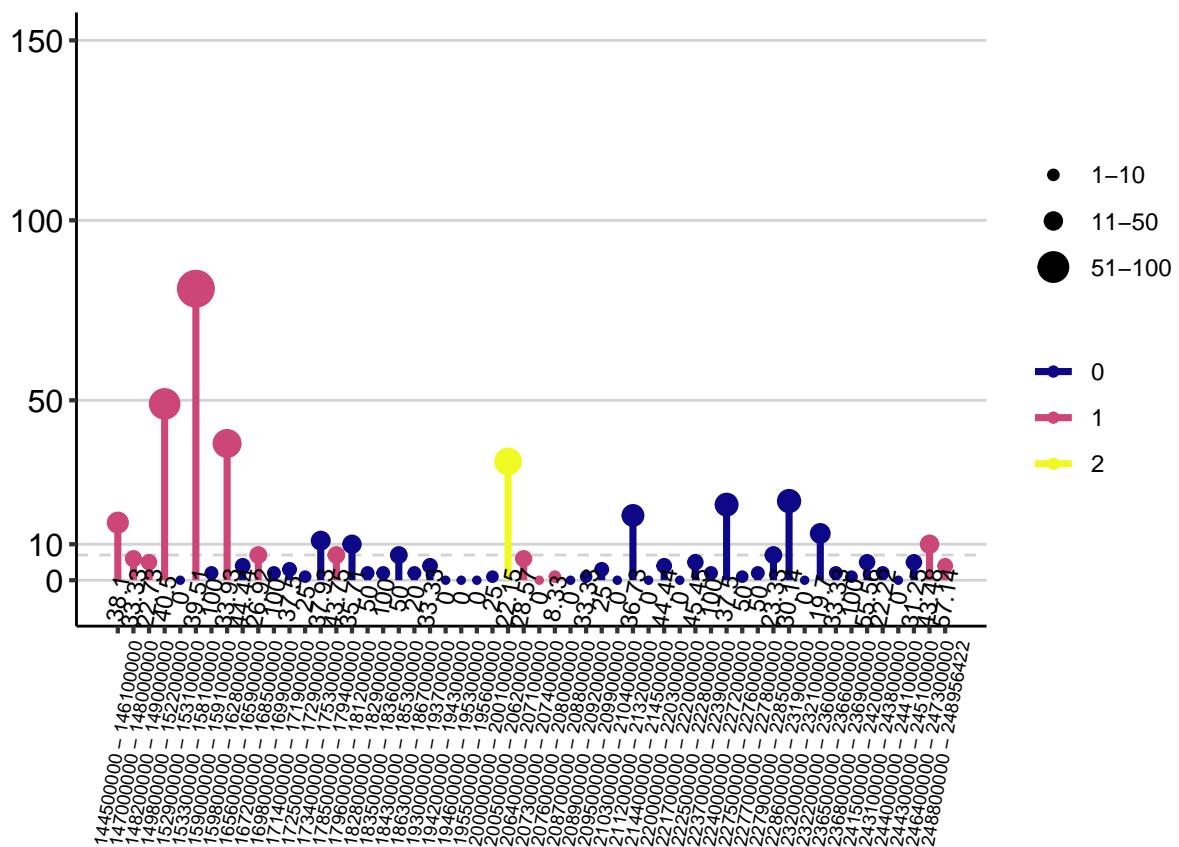
lollipop_chr1q_plot <- ggplot(lollipop_chr1q) +
  geom_hline(
    yintercept = c(0, 10, 50, 100, 150),
    color = "lightgrey"
  ) +
  geom_hline(
    yintercept = 7,
    color = "lightgrey",
    linetype = "dashed"
  ) +
  geom_segment(
    aes(
      x = order, xend = order,
      y = 0, yend = as.numeric(num_of_genes),
      color = as_factor(hotspots)
    ),
    size = 1.25
  ) +
  geom_point(aes(
    x = order,
    y = as.numeric(num_of_genes),
    size = as.numeric(num_of_genes),
    color = as_factor(hotspots)
  )) +
  scale_color_viridis_d(option = "plasma") +
  scale_size(
    range = c(1, 6),
    breaks = c(1, 10, 50, 100),
    labels = c("1-10", "11-50", "51-100", "101-250")
  ) +
  theme(
    panel.border = element_blank(),
    axis.title.x = element_blank(),
    axis.text.x = element_text(angle = 80, size = 7, vjust = 0.5),
    legend.title = element_blank()
  ) +
  scale_x_continuous(breaks = seq(1, 54, 1), labels = lollipop_chr1q$segment) +
  scale_y_continuous(breaks = c(0, 10, 50, 100, 150)) +
  annotate(
    geom = "text",
    x = lollipop_chr1q$order,
    y = -5,
    label = replace_na(

```

```
as.double(
  round(lollipop_chr1q$num_of_genes /
        lollipop_all_genes_chr1q$num_of_genes * 100,
        digits = 2
      )
    ),
  replace = 0
),
angle = 80,
size = 3
) +
ylab("")

# ggarrange(plotlist = list(lollipop_chrip_plot,
#                           lollipop_chr1q_plot),
#           common.legend = TRUE,
#           legend = "right")

#lollipop_chrip_plot
lollipop_chr1q_plot
```



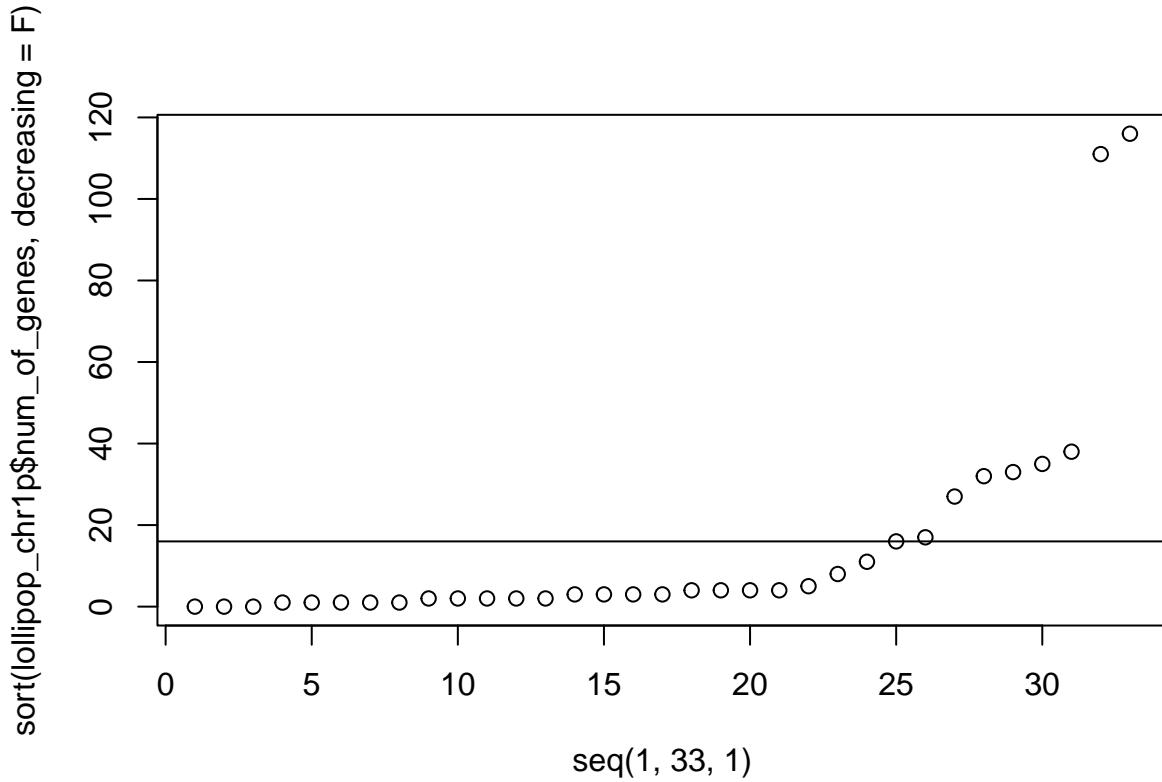
Calculate the concentration of genes in A compartment segments

```
plot(  
  x = seq(1, 33, 1),  
  y = sort(lollipop_chr1p$num_of_genes,
```

```

        decreasing = F
    )
)
abline(h = 16)

```



```

sum(lollipop_chr1p$num_of_genes >= 16)

## [1] 9

sum(sort(lollipop_chr1p$num_of_genes, decreasing = T)[1:9])

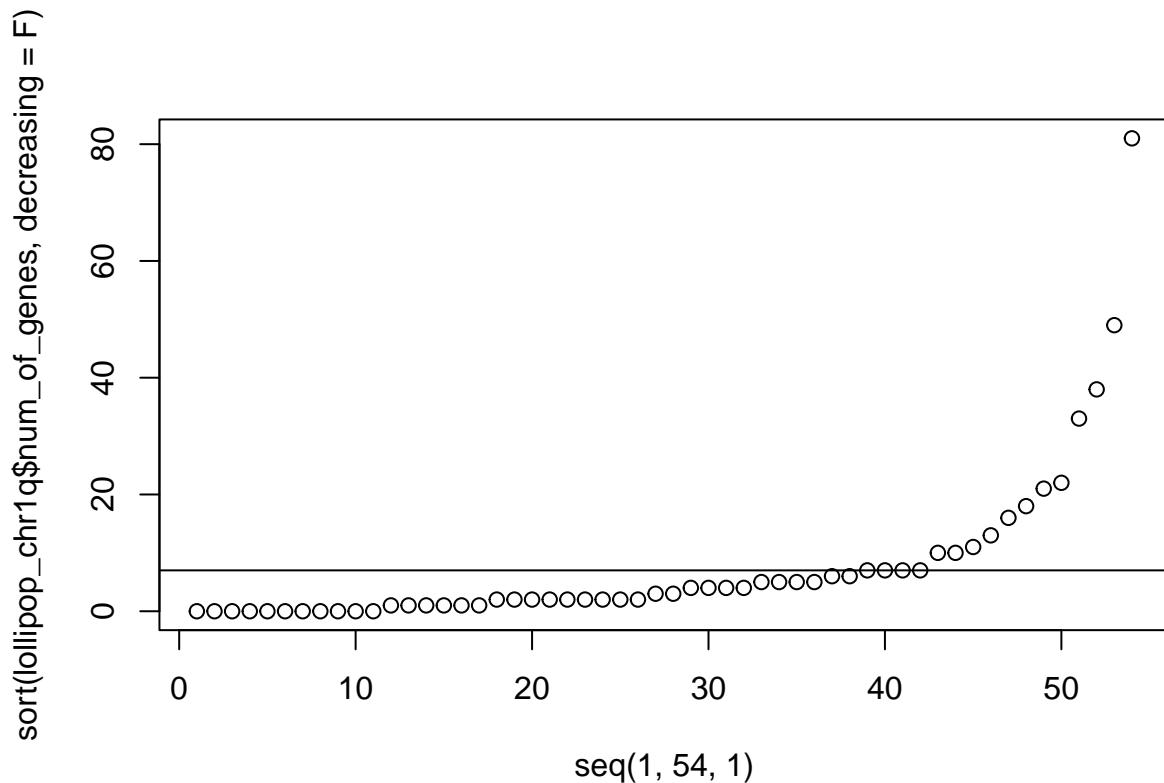
## [1] 425

425 / 498

## [1] 0.8534137

plot(
  x = seq(1, 54, 1),
  y = sort(lollipop_chr1p$num_of_genes,
            decreasing = F
  )
)
abline(h = 7)

```



```

sum(lollipop_chr1q$num_of_genes >= 7)

## [1] 16

sum(sort(lollipop_chr1q$num_of_genes, decreasing = T)[1:16])

## [1] 350

350 / 432

## [1] 0.8101852

```

Main Analysis - Methylation Profile

Import Walker et al methylation array data from GEO and perform methylation analysis with RnBeads. Also read in HM27.hg38 manifest for conversion from hg19 coordinates to hg38

```

# Load the entire methylation dataset from GEO and instantiate the RnBeads object
rnb.options(
  assembly = "hg19",
  logging = FALSE
)

```

```

geo_methylation_array_data <- RnBeads::rnb.read.geo(
  accession = "GSE21304",
  verbose = F,
  destdir = "data/differentialMethylation/"
)

# Set the global RnBeads parameters including filtering out of the sex
# chromosomes from the analysis
rnb.options(
  filtering.sex.chromosomesremoval = TRUE,
  identifiers.column = "title"
)

# Set some of the analysis groups for visualization
core_groups <- c(
  rep("BC", 6),
  rep("MGUS", 4),
  rep("MM", 161),
  rep("PC", 3),
  rep("PCL", 7),
  rep("HMCL", 9)
)
core_groups <- factor(core_groups, levels = c("BC", "PC", "MGUS", "MM", "PCL", "HMCL"))

methyl_data <- addPheno(
  object = geo_methylation_array_data,
  core_groups, "core_groups"
)

# The preprocessing includes a first pass filter that removes probes overlapping SNPs
# that stand a high chance of influencing DNA methylation measurements.
# No normalization is performed as this was already done for this data before
# depositing on GEO. Finally, the probes on the sex chromosomes are filtered out
preprocessed_methyl_data <- rnb.run.preprocessing(
  rnb.set = methyl_data,
  dir.reports = "data/differentialMethylation/"
)

## Saved removed sites to /Users/blanep01/morganLab/multiomicMappingOfChr1Analysis/scripts/Clin-Cancer-
## Omitting GreedyCut because detection p-values are missing

## Coordinate system already present. Adding new coordinate system, which will
## replace the existing one.

## Warning: Incompatible dataset and normalization method: missing intensity data.
## Changed the method to "none"

## Performed background subtraction with method sesame.noobsb
## Coordinate system already present. Adding new coordinate system, which will
## replace the existing one.Saved removed sites to /Users/blanep01/morganLab/multiomicMappingOfChr1Anal
## Coordinate system already present. Adding new coordinate system, which will
## replace the existing one.

```

```

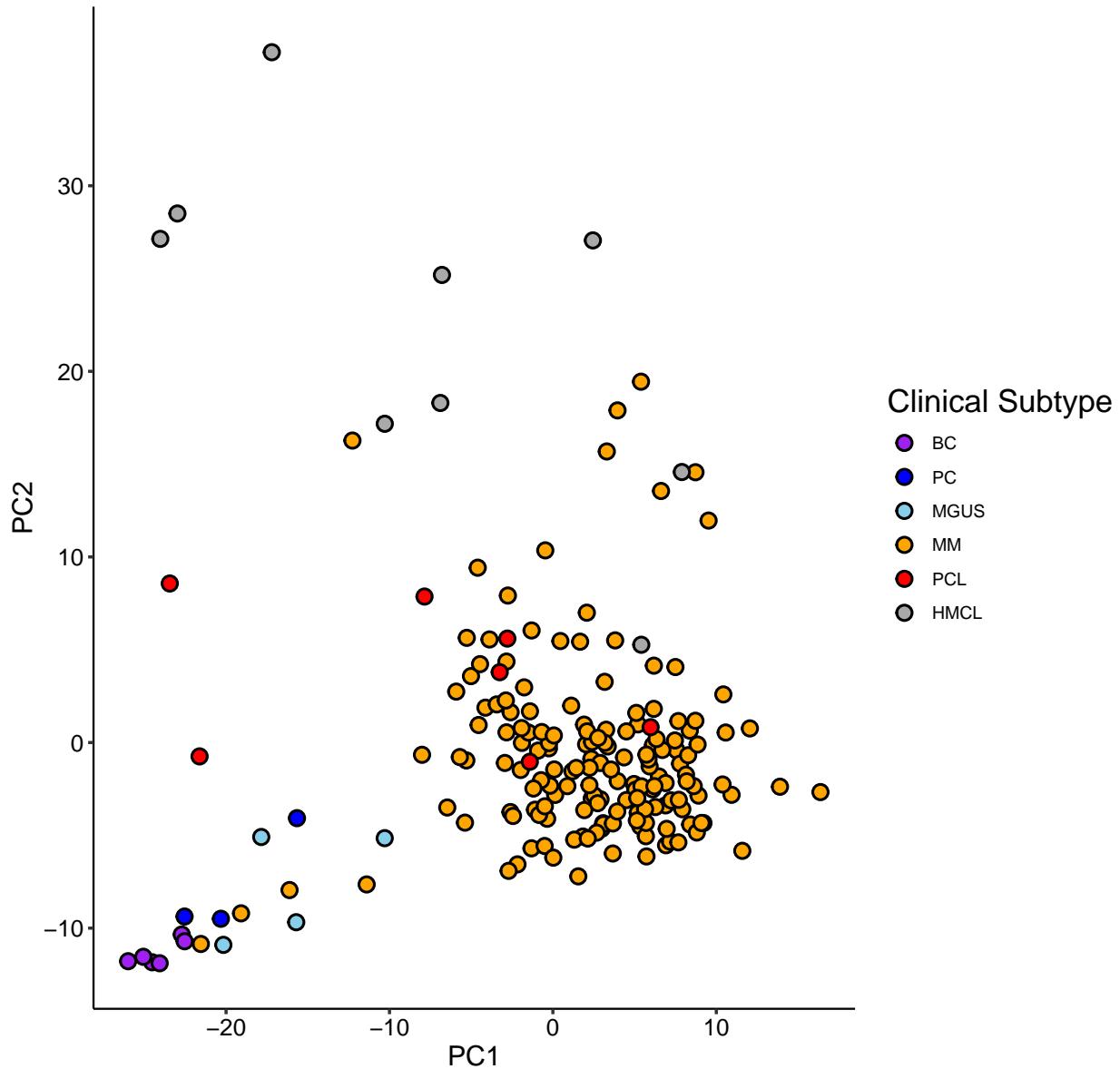
# Run a PCA on the complete genome-wide set of probes across all the core groups
genome_wide_pca_df <- rnb.execute.dreduction(rnb.set = preprocessed_methyl_data$rnb.set,
                                              target = "sites")$pca$x[, 1:2] %>%
  as.data.frame() %>%
  rownames_to_column(var = "sample") %>%
  dplyr::mutate("core_group" = core_groups)

## Calculated MDS coordinates using manhattan distance
## Calculated MDS coordinates using euclidean distance
## Removed 1770 loci ( sites ) because they contain missing values

ggplot(data = genome_wide_pca_df) +
  geom_point(aes(x = PC1, y = PC2, fill = core_group),
             shape = 21, size = 3, color = "black", stroke = 1) +
  scale_fill_manual(values = c("purple", "blue", "skyblue",
                               "orange", "red", "darkgrey")) +
  labs(
    title = "Genome-Wide Methylation",
    fill = "Clinical Subtype"
)

```

Genome-Wide Methylation



```
# Now subset the analysis to only probes on chromosome 1
# First load hg38 reference probes and their coordinates to ensure only
# analyzing hg38 compatible probes
hm27_hg38_manifest <- read_delim(
  file = "data/differentialMethylation/HM27.hg38.manifest.gencode.v36.tsv.gz",
  delim = "\t",
  col_names = T,
  show_col_types = F
)

chr1_probes_hm27_hg38 <- hm27_hg38_manifest %>%
  dplyr::filter(CpG_chrm == "chr1") %>%
  dplyr::select(probeID)
```

```

sites_to_remove <- !row.names(sites(preprocessed_methyl_data$rnb.set)) %in%
  chr1_probes_hm27_hg38$probeID

chr1_methyl_data <- remove.sites(
  object = preprocessed_methyl_data$rnb.set,
  probelist = sites_to_remove
)

# Rerun a PCA on the chromosome 1 set of probes across the groups
chr1_pca_df <- rnb.execute.dreduction(rnb.set = chr1_methyl_data,
                                         target = "sites")$pca$x[, 1:2] %>%
  as.data.frame() %>%
  rownames_to_column(var = "sample") %>%
  dplyr::mutate("core_group" = core_groups)

## Calculated MDS coordinates using manhattan distance
## Calculated MDS coordinates using euclidean distance
## Removed 219 loci ( sites ) because they contain missing values

core_group_pca <- ggplot(data = chr1_pca_df) +
  geom_point(aes(x = PC1, y = PC2, fill = core_group), shape = 21, size = 3.5,
             color = "black", stroke = 1.5) +
  scale_fill_manual(values = c("purple", "blue", "skyblue",
                             "orange", "red", "darkgray")) +
  labs(
    title = "Clinical Subtype",
    fill = ""
  )

# Run the PCA visualization by subtype, cell line annotations from Keat's lab
patient_subtype_anno <- geo_methylation_array_data@pheno[["myeloma translocation"]][11:171] %>%
  str_replace("No split", "No Translocation") %>%
  str_replace("Not known", "Unknown") %>%
  str_replace("t\\\\(\\\\?\\\\;14\\\\)", "Unknown") %>%
  replace_na(replace = "Unknown")

hmcl_subtype_anno <- c(
  "t(16;22)", "t(11;14)", "t(4;14)",
  "t(4;14)", "t(11;14)", "t(4;14)",
  "t(4;14)", "t(14;16)", "t(14;16)"
)

subtype_groups <- c(
  rep("Unknown", 10),
  patient_subtype_anno,
  rep("Unknown", 10),
  hmcl_subtype_anno
)
subtype_groups <- factor(subtype_groups, levels = c(
  "t(11;14)", "t(4;14)", "t(14;16)", "t(6;14)",
  "t(14;20)", "t(16;22)", "No Translocation", "Unknown"
))

```

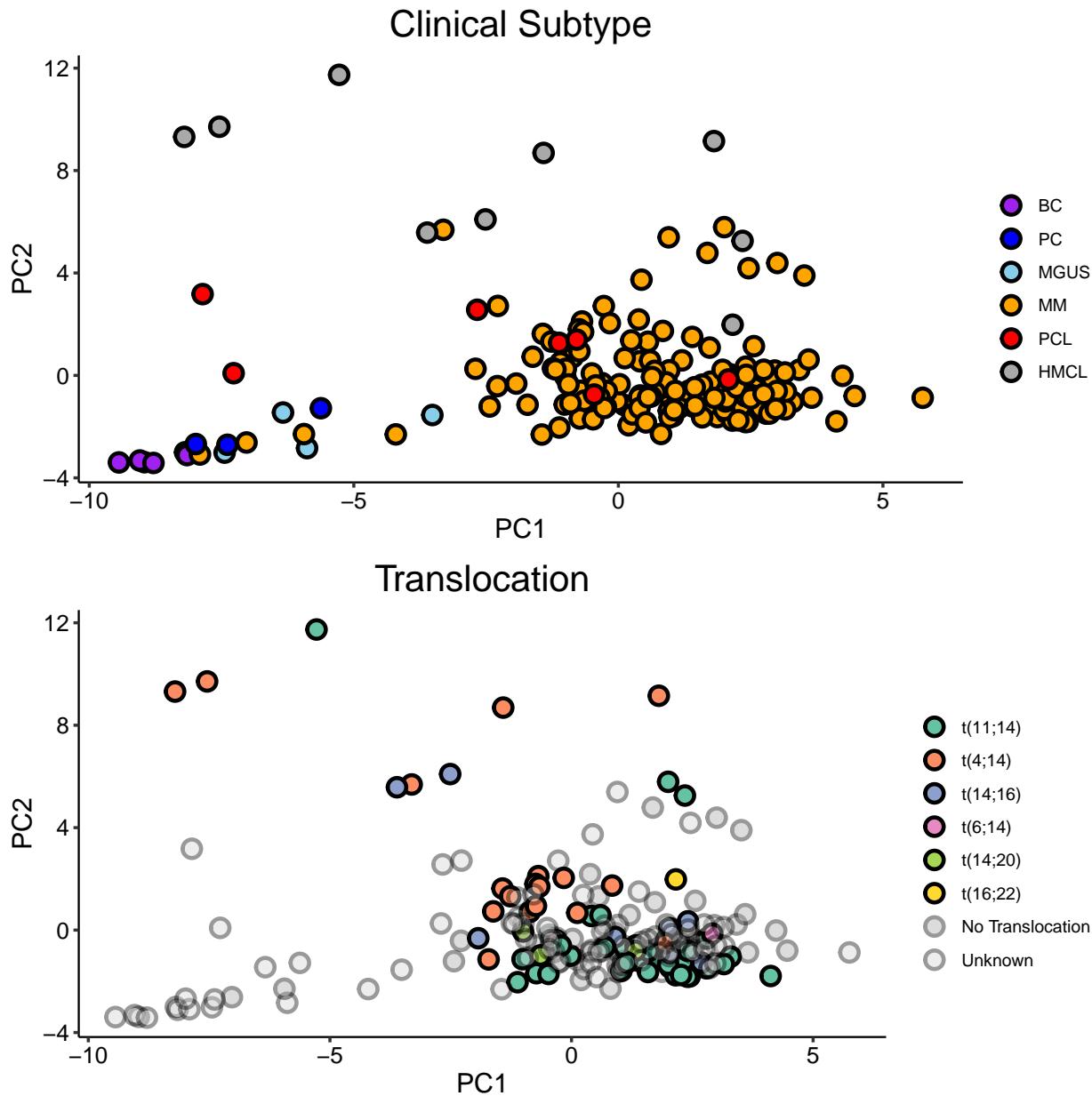
```

chr1_pca_df$subtype_groups <- subtype_groups
subtype_groups_colors <- c(RColorBrewer::brewer.pal(n = 6, name = "Set2"),
                           "darkgray", "lightgray")

translocation_pca <- ggplot(chr1_pca_df) +
  geom_point(aes(x = PC1, y = PC2, fill = subtype_groups, alpha = subtype_groups),
             shape = 21, size = 3.5, color = "black", stroke = 1.5
  ) +
  scale_fill_manual(name = "", values = subtype_groups_colors) +
  scale_alpha_manual(name = "", values = c(1, 1, 1, 1, 1, 0.4)) +
  labs(title = "Translocation")

# Group the two chr1 PCAs together
ggarrange(
  plotlist = list(core_group_pca, translocation_pca),
  ncol = 1,
  legend = "right"
)

```



```
# Export at 6x13 landscape
```

```
# Run the exploratory analysis to produce clustered heatmap of chr1 methylation
# First is coordinate sorted
group_anno <- HeatmapAnnotation(
  "Clinical Subtype" = core_groups,
  "Translocation" = subtype_groups,
  col = list(
    "Clinical Subtype" = c(
      "BC" = "purple",
      "PC" = "blue",
      "MGUS" = "skyblue",
      "MM" = "orange",
      "PCL" = "red",
      "HMCL" = "grey"
    )
  )
)
```

```

        "HMCL" = "darkgray"
    ),
    "Translocation" = c(
        "t(11;14)" = subtype_groups_colors[1],
        "t(4;14)" = subtype_groups_colors[2],
        "t(14;16)" = subtype_groups_colors[3],
        "t(6;14)" = subtype_groups_colors[4],
        "t(14;20)" = subtype_groups_colors[5],
        "t(16;22)" = subtype_groups_colors[6],
        "No Translocation" = "#E5C494",
        "Unknown" = "black"
    )
)
)

# Cytoband annotations for genomic locus reference
cytoband_markers <- c(
    "1p36.3", "1p35", "1p34", "1p32", "1p22", "1p21", "1p12",
    "1q21", "1q22", "1q23", "1q24", "1q25", "1q32", "1q44"
)
cytoband_start_points <- c(
    1, 475, 592, 852, 1063, 1151, 1350,
    1373, 1622, 1703, 1872, 1945, 2090, 2475
)

# Recurrent regions annotations, (need to find all probes that fall within each region)
recurrent_regions <- read_delim(
    file = "data/differentialMethylation/recurrentRegionsForHeatmapAnno.txt",
    delim = "\t",
    col_names = T,
    show_col_types = F
)

probe_loci <- hm27_hg38_manifest %>%
    dplyr::filter(probeID %in% rownames(chr1_methyl_data@sites)) %>%
    dplyr::select(CpG_beg)

recurrent_region_anno_labels <- c()
for (i in 1:length(probe_loci$CpG_beg)) {
    query_probe <- probe_loci$CpG_beg[i]

    for (j in 1:nrow(recurrent_regions)) {
        region_to_test <- recurrent_regions[j, ]

        if (between(x = query_probe, left = region_to_test$start,
                    right = region_to_test$end)) {
            recurrent_region_anno_labels[i] <- region_to_test$anno
            break
        } else {
            recurrent_region_anno_labels[i] <- "No"
        }
    }
}

```

```

recurrent_region_anno_labels <- factor(
  x = recurrent_region_anno_labels,
  levels = c(
    "Deletion", "Gain", "Templated Insertion",
    "Chromothripsis", "Multiple", "No"
  )
)

recurrent_region_anno <- HeatmapAnnotation(
  cytoband = anno_mark(
    at = cytoband_start_points,
    labels = cytoband_markers,
    side = "left",
    which = "row"
  ),
  "Recurrent Region" = recurrent_region_anno_labels,
  show_annotation_name = F,
  which = "row",
  col = list("Recurrent Region" = c(
    "Deletion" = "#0C06E8",
    "Gain" = "#FA160C",
    "Templated Insertion" = "#FF6307",
    "Chromothripsis" = "#A10CE8",
    "Multiple" = "#00D105",
    "No" = "lightgray"
  ))
)
)

heatmap_cols <- colorRampPalette(colors = c("dodgerblue", "white", "orangered"))

chr1_methyl_heatmap <- Heatmap(
  matrix = as.matrix.data.frame(as.data.frame.matrix(chr1_methyl_data@meth.sites)),
  col = heatmap_cols(3),
  cluster_rows = F,
  clustering_distance_columns = "euclidean",
  clustering_method_columns = "average",
  column_dend_height = unit(45, "mm"),
  column_names_gp = gpar(fontsize = 0),
  row_order = c(1:nrow(chr1_methyl_data@meth.sites)),
  heatmap_legend_param = list(title = "Beta Value"),
  top_annotation = group_anno,
  left_annotation = recurrent_region_anno,
  heatmap_width = unit(300, "mm"),
  heatmap_height = unit(200, "mm")
)

## 'use_raster' is automatically set to TRUE for a matrix with more than
## 2000 rows. You can control 'use_raster' argument by explicitly setting
## TRUE/FALSE to it.
##
## Set 'ht_opt$message = FALSE' to turn off this message.

```

Read in differential methylation table and convert coordinates from hg19 to hg38 for analysis

```
mm_vs_pc_chr1_diffmethyl_table <- read_delim(  
    file = "data/differentialMethylation/myelo
```

```

    delim = ",",
    col_names = T,
    show_col_types = F
) %>%
  dplyr::rename("probeID" = cgid)

mm_vs_pc_chr1_diffmethyl_table_hg38 <- left_join(
  x = mm_vs_pc_chr1_diffmethyl_table,
  y = hm27_hg38_manifest,
  by = "probeID"
) %>%
  dplyr::filter(Chromosome == "chr1" & CpG_chrm == "chr1") %>%
  dplyr::select(!c(id, Start, Strand, CpG_chrm))

```

Visualize various metrics of the differential methylation data

```

# First create volcano plot of all chr1 probes that are statistically differentially methylated
passing_diff_meth_probes <- mm_vs_pc_chr1_diffmethyl_table_hg38 %>%
  dplyr::filter(mean.diff < -0.25 | mean.diff > 0.25) %>%
  dplyr::filter(diffmeth.p.val < 0.05)
passing_diff_meth_probes <- passing_diff_meth_probes[, -c(6, 8:14, 17:18, 23)]

# Write table of passing differentially methylated probes
# write_delim(x = passing_diff_meth_probes,
#             file = "data/differentialMethylation/mm_vs_pc_diffmethyl_probes.txt",
#             delim = "\t",
#             col_names = T)

# Total number of differentially hypo/hyper-methylated probes/genes in MM vs PC
passing_diff_meth_probes %>%
  dplyr::filter(mean.diff > 0.0) %>%
  nrow() %>%
  paste("Hypomethylated Probes", sep = " ")

## [1] "378 Hypomethylated Probes"

passing_diff_meth_probes %>%
  dplyr::filter(mean.diff < 0.0) %>%
  nrow() %>%
  paste("Hypermethylated Probes", sep = " ")

## [1] "5 Hypermethylated Probes"

passing_diff_meth_probes %>%
  dplyr::filter(mean.diff > 0.0) %>%
  dplyr::select(genesUniq) %>%
  as_vector() %>%
  unique() %>%
  str_split(pattern = ";") %>%
  as_vector() %>%
  length() %>%
  paste("Hypomethylated Genes", sep = " ")

```

```

## [1] "360 Hypomethylated Genes"

passing_diff_meth_probes %>%
  dplyr::filter(mean.diff < 0.0) %>%
  dplyr::select(genesUniq) %>%
  as_vector() %>%
  unique() %>%
  str_split(pattern = ";") %>%
  as_vector() %>%
  length() %>%
  paste("Hypermethylated Genes", sep = " ")

## [1] "7 Hypermethylated Genes"

# Query different gene sets against the volcano plot to see where genes lie
# 1) GEP70 chr1 genes, 2) previously implicated drivers
gep70_chr1_geneset <- read_delim(
  file = "data/differentialMethylation/gep70Chr1GeneSet.txt",
  delim = "\t",
  col_names = T,
  show_col_types = F
)

previously_implicated_geneset <- read_delim(
  file = "data/differentialMethylation/PreviouslyImplicatedDriversChr1GeneSet.txt",
  delim = "\t",
  col_names = T,
  show_col_types = F
)

overexpressed_gain1q_geneset <- read_delim(
  file = "data/differentialMethylation/expressionComparision_1q_no_1q.txt",
  delim = "\t",
  col_names = T,
  show_col_types = F
) %>%
  dplyr::filter(p.adj_mn <= 0.05) %>%
  dplyr::filter(foldchange_NG > 0) %>%
  dplyr::filter(!is.na(LOCUS)) %>%
  dplyr::rename("Gene" = ID)

# COSMIC Cancer Census - Found all current oncogenes in v96 cosmic release
# grep 'oncogene' cancer_gene_census.csv \
# | cut -d ',' -f 1 \
# | sed 's/"//g' \
# | sort > cosmicCancerOncogeneCensus.v96.txt
cosmic_oncogenes <- read_delim(
  file = "data/differentialMethylation/cosmicCancerOncogeneCensus.v96.txt",
  delim = "\t",
  col_names = "Gene",
  show_col_types = F
)

```

```

t4_14_and_t11_14_gain1q_common <- read_delim(
  file = "data/differentialMethylation/t4_14Andt11_14WithGain1qOverlapGeneSet.txt",
  delim = "\t",
  col_names = T,
  show_col_types = F
)

# Create function that will link a gene set of interest to the methylation probes
# in the heatmap
volcano_plot_annotation <- function(gene_list, full_or_diff = "full") {
  # Identify the probes linked to the genes in the query gene list
  labels_of_interest <- c()
  for (i in 1:length(mm_vs_pc_chr1_diffmethyl_table_hg38$genesUniq)) {
    # To focus the label annotation on only diff. hypomethylated probes
    if (full_or_diff == "diff") {
      passing_diff_hypomethyl_probes <- passing_diff_meth_probes %>%
        dplyr::filter(mean.diff > 0) %>%
        dplyr::filter(diffmeth.p.val < 0.05)

      # First, find all genes common between the full table and the
      # hypomethylated probes
      if (mm_vs_pc_chr1_diffmethyl_table_hg38$probeID[i] %in%
          passing_diff_hypomethyl_probes$probeID &&
          !is.na(mm_vs_pc_chr1_diffmethyl_table_hg38$genesUniq[i])) {
        # Then extract the individual genes attributed to each probe
        genes_attrib_to_probe <- str_split(
          string = mm_vs_pc_chr1_diffmethyl_table_hg38$genesUniq[i],
          pattern = "\\|;",
          simplify = T
        )

        # Finally, test if the separated genes are in the query gene set,
        # if test is true add gene name to output
        # if test is false, write placeholder to output
        if (sum(genes_attrib_to_probe %in% gene_list$Gene) > 0) {
          correct_gene_to_label <- genes_attrib_to_probe %in%
            gene_list$Gene %>% which()

          # Will need to combine multiple gene hits into a single label
          if (length(correct_gene_to_label) >= 2) {
            newly_combined_gene_label <- paste0(
              genes_attrib_to_probe[, correct_gene_to_label],
              collapse = ";")
            labels_of_interest[i] <- newly_combined_gene_label
          } else {
            labels_of_interest[i] <- genes_attrib_to_probe[, correct_gene_to_label]
          }
        } else {
          labels_of_interest[i] <- "X"
        }
      } else {
        labels_of_interest[i] <- "X"
      }
    }
  }
}

```

```

# To focus the label annotation on all probes
} else if (full_or_diff == "full") {
    # First extract the individual genes attributed to each probe
    genes_attrib_to_probe <- str_split(
        string = mm_vs_pc_chr1_diffmethyl_table_hg38$genesUniq[i],
        pattern = "\\|;",
        simplify = T
    )

    # Test if the separated genes are in the query gene set, if test is
    # true add gene name to output
    # if test is false, write placeholder to output
    if (sum(genes_attrib_to_probe %in% gene_list$Gene) > 0) {
        correct_gene_to_label <- genes_attrib_to_probe %in%
            gene_list$Gene %>% which()

        # Will need to combine multiple gene hits into a single label
        if (length(correct_gene_to_label) >= 2) {
            newly_combined_gene_label <- paste0(
                genes_attrib_to_probe[, correct_gene_to_label],
                collapse = ";")
            labels_of_interest[i] <- newly_combined_gene_label
        } else {
            labels_of_interest[i] <- genes_attrib_to_probe[, correct_gene_to_label]
        }
    } else {
        labels_of_interest[i] <- "X"
    }
}
}

return(labels_of_interest)
}

# Create the set of labels for each gene list
 gep70_chr1_labels <- volcano_plot_annotator(
    gene_list = gep70_chr1_geneset,
    full_or_diff = "full"
)

previously_implicated_chr1_labels <- volcano_plot_annotator(
    gene_list = previously_implicated_geneset,
    full_or_diff = "full"
)

overexpressed_gain1q_labels <- volcano_plot_annotator(
    gene_list = overexpressed_gain1q_geneset,
    full_or_diff = "diff"
)

t4_14_and_t11_14_gain1q_common_labels <- volcano_plot_annotator(
    gene_list = t4_14_and_t11_14_gain1q_common,
    full_or_diff = "diff"
)

```

```

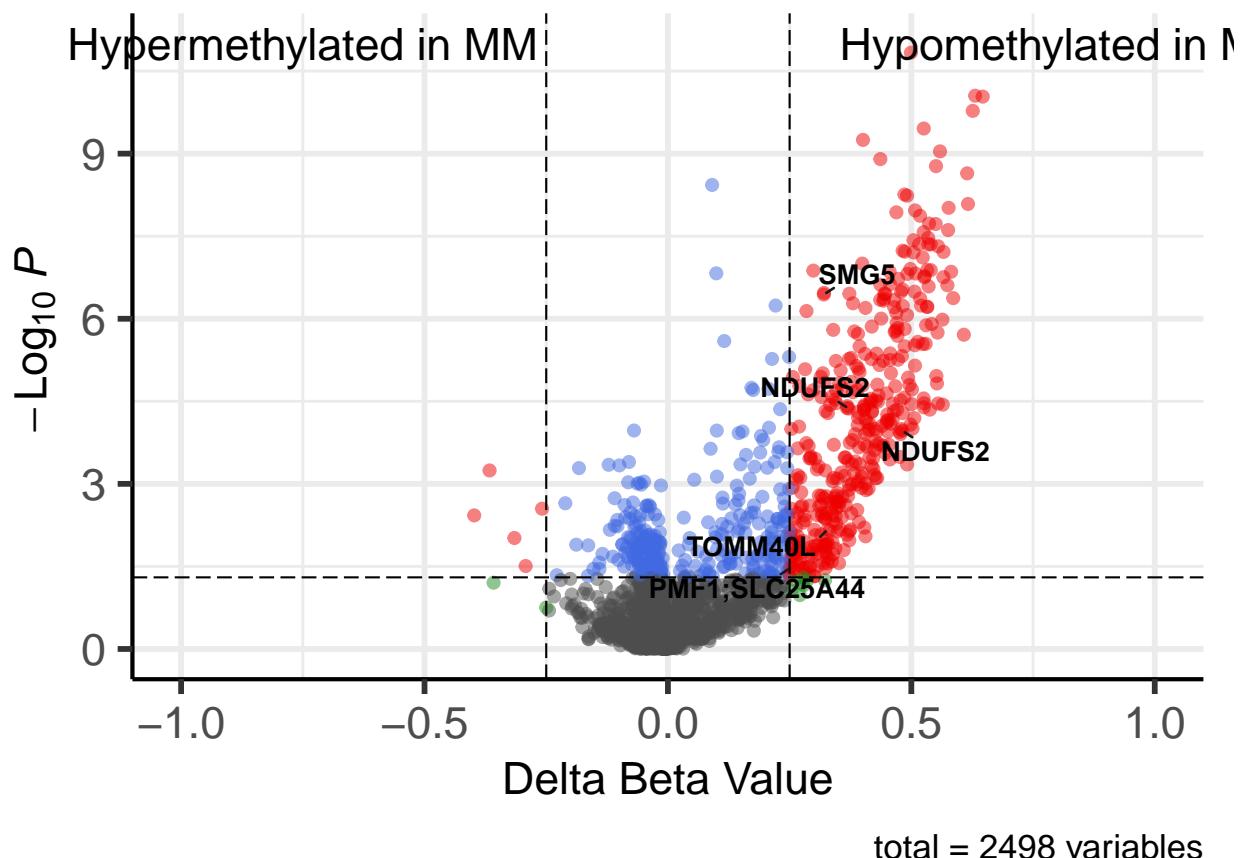
# Generate a series of volcano plots by swapping out the labels for each gene list
EnhancedVolcano(
  toptable = mm_vs_pc_chr1_diffmethyl_table_hg38,
  lab = t4_14_and_t11_14_gain1q_common_labels,
  selectLab = t4_14_and_t11_14_gain1q_common_labels[t4_14_and_t11_14_gain1q_common_labels != "X"],
  drawConnectors = T,
  labSize = 4,
  arrowheads = F,
  x = "mean.diff",
  y = "diffmeth.p.val",
  xlim = c(-1, 1),
  ylim = c(0, 11),
  pCutoff = 0.05,
  FCCutoff = 0.25,
  xlab = "Delta Beta Value",
  legendPosition = "top",
  title = "Common Overexpressed Genes in t(4;14)s and \nt(11;14)s with Gain(1q) (n=5)",
  legendLabels = c("NS", "Delta Beta >|0.25|", "p-value <0.05", "p-value and Delta Beta"),
  maxoverlapsConnectors = 1000,
  labFace = "bold"
) +
  annotate(
    geom = "text",
    label = c("Hypomethylated in MM", "Hypermethylated in MM"),
    x = c(0.82, -0.75),
    y = c(10.99, 10.99),
    size = c(6.5, 6.5),
    color = "black"
)

```

Common Overexpressed Genes in t(4;14)s and t(11;14)s with Gain(1q) (n=5)

EnhancedVolcano

● NS ● Delta Beta >|0.25| ● p-value <0.05 ● p-value and Delta E



```
# Export at 9x11 landscape

# Find number of overexpressed genes that are diff. hypomethylated
overexpressed_gain1q_labels[overexpressed_gain1q_labels != "X"] %>%
  str_split(pattern = ";") %>%
  as_vector() %>%
  unique() %>%
  length() %>%
  paste("Hypomethylated Genes that are overexpressed in Gain(1q) MM", sep = " ")
```

```
## [1] "62 Hypomethylated Genes that are overexpressed in Gain(1q) MM"
```

```

# Write table of probes for overexpressed and hypomethylated genes
mm_hypometh_overexpressed_probes_table <-
  mm_vs_pc_chr1_diffmethyl_table_hg38[overexpressed_gain1q_labels != "X", ] %>%
  dplyr::mutate("Overexpressed.Gene" = overexpressed_gain1q_labels[overexpressed_gain1q_labels != "X"])
# write_delim(x = mm_hypometh_overexpressed_probes_table,
#   file = "../data/differentialMethylation/revision2/working/mm_gain1q_hypomethyl_overexpressed_probes",
#   delim = "\t",
#   col_names = T)

# Find overexpressed genes in gain(1q) that are confirmed oncogenes in COSMIC Cancer
# Gene Census v96 and hypomethylated in MM
cosmic_oncogenes$Gene[cosmic_oncogenes$Gene %in% overexpressed_gain1q_geneset$Gene]

## [1] "ABL2"    "ARNT"    "BCL9"    "DDR2"    "ELK4"    "MDM4"    "NTRK1"   "SETDB1"

# Now revisualize the probe heatmap across chromosome 1 using only the passing
# differentially methylated probes
diffmethyl_group_anno <- HeatmapAnnotation(
  "Clinical Subtype" = core_groups[core_groups %in% c("PC", "MM")],
  "Translocation" = subtype_groups[11:174],
  col = list(
    "Clinical Subtype" = c(
      "PC" = "blue",
      "MM" = "orange"
    ),
    "Translocation" = c(
      "t(11;14)" = subtype_groups_colors[1],
      "t(4;14)" = subtype_groups_colors[2],
      "t(14;16)" = subtype_groups_colors[3],
      "t(6;14)" = subtype_groups_colors[4],
      "t(14;20)" = subtype_groups_colors[5],
      "t(16;22)" = subtype_groups_colors[6],
      "No Translocation" = "#E5C494",
      "Unknown" = "black"
    )
  )
)

# Reannotate the cytoband references
diffmethyl_cytoband_markers <- c(
  "1p36.3", "1p35.3", "1p34.3", "1p32.3", "1p22.3", "1p21.3", "1p12",
  "1q21.1", "1q21.3", "1q22", "1q23.1", "1q24.1", "1q25.1", "1q32.1", "1q44"
)
diffmethyl_cytoband_start_points <- c(
  1, 88, 101, 132, 148, 160, 184,
  189, 195, 252, 259, 306, 318, 334, 382
)

# Reannotate the recurrent regions (near duplication of previous work)
diffmethyl_recurrent_region_anno_labels <- tibble(
  "anno" = as.character(recurrent_region_anno_labels),
  "probeID" = rownames(chr1_methyl_data@sites)
) %>%

```

```

filter(probeID %in% passing_diff_meth_probes$probeID)

diffmethyl_recurrent_region_anno_labels$anno <- factor(
  x = diffmethyl_recurrent_region_anno_labels$anno,
  levels = c(
    "Deletion", "Gain", "Templated Insertion",
    "Chromothripsis", "Multiple", "No"
  )
)

diffmethyl_recurrent_region_anno <- HeatmapAnnotation(
  cytoband = anno_mark(
    at = diffmethyl_cytoband_start_points,
    labels = diffmethyl_cytoband_markers,
    side = "left",
    which = "row"
  ),
  "Recurrent Region" = diffmethyl_recurrent_region_anno_labels$anno,
  show_annotation_name = F,
  which = "row",
  col = list("Recurrent Region" = c(
    "Deletion" = "#OC06E8",
    "Gain" = "#FA160C",
    "Templated Insertion" = "#FF6307",
    "Chromothripsis" = "#A10CE8",
    "Multiple" = "#00D105",
    "No" = "lightgray"
  ))
)

mm_vs_pc_heatmap_main <- Heatmap(
  matrix = as.matrix.data.frame(
    as.data.frame.matrix(
      chr1_methyl_data@methyl.sites[rownames(chr1_methyl_data@sites) %in%
        passing_diff_meth_probes$probeID, ])[, 11:174],
    col = heatmap_cols(3),
    cluster_rows = F,
    clustering_method_columns = "average",
    column_dend_height = unit(30, "mm"),
    column_names_gp = gpar(fontsize = 0),
    heatmap_legend_param = list(title = "Beta Value"),
    top_annotation = diffmethyl_group_anno,
    left_annotation = diffmethyl_recurrent_region_anno
  )
)

diffmethyl_heatmap <- mm_vs_pc_heatmap_main + Heatmap(
  matrix = as.matrix.data.frame(passing_diff_meth_probes[, 5]),
  col = RColorBrewer::brewer.pal(n = 3, name = "PiYG"),
  cluster_rows = F,
  cluster_columns = F,
  column_names_gp = gpar(fontsize = 0),
  heatmap_legend_param = list(title = "Mean Diff MM vs PC"),
  heatmap_width = unit(10, "mm")
)

```

```

)
# draw(diffmethyl_heatmap)
# Export at 10x14 landscape

# Now combine the chromosome ideogram with hypomethylated probes on top and ratio of
# diff methyl probes per overall probes in a window
recurrent_regions_midpoint_indicators <- read_delim(
  file = "data/gisticRegionsWithSvHotspots.hg38.bed",
  delim = "\t",
  col_names = c("chrom", "start", "end", "id"),
  show_col_types = F
) %>%
  dplyr::mutate("midpoint" = round(start + (end - start) / 2, digits = 0))

# Final all genes that are DE in gain(1q) patients and convert them to GRanges
# objects for plotting
hypometh_and_overexp_loci_in_mm <-
  mm_vs_pc_chr1_diffmethyl_table_hg38[which(overexpressed_gain1q_labels != "X"), ] %>%
  dplyr::select(Chromosome, CpG_beg, CpG_end, mean.diff)

hypometh_loci_in_mm <- dplyr::setdiff(
  x = as.data.frame(passing_diff_meth_probes) %>%
    dplyr::filter(mean.diff > 0) %>%
    dplyr::select(Chromosome, CpG_beg, CpG_end, mean.diff),
  y = hypometh_and_overexp_loci_in_mm
)

chrom_wide_diffmethyl_probes_plot <- plotKaryotype(
  genome = "hg38",
  plot.type = 2,
  chromosomes = "chr1",
)
kpAddBaseNumbers(chrom_wide_diffmethyl_probes_plot,
  add.units = T, cex = 0.8
)
kpDataBackground(chrom_wide_diffmethyl_probes_plot,
  data.panel = 1
)
kpDataBackground(chrom_wide_diffmethyl_probes_plot,
  data.panel = 2
)
kpAxis(chrom_wide_diffmethyl_probes_plot,
  ymin = 0,
  ymax = 0.75,
  data.panel = 1
)
kpRect(chrom_wide_diffmethyl_probes_plot,
  chr = "chr1",
  data.panel = 1,
  x0 = min(hypometh_and_overexp_loci_in_mm$CpG_end[hypometh_and_overexp_loci_in_mm$CpG_end >
    125000000]),
  x1 = 248956422,
  y0 = 0,

```

```

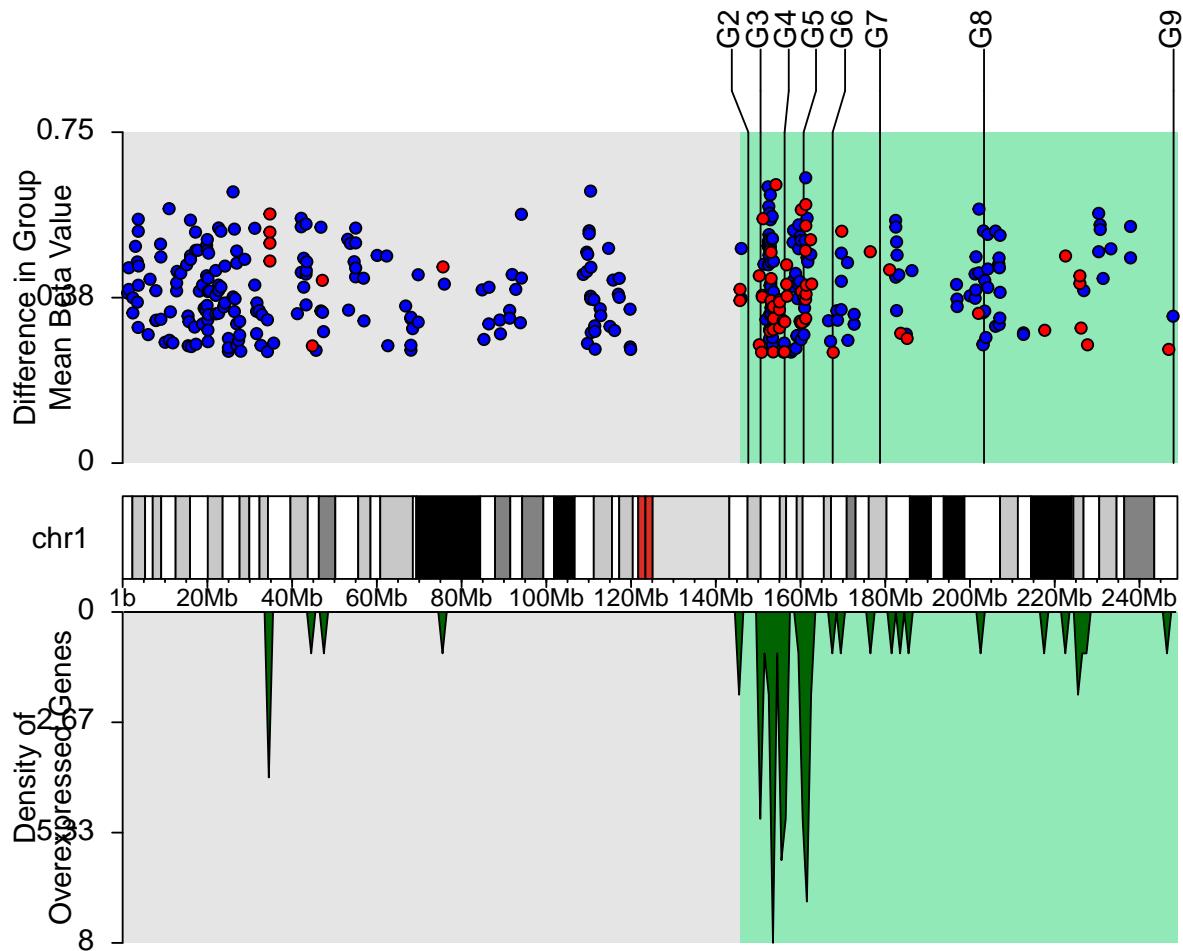
    y1 = 1,
    border = NA,
    col = alpha("seagreen2", alpha = 0.55)
)
kpRect(chrom_wide_diffmethyl_probes_plot,
    chr = "chr1",
    data.panel = 2,
    x0 = min(hypometh_and_overexp_loci_in_mm$CpG_end[hypometh_and_overexp_loci_in_mm$CpG_end >
        125000000]),
    x1 = 248956422,
    y0 = 0,
    y1 = 1,
    border = NA,
    col = alpha("seagreen2", alpha = 0.55)
)
kpPlotMarkers(chrom_wide_diffmethyl_probes_plot,
    chr = "chr1",
    data.panel = 1,
    x = recurrent_regions_midpoint_indicators %>%
        dplyr::filter(str_detect(string = id, pattern = "G[2-9]")) %>%
        dplyr::select(midpoint) %>%
        purrr::as_vector(),
    labels = recurrent_regions_midpoint_indicators %>%
        dplyr::filter(str_detect(string = id, pattern = "G[2-9]")) %>%
        dplyr::select(id) %>%
        purrr::as_vector(),
    ymax = 0.60
)
kpPoints(chrom_wide_diffmethyl_probes_plot,
    chr = "chr1",
    x = hypometh_loci_in_mm$CpG_end,
    y = hypometh_loci_in_mm$mean.diff,
    data.panel = 1,
    cex = 0.85,
    ymax = 0.75,
    pch = 21,
    col = "black",
    bg = "blue"
)
kpAddLabels(chrom_wide_diffmethyl_probes_plot,
    labels = "Difference in Group
Mean Beta Value",
    data.panel = 1,
    pos = 1,
    srt = 90,
    cex = 1.0,
    label.margin = 0.08
)
kpAddLabels(chrom_wide_diffmethyl_probes_plot,
    labels = "Density of
Overexpressed Genes",
    data.panel = 2,
    pos = 1,

```

```

    srt = 90,
    cex = 1.0,
    label.margin = 0.08
)
kpPoints(chrom_wide_diffmethyl_probes_plot,
  chr = "chr1",
  x = hypometh_and_overexp_loci_in_mm$CpG_end,
  y = hypometh_and_overexp_loci_in_mm$mean.diff,
  data.panel = 1,
  cex = 0.85,
  ymax = 0.75,
  pch = 21,
  col = "black",
  bg = "red"
)
density_plot_data <- kpPlotDensity(chrom_wide_diffmethyl_probes_plot,
  data.panel = 2,
  data = toGRanges(as.data.frame(hypometh_and_overexp_loci_in_mm), genome = "hg38"),
  window.size = 1000000,
  col = "darkgreen"
)
kpAxis(density_plot_data,
  ymin = 0,
  ymax = density_plot_data$latest.plot$computed.values$max.density,
  data.panel = 2,
  numticks = 4
)

```



```
# Export at 7x13 landscape
```

Generate boxplots of methylation probes across hypomethylated genes of interest in MM

```
# S100A4, S100A8, S100A9, SLAMF7, SLAMF1, PDZK1, and NTRK1
probes_for_interesting_genes <- c(
  "cg07426848", "cg24898863", "cg03165378", "cg11721194",
  "cg20535085", "cg10321723", "cg00626119"
)

# Create dataframe of methylation beta value across all samples for genes of interest
methyl_genes_of_interest <- chr1_methyl_data@methyl.sites[rownames(chr1_methyl_data@sites) %in%
  probes_for_interesting_genes, ]
```

```

rownames(methyl_genes_of_interest) <- probes_for_interesting_genes
methyl_genes_of_interest <- methyl_genes_of_interest %>%
  as.data.frame() %>%
  mutate(gene = c(
    "S100A4", "S100A8", "S100A9", "SLAMF7",
    "SLAMF1", "PDZK1", "NTRK1"
  ))
methyl_genes_of_interest <- reshape2::melt(methyl_genes_of_interest,
  value.name = "beta_value"
) %>%
  dplyr::rename("sampleID" = variable) %>%
  dplyr::mutate("clinical_subtype" = rep(core_groups, 7) %>%
    factor(levels = c("BC", "MGUS", "MM", "PC", "PCL", "HMCL")) %>%
    sort())

```

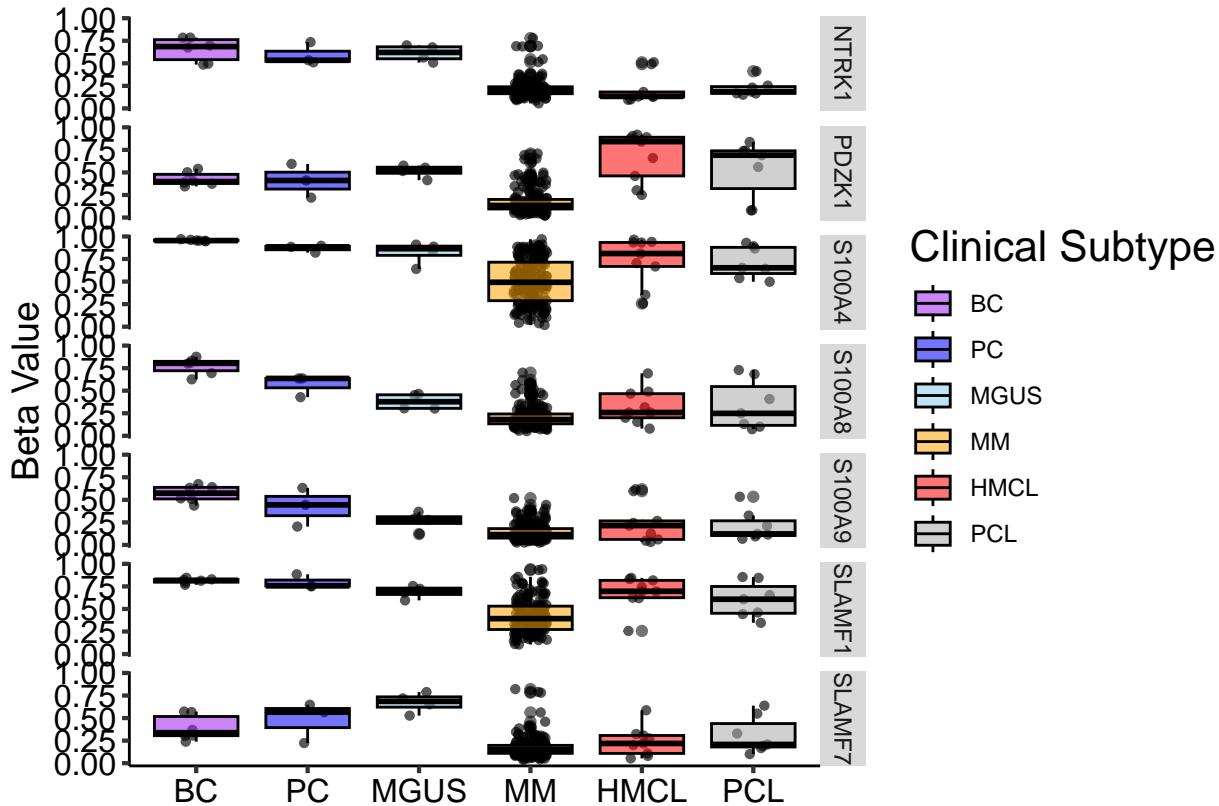
Using gene as id variables

```

methyl_genes_of_interest$clinical_subtype <- factor(methyl_genes_of_interest$clinical_subtype,
  levels = c("BC", "PC", "MGUS", "MM", "HMCL", "PCL")
)

ggplot(methyl_genes_of_interest) +
  geom_jitter(aes(x = clinical_subtype, y = beta_value),
    color = "black",
    size = 1.15,
    alpha = 0.65,
    width = 0.15
  ) +
  geom_boxplot(aes(x = clinical_subtype, y = beta_value, fill = clinical_subtype),
    color = "black",
    alpha = 0.55
  ) +
  scale_fill_manual(values = c("purple", "blue", "skyblue",
    "orange", "red", "darkgray")) +
  facet_grid(rows = vars(gene)) +
  labs(
    y = "Beta Value",
    x = "",
    fill = "Clinical Subtype"
  )

```



```
# Export at 8x13
```

Look at the methylation patterns within the t(4;14) subgroup, as nearly have have gain/amp(1q)

```
# Set samples to remove: all but t(4;14)s and PCs
t4_14s <- samples(
  geo_methylation_array_data)[which(chr1_methyl_data$pheno$`myeloma translocation` == "t(4;14)")]
samples_to_remove2 <- samples(
  geo_methylation_array_data)[!samples(geo_methylation_array_data) %in%
    c(t4_14s, "Plasma cell-1", "Plasma cell-2", "Plasma cell-3")]

# Subset the methylation data the t(4;14)s and PCs
t4_14s_methyl_chr1_data <- remove.samples(
  object = chr1_methyl_data,
  samplelist = samples_to_remove2
)

t4_14_analysis_group <- c(
  rep("t(4;14)", 15),
  rep("PC", 3)
)

t4_14s_methyl_chr1_data <- addPheno(
  object = t4_14s_methyl_chr1_data,
  t4_14_analysis_group,
```

```

    "t4_14_analysis_group"
)

rnb.options(
  min.group.size = 1,
  exploratory.columns = "t4_14_analysis_group",
  differential.comparison.columns = "t4_14_analysis_group"
)

rnb.run.exploratory(
  rnb.set = t4_14s_methyl_chr1_data,
  dir.reports = "data/differentialMethylation/t4_14s"
)

## Designed color mappings for probe type and CGI status

## Warning: Removed 150 rows containing non-finite outside the scale range
## ('stat_density()').
## Removed 150 rows containing non-finite outside the scale range
## ('stat_density()').

## Warning: Removed 74 rows containing non-finite outside the scale range
## ('stat_density()').
## Removed 74 rows containing non-finite outside the scale range
## ('stat_density()').

## Calculated MDS coordinates using manhattan distance

## Calculated MDS coordinates using euclidean distance

## Removed 13 loci ( sites ) because they contain missing values

## Calculated MDS coordinates using manhattan distance

## Calculated MDS coordinates using euclidean distance

## Removed 3 loci ( tiling ) because they contain missing values

## Calculated MDS coordinates using manhattan distance

## Calculated MDS coordinates using euclidean distance

## Removed 5 loci ( genes ) because they contain missing values

## Calculated MDS coordinates using manhattan distance

## Calculated MDS coordinates using euclidean distance

## Removed 5 loci ( promoters ) because they contain missing values

```

```

## Calculated MDS coordinates using manhattan distance

## Calculated MDS coordinates using euclidean distance

## Removed 3 loci ( cpgislands ) because they contain missing values

## Performed clustering on sites using correlation as a distance metric

## Performed clustering on sites using manhattan as a distance metric

## Performed clustering on sites using euclidean as a distance metric

## Performed clustering on tiling using correlation as a distance metric

## Performed clustering on tiling using manhattan as a distance metric

## Performed clustering on tiling using euclidean as a distance metric

## Performed clustering on genes using correlation as a distance metric

## Performed clustering on genes using manhattan as a distance metric

## Performed clustering on genes using euclidean as a distance metric

## Performed clustering on promoters using correlation as a distance metric

## Performed clustering on promoters using manhattan as a distance metric

## Performed clustering on promoters using euclidean as a distance metric

## Performed clustering on cpgislands using correlation as a distance metric

## Performed clustering on cpgislands using manhattan as a distance metric

## Performed clustering on cpgislands using euclidean as a distance metric

rnb.run.differential(
  rnb.set = t4_14s_methyl_chr1_data,
  dir.reports = "data/differentialMethylation/t4_14s"
)

## opening ff /private/var/folders/dp/sm5cqpy97zlc5cdvm58rn2mdpv6hnk/T/Rtmp010dvk/diffMethTables_111262

```

```
t4_14_diff_methyl_probes <- read_csv(
  file = "data/differentialMethylation/t4_14s/differential_methylation_data/diffMethTable_site_cmp1.csv",
  col_names = T,
  show_col_types = F
) %>%
  filter(mean.diff > 0.25 | mean.diff < -0.25) %>%
  filter(diffmeth.p.val < 0.05)
```

Main Analysis - Quantification of TADs in Context of Chromosome 1 Arms

Read in BED containing the TAD domains per arm

```
tad_domains_1p <- read_delim(
  file = paste0(base_data_path, "U266-MboI.domains.chr1p.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end"),
  show_col_types = F
) %>%
  mutate(length = end - start)

tad_domains_1q <- read_delim(
  file = paste0(base_data_path, "U266-MboI.domains.chr1q.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end"),
  show_col_types = F
) %>%
  mutate(length = end - start)
```

Calculate quantitative metrics for TAD structures per arm

```
median(tad_domains_1p$length)
```

```
## [1] 520000
```

```
range(tad_domains_1p$length)
```

```
## [1] 6459 5160000
```

```
median(tad_domains_1q$length)
```

```
## [1] 560000
```

```
range(tad_domains_1q$length)
```

```
## [1] 220000 18340000
```

Evaluate the histone mark signals and possible enrichment

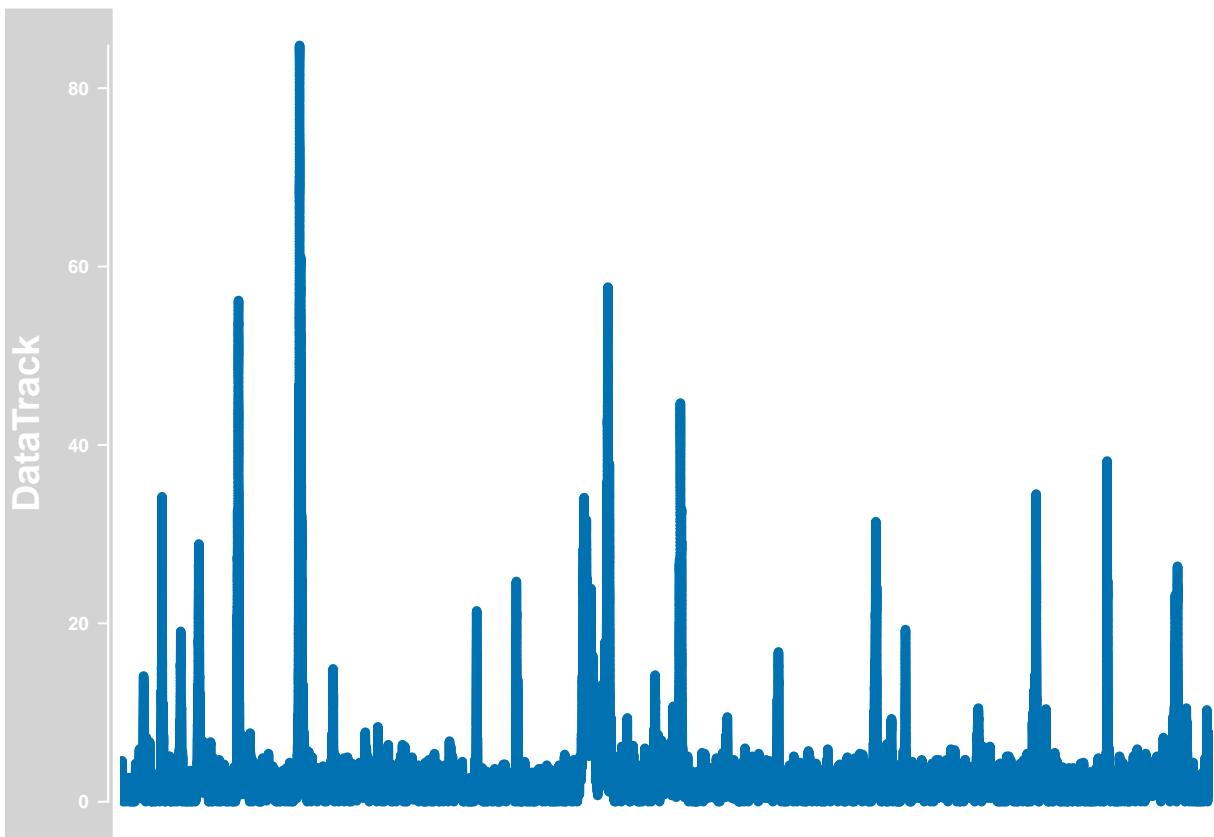
```

h3k27ac <- import.bw(
  paste0(base_data_path,
    "hicbenchOutput/blueprintChipSeq/U-266_c01.ERX297418.H3K27ac.bwa.GRCh38.20150528.bw"),
  as = "GRanges"
)

test_track <- DataTrack(h3k27ac, chromosome = "chr1")

# Signal across entire TAD
plotTracks(test_track, from = 150240000, to = 151000000, chromosome = "chr1")

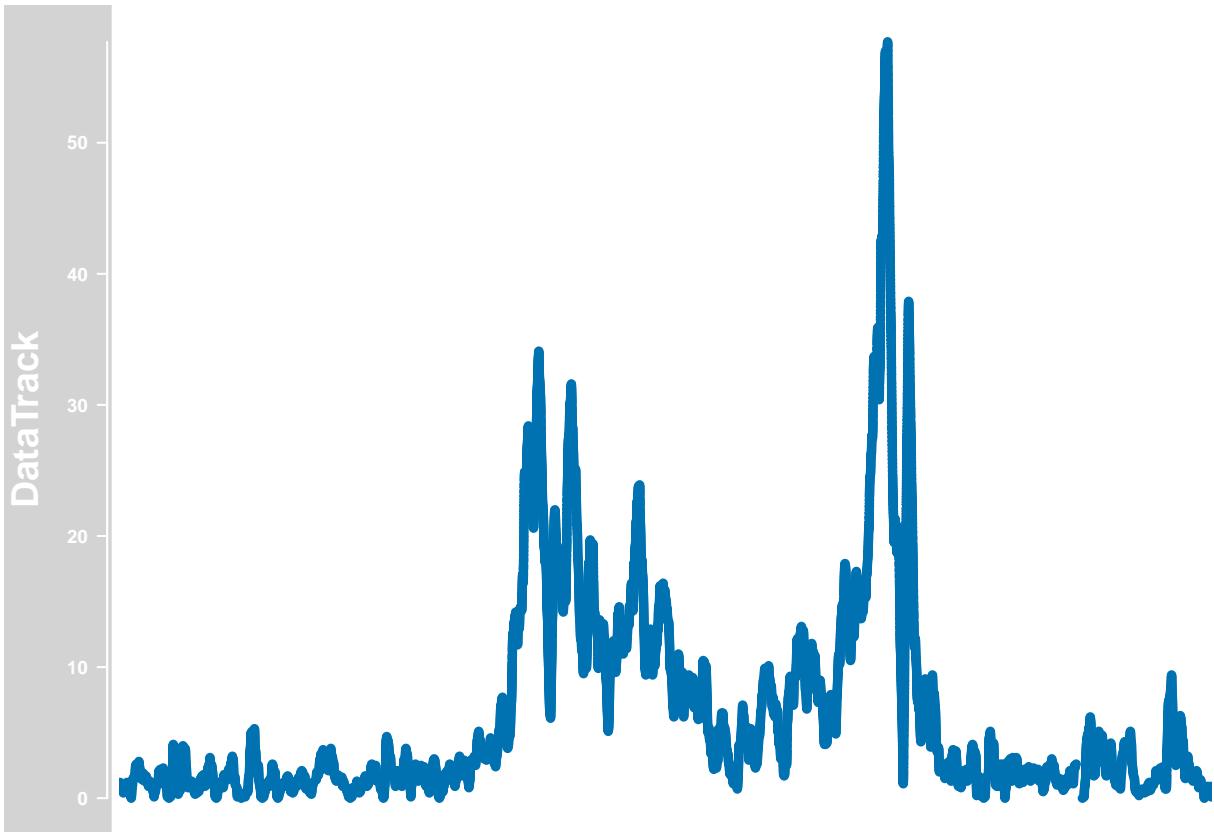
```



```

# Signal across hotspot located within TAD
plotTracks(test_track, from = 150542525, to = 150594423, chromosome = "chr1")

```



Main Analysis - Compartment Quantification in Regions of Gain and Deletion

Read in BED files for chromosome 1 compartment scores, A/B designation, and GISTIC2.0/PCF regions of interest

```
nbc_compartment_scores <- read_delim(
  file = paste0(
    base_data_path,
    "hicbenchOutput/compartments/NBC-untreated-DpnII-rep1/compartments.scores.bedGraph"),
  delim = "\t",
  skip = 1,
  col_names = c("chrom", "start", "end", "score"),
  show_col_types = F
) %>%
  dplyr::filter(chrom == "chr1")

mbc_compartment_scores <- read_delim(
  file = paste0(
    base_data_path,
    "hicbenchOutput/compartments/MBC-untreated-DpnII-rep1/compartments.scores.bedGraph"),
  delim = "\t",
  skip = 1,
  col_names = c("chrom", "start", "end", "score"),
  show_col_types = F
```

```

) %>%
  dplyr::filter(chrom == "chr1")

gcbc_compartment_scores <- read_delim(
  file = paste0(
    base_data_path,
    "hicbenchOutput/compartments/GCBC-untreated-DpnII-rep1/compartments.scores.bedGraph"),
  delim = "\t",
  skip = 1,
  col_names = c("chrom", "start", "end", "score"),
  show_col_types = F
) %>%
  dplyr::filter(chrom == "chr1")

pc_compartment_scores <- read_delim(
  file = paste0(
    base_data_path,
    "hicbenchOutput/compartments/PC-untreated-DpnII-rep1/compartments.scores.bedGraph"),
  delim = "\t",
  skip = 1,
  col_names = c("chrom", "start", "end", "score"),
  show_col_types = F
) %>%
  dplyr::filter(chrom == "chr1")

u266_compartment_scores <- read_delim(
  file = paste0(
    base_data_path,
    "hicbenchOutput/compartments/U266-untreated-MboI-rep1/compartments.scores.bedGraph"),
  delim = "\t",
  skip = 1,
  col_names = c("chrom", "start", "end", "score"),
  show_col_types = F
) %>%
  dplyr::filter(chrom == "chr1")

rpmi8226_compartment_scores <- read_delim(
  file = paste0(
    base_data_path,
    "hicbenchOutput/compartments/RPMI8226-untreated-MboI-rep1/compartments.scores.bedGraph"),
  delim = "\t",
  skip = 1,
  col_names = c("chrom", "start", "end", "score"),
  show_col_types = F
) %>%
  dplyr::filter(chrom == "chr1")

kms11_compartment_scores <- read_delim(
  file = paste0(
    base_data_path,
    "hicbenchOutput/compartments/KMS11-untreated-MboI-rep1/compartments.scores.bedGraph"),
  delim = "\t",
  skip = 1,

```

```
  col_names = c("chrom", "start", "end", "score"),
  show_col_types = F
) %>%
  dplyr::filter(chrom == "chr1")

regions_of_interest <- read_delim(
  file = paste0(base_data_path, "gisticRegionsWithSvHotspots.hg38.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "region"),
  show_col_types = F
) %>%
  dplyr::mutate(length = end - start)
```

First, perform pairwise Pearson correlation on compartment scores across entire chromosome for all samples

```

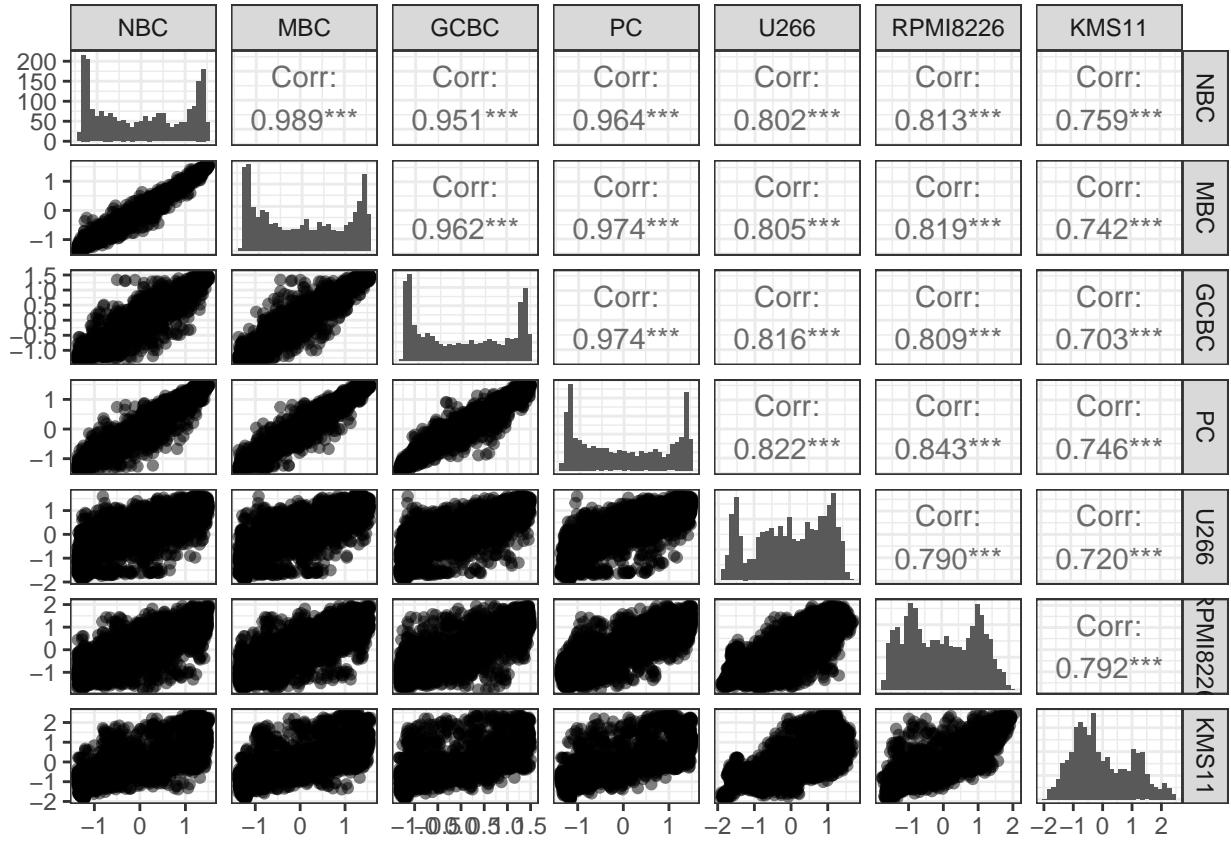
compartment_sample_ids <- c("NBC", "MBC", "GCBC", "PC", "U266", "RPMI8226", "KMS11")

compartment_score_similarity <- left_join(x = nbc_compartment_scores,
                                             y = mbc_compartment_scores,
                                             by = "start") %>%
  dplyr::rename(NBC = "score.x") %>%
  dplyr::rename(MBC = "score.y") %>%
  left_join(y = gcbc_compartment_scores, by = "start") %>%
  dplyr::rename(GCBC = "score") %>%
  left_join(y = pc_compartment_scores, by = "start") %>%
  dplyr::rename(PC = "score") %>%
  left_join(y = u266_compartment_scores, by = "start") %>%
  dplyr::rename(U266 = "score") %>%
  left_join(y = rpmi8226_compartment_scores, by = "start") %>%
  dplyr::rename(RPMI8226 = "score") %>%
  left_join(y = kms11_compartment_scores, by = "start") %>%
  dplyr::rename(KMS11 = "score") %>%
  dplyr::select(chrom, start, end, NBC, MBC, GCBC, PC, U266, RPMI8226, KMS11) %>%
  na.omit()

ggpairs(
  data = compartment_score_similarity,
  columns = 4:10,
  diag = list(continuous = "barDiag"),
  lower = list(continuous = "autopoint")
) +
  theme_bw()

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat bin()` using `bins = 30`. Pick better value with `binwidth`.

```



```
# export 8x10 landscape
```

Restrict the compartment scores to only the segments within the boundaries of the GISTIC2.0 regions of gain

```
regions_of_gain <- regions_of_interest %>% filter(str_detect(string = region,
                                                               pattern = "G"))

nbc_gain_compartments <- read_delim(
  file = paste0(base_data_path,
                "hicbenchOutput/compartments/NBC_roi_AB_compartments.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "compartment", "region"),
  show_col_types = F
) %>%
  dplyr::mutate(length = end - start) %>%
  dplyr::mutate(id = "NBC") %>%
  dplyr::filter(str_detect(string = region, pattern = "G"))

mbc_gain_compartments <- read_delim(
  file = paste0(base_data_path,
                "hicbenchOutput/compartments/MBC_roi_AB_compartments.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "compartment", "region"),
  show_col_types = F
```

```

) %>%
  dplyr::mutate(length = end - start) %>%
  dplyr::mutate(id = "MBC") %>%
  dplyr::filter(str_detect(string = region, pattern = "G"))

gcbc_gain_compartments <- read_delim(
  file = paste0(base_data_path,
    "hicbenchOutput/compartments/GCBC_roi_AB_compartments.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "compartment", "region"),
  show_col_types = F
) %>%
  dplyr::mutate(length = end - start) %>%
  dplyr::mutate(id = "GCBC") %>%
  dplyr::filter(str_detect(string = region, pattern = "G"))

pc_gain_compartments <- read_delim(
  file = paste0(base_data_path,
    "hicbenchOutput/compartments/PC_roi_AB_compartments.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "compartment", "region"),
  show_col_types = F
) %>%
  dplyr::mutate(length = end - start) %>%
  dplyr::mutate(id = "PC") %>%
  dplyr::filter(str_detect(string = region, pattern = "G"))

u266_gain_compartments <- read_delim(
  file = paste0(base_data_path,
    "hicbenchOutput/compartments/U266_roi_AB_compartments.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "compartment", "region"),
  show_col_types = F
) %>%
  dplyr::mutate(length = end - start) %>%
  dplyr::mutate(id = "U266") %>%
  dplyr::filter(str_detect(string = region, pattern = "G"))

rpmi8226_gain_compartments <- read_delim(
  file = paste0(base_data_path,
    "hicbenchOutput/compartments/RPMI8226_roi_AB_compartments.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "compartment", "region"),
  show_col_types = F
) %>%
  dplyr::mutate(length = end - start) %>%
  dplyr::mutate(id = "RPMI8226") %>%
  dplyr::filter(str_detect(string = region, pattern = "G"))

kms11_gain_compartments <- read_delim(
  file = paste0(base_data_path,
    "hicbenchOutput/compartments/KMS11_roi_AB_compartments.bed"),
  delim = "\t",

```

```

col_names = c("chrom", "start", "end", "compartment", "region"),
show_col_types = F
) %>%
  dplyr::mutate(length = end - start) %>%
  dplyr::mutate(id = "KMS11") %>%
  dplyr::filter(str_detect(string = region, pattern = "G"))

all_samples_gain_compartments <- rbind(
  nbc_gain_compartments, mbc_gain_compartments, gcbc_gain_compartments,
  pc_gain_compartments, u266_gain_compartments, rpmi8226_gain_compartments,
  kms11_gain_compartments
)

gain_compartment_quant <- tibble()
for (i in 1:length(compartment_sample_ids)) {
  # Quantify all A compartment percentages
  sample_gain_a_compartments <- all_samples_gain_compartments %>%
    dplyr::filter(id == compartment_sample_ids[i]) %>%
    dplyr::group_by(region) %>%
    dplyr::filter(compartment == "A") %>%
    dplyr::mutate(total_length = sum(length)) %>%
    dplyr::select(region, total_length) %>%
    unique()

  sample_gain_a_compartment_percent <- sample_gain_a_compartments$total_length /
    regions_of_gain$length * 100

  gain_compartment_quant <- rbind(
    gain_compartment_quant,
    tibble(
      region = sample_gain_a_compartments$region,
      percent = round(sample_gain_a_compartment_percent, digits = 0),
      id = rep(compartment_sample_ids[i], 9),
      compartment = rep("A", 9)
    )
  )

  # Quantify all B compartment percentages
  sample_gain_b_compartments <- all_samples_gain_compartments %>%
    dplyr::filter(id == compartment_sample_ids[i]) %>%
    dplyr::group_by(region) %>%
    dplyr::filter(compartment == "B") %>%
    dplyr::mutate(total_length = sum(length)) %>%
    dplyr::select(region, total_length) %>%
    unique()

  # Not all regions have B compartments
  if (nrow(sample_gain_b_compartments) != 0) {
    sample_gain_b_compartment_percent <- sample_gain_b_compartments$total_length /
      (regions_of_gain %>%
        dplyr::filter(region %in% sample_gain_b_compartments$region) %>%
        dplyr::select(length)) * 100
  }
}

```

```

gain_compartment_quant <- rbind(
  gain_compartment_quant,
  tibble(
    region = sample_gain_b_compartments$region,
    percent = round(sample_gain_b_compartment_percent$length, digits = 0),
    id = rep(compartment_sample_ids[i], nrow(sample_gain_b_compartment_percent)),
    compartment = rep("B", nrow(sample_gain_b_compartment_percent))
  )
)
}

# Prep for data for better visualization
for (i in 1:length(compartment_sample_ids)) {
  sample_specific_gain_compartment_quant <- gain_compartment_quant %>%
    dplyr::filter(id == compartment_sample_ids[i])

  for (j in 1:length(regions_of_gain$region)) {
    sample_and_region_specific_gain_compartment_quant <-
      sample_specific_gain_compartment_quant %>%
        dplyr::filter(region == regions_of_gain$region[j])

    # Add row if there are any unassigned segments to the region
    # Check if total compartment percentage per region is equal to 100
    gain_compartment_quant <- rbind(
      gain_compartment_quant,
      tibble(
        region = regions_of_gain$region[j],
        percent = 100 - round(sum(sample_and_region_specific_gain_compartment_quant$percent),
          digits = 0
        ),
        id = compartment_sample_ids[i],
        compartment = "unassigned"
      )
    )

    # Add row for B compartment if the value is zero
    if (!"B" %in% sample_and_region_specific_gain_compartment_quant$compartment) {
      gain_compartment_quant <- rbind(
        gain_compartment_quant,
        tibble(
          region = regions_of_gain$region[j],
          percent = 0,
          id = compartment_sample_ids[i],
          compartment = "B"
        )
      )
    }
  }
}

gain_compartment_quant$region <- factor(gain_compartment_quant$region)
gain_compartment_quant$id <- factor(gain_compartment_quant$id,

```

```

    levels = compartment_sample_ids
)
gain_compartment_quant$compartment <- factor(gain_compartment_quant$compartment)

gain_compartment_grid_plot <- ggplot(gain_compartment_quant) +
  geom_bar(aes(x = "", y = percent, fill = compartment), stat = "identity",
            width = 1, color = "black", position = "dodge") +
  scale_fill_manual(values = c("chartreuse2", "orchid2", "gray")) +
  facet_grid(id ~ region) +
  theme(
    axis.line.x = element_blank(),
    axis.line.y = element_blank(),
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.text.y = element_text(size = 8),
    legend.text = element_text(size = 12),
    panel.background = element_rect(
      color = "black",
      size = 1
    )
  )
# export 8x11 landscape

```

Restrict the compartment scores to only the segments within the boundaries of the GISTIC2.0 regions of deletion

```

regions_of_deletion <- regions_of_interest %>%
  filter(str_detect(string = region, pattern = "D"))

nbc_deletion_compartments <- read_delim(
  file = paste0(base_data_path,
                "hicbenchOutput/compartments/NBC_roi_AB_compartments.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "compartment", "region"),
  show_col_types = F
) %>%
  dplyr::mutate(length = end - start) %>%
  dplyr::mutate(id = "NBC") %>%
  dplyr::filter(str_detect(string = region, pattern = "D"))

mbc_deletion_compartments <- read_delim(
  file = paste0(base_data_path,
                "hicbenchOutput/compartments/MBC_roi_AB_compartments.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "compartment", "region"),
  show_col_types = F
) %>%
  dplyr::mutate(length = end - start) %>%
  dplyr::mutate(id = "MBC") %>%
  dplyr::filter(str_detect(string = region, pattern = "D"))

```

```

gcbc_deletion_compartments <- read_delim(
  file = paste0(base_data_path,
                "hicbenchOutput/compartments/GCBC_roi_AB_compartments.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "compartment", "region"),
  show_col_types = F
) %>%
  dplyr::mutate(length = end - start) %>%
  dplyr::mutate(id = "GCBC") %>%
  dplyr::filter(str_detect(string = region, pattern = "D"))

pc_deletion_compartments <- read_delim(
  file = paste0(base_data_path,
                "hicbenchOutput/compartments/PC_roi_AB_compartments.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "compartment", "region"),
  show_col_types = F
) %>%
  dplyr::mutate(length = end - start) %>%
  dplyr::mutate(id = "PC") %>%
  dplyr::filter(str_detect(string = region, pattern = "D"))

u266_deletion_compartments <- read_delim(
  file = paste0(base_data_path,
                "hicbenchOutput/compartments/U266_roi_AB_compartments.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "compartment", "region"),
  show_col_types = F
) %>%
  dplyr::mutate(length = end - start) %>%
  dplyr::mutate(id = "U266") %>%
  dplyr::filter(str_detect(string = region, pattern = "D"))

rpmi8226_deletion_compartments <- read_delim(
  file = paste0(base_data_path,
                "hicbenchOutput/compartments/RPMI8226_roi_AB_compartments.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "compartment", "region"),
  show_col_types = F
) %>%
  dplyr::mutate(length = end - start) %>%
  dplyr::mutate(id = "RPMI8226") %>%
  dplyr::filter(str_detect(string = region, pattern = "D"))

kms11_deletion_compartments <- read_delim(
  file = paste0(base_data_path,
                "hicbenchOutput/compartments/KMS11_roi_AB_compartments.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "compartment", "region"),
  show_col_types = F
) %>%
  dplyr::mutate(length = end - start) %>%
  dplyr::mutate(id = "KMS11") %>%

```

```

dplyr::filter(str_detect(string = region, pattern = "D"))

all_samples_deletion_compartments <- rbind(
  nbc_deletion_compartments, mbc_deletion_compartments,
  gcbc_deletion_compartments, pc_deletion_compartments,
  u266_deletion_compartments, rpmi8226_deletion_compartments,
  kms11_deletion_compartments
)

deletion_compartment_quant <- tibble()
for (i in 1:length(compartment_sample_ids)) {
  # Quantify all A compartment percentages
  sample_deletion_a_compartments <- all_samples_deletion_compartments %>%
    dplyr::filter(id == compartment_sample_ids[i]) %>%
    dplyr::group_by(region) %>%
    dplyr::filter(compartment == "A") %>%
    dplyr::mutate(total_length = sum(length)) %>%
    dplyr::select(region, total_length) %>%
    unique()

  if (nrow(sample_deletion_a_compartments) != 0) {
    sample_deletion_a_compartment_percent <- sample_deletion_a_compartments$total_length /
      (regions_of_deletion %>%
        dplyr::filter(region %in% sample_deletion_a_compartments$region) %>%
        dplyr::select(length)) * 100

    deletion_compartment_quant <- rbind(
      deletion_compartment_quant,
      tibble(
        region = sample_deletion_a_compartments$region,
        percent = sample_deletion_a_compartment_percent$length,
        id = rep(compartment_sample_ids[i], nrow(sample_deletion_a_compartment_percent)),
        compartment = rep("A", nrow(sample_deletion_a_compartment_percent))
      )
    )
  }
}

# Quantify all B compartment percentages
sample_deletion_b_compartments <- all_samples_deletion_compartments %>%
  dplyr::filter(id == compartment_sample_ids[i]) %>%
  dplyr::group_by(region) %>%
  dplyr::filter(compartment == "B") %>%
  dplyr::mutate(total_length = sum(length)) %>%
  dplyr::select(region, total_length) %>%
  unique()

# Not all regions have B compartments
if (nrow(sample_deletion_b_compartments) != 0) {
  sample_deletion_b_compartment_percent <- sample_deletion_b_compartments$total_length /
    (regions_of_deletion %>%
      dplyr::filter(region %in% sample_deletion_b_compartments$region) %>%
      dplyr::select(length)) * 100
}

```

```

deletion_compartment_quant <- rbind(
  deletion_compartment_quant,
  tibble(
    region = sample_deletion_b_compartments$region,
    percent = sample_deletion_b_compartment_percent$length,
    id = rep(compartment_sample_ids[i], nrow(sample_deletion_b_compartment_percent)),
    compartment = rep("B", nrow(sample_deletion_b_compartment_percent))
  )
)
}

# Prep for data for better visualization
for (i in 1:length(compartment_sample_ids)) {
  sample_specific_deletion_compartment_quant <- deletion_compartment_quant %>%
    dplyr::filter(id == compartment_sample_ids[i])

  for (j in 1:length(regions_of_deletion$region)) {
    sample_and_region_specific_deletion_compartment_quant <-
      sample_specific_deletion_compartment_quant %>%
        dplyr::filter(region == regions_of_deletion$region[j])

    # Add row if there are any unassigned segments to the region
    # Check if total compartment percentage per region is equal to 100
    deletion_compartment_quant <- rbind(
      deletion_compartment_quant,
      tibble(
        region = regions_of_deletion$region[j],
        percent = 100 - round(sum(sample_and_region_specific_deletion_compartment_quant$percent),
          digits = 0),
        id = compartment_sample_ids[i],
        compartment = "unassigned"
      )
    )

    # Add row for A compartment if the value is zero
    if (!"A" %in% sample_and_region_specific_deletion_compartment_quant$compartment) {
      deletion_compartment_quant <- rbind(
        deletion_compartment_quant,
        tibble(
          region = regions_of_deletion$region[j],
          percent = 0,
          id = compartment_sample_ids[i],
          compartment = "A"
        )
      )
    }
  }

  # Add row for B compartment if the value is zero
  if (!"B" %in% sample_and_region_specific_deletion_compartment_quant$compartment) {
    deletion_compartment_quant <- rbind(
      deletion_compartment_quant,

```

```

        tibble(
            region = regions_of_deletion$region[j],
            percent = 0,
            id = compartment_sample_ids[i],
            compartment = "B"
        )
    )
}
}

deletion_compartment_quant$region <- factor(deletion_compartment_quant$region)
deletion_compartment_quant$id <- factor(deletion_compartment_quant$id,
    levels = compartment_sample_ids
)
deletion_compartment_quant$compartment <- factor(deletion_compartment_quant$compartment)

deletion_compartment_grid_plot <- ggplot(deletion_compartment_quant) +
    geom_bar(aes(x = "", y = percent, fill = compartment), stat = "identity",
        width = 1, color = "black", position = "dodge") +
    scale_fill_manual(values = c("chartreuse2", "orchid2", "gray")) +
    facet_grid(id ~ region) +
    theme(
        axis.line.x = element_blank(),
        axis.line.y = element_blank(),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_text(size = 8),
        axis.ticks.x = element_blank(),
        legend.text = element_text(size = 12),
        panel.background = element_rect(
            color = "black",
            size = 1
        )
    )
# export 8x11 landscape

```

Restrict the compartment scores to only the segments within the boundaries of the PCF regions of TI and CT

```

regions_of_sv <- regions_of_interest %>% filter(str_detect(string = region,
    pattern = "TI|CT"))

nbc_sv_compartments <- read_delim(
    file = paste0(base_data_path,
        "hicbenchOutput/compartments/NBC_roi_AB_compartments.bed"),
    delim = "\t",
    col_names = c("chrom", "start", "end", "compartment", "region"),
    show_col_types = F
) %>%
    dplyr::mutate(length = end - start) %>%
    dplyr::mutate(id = "NBC") %>%

```

```

dplyr::filter(str_detect(string = region, pattern = "TI|CT"))

mbc_sv_compartments <- read_delim(
  file = paste0(base_data_path,
                "hicbenchOutput/compartments/MBC_roi_AB_compartments.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "compartment", "region"),
  show_col_types = F
) %>%
  dplyr::mutate(length = end - start) %>%
  dplyr::mutate(id = "MBC") %>%
  dplyr::filter(str_detect(string = region, pattern = "TI|CT"))

gcbc_sv_compartments <- read_delim(
  file = paste0(base_data_path,
                "hicbenchOutput/compartments/GCBC_roi_AB_compartments.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "compartment", "region"),
  show_col_types = F
) %>%
  dplyr::mutate(length = end - start) %>%
  dplyr::mutate(id = "GCBC") %>%
  dplyr::filter(str_detect(string = region, pattern = "TI|CT"))

pc_sv_compartments <- read_delim(
  file = paste0(base_data_path,
                "hicbenchOutput/compartments/PC_roi_AB_compartments.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "compartment", "region"),
  show_col_types = F
) %>%
  dplyr::mutate(length = end - start) %>%
  dplyr::mutate(id = "PC") %>%
  dplyr::filter(str_detect(string = region, pattern = "TI|CT"))

u266_sv_compartments <- read_delim(
  file = paste0(base_data_path,
                "hicbenchOutput/compartments/U266_roi_AB_compartments.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "compartment", "region"),
  show_col_types = F
) %>%
  dplyr::mutate(length = end - start) %>%
  dplyr::mutate(id = "U266") %>%
  dplyr::filter(str_detect(string = region, pattern = "TI|CT"))

rpml8226_sv_compartments <- read_delim(
  file = paste0(base_data_path,
                "hicbenchOutput/compartments/RPMI8226_roi_AB_compartments.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "compartment", "region"),
  show_col_types = F
) %>%

```

```

dplyr::mutate(length = end - start) %>%
dplyr::mutate(id = "RPMI8226") %>%
dplyr::filter(str_detect(string = region, pattern = "TI|CT"))

kms11_sv_compartments <- read_delim(
  file = paste0(base_data_path,
    "hicbenchOutput/compartments/KMS11_roi_AB_compartments.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "compartment", "region"),
  show_col_types = F
) %>%
  dplyr::mutate(length = end - start) %>%
  dplyr::mutate(id = "KMS11") %>%
  dplyr::filter(str_detect(string = region, pattern = "TI|CT"))

all_samples_sv_compartments <- rbind(
  nbc_sv_compartments, mbc_sv_compartments,
  gcbc_sv_compartments, pc_sv_compartments,
  u266_sv_compartments, rmpmi8226_sv_compartments,
  kms11_sv_compartments
)

sv_compartment_quant <- tibble()
for (i in 1:length(compartment_sample_ids)) {
  # Quantify all A compartment percentages
  sample_sv_a_compartments <- all_samples_sv_compartments %>%
    dplyr::filter(id == compartment_sample_ids[i]) %>%
    dplyr::group_by(region) %>%
    dplyr::filter(compartment == "A") %>%
    dplyr::mutate(total_length = sum(length)) %>%
    dplyr::select(region, total_length) %>%
    unique()

  if (nrow(sample_sv_a_compartments) != 0) {
    sample_sv_a_compartment_percent <- sample_sv_a_compartments$total_length /
      (regions_of_sv %>%
        dplyr::filter(region %in% sample_sv_a_compartments$region) %>%
        dplyr::select(length) * 100

    sv_compartment_quant <- rbind(
      sv_compartment_quant,
      tibble(
        region = sample_sv_a_compartments$region,
        percent = sample_sv_a_compartment_percent$length,
        id = rep(compartment_sample_ids[i], nrow(sample_sv_a_compartment_percent)),
        compartment = rep("A", nrow(sample_sv_a_compartment_percent))
      )
    )
  }
}

# Quantify all B compartment percentages
sample_sv_b_compartments <- all_samples_sv_compartments %>%
  dplyr::filter(id == compartment_sample_ids[i]) %>%

```

```

dplyr::group_by(region) %>%
dplyr::filter(compartment == "B") %>%
dplyr::mutate(total_length = sum(length)) %>%
dplyr::select(region, total_length) %>%
unique()

# Not all regions have B compartments
if (nrow(sample_sv_b_compartments) != 0) {
  sample_sv_b_compartment_percent <- sample_sv_b_compartments$total_length /
  (regions_of_sv %>%
    dplyr::filter(region %in% sample_sv_b_compartments$region) %>%
    dplyr::select(length)) * 100

  sv_compartment_quant <- rbind(
    sv_compartment_quant,
    tibble(
      region = sample_sv_b_compartments$region,
      percent = sample_sv_b_compartment_percent$length,
      id = rep(compartment_sample_ids[i], nrow(sample_sv_b_compartment_percent)),
      compartment = rep("B", nrow(sample_sv_b_compartment_percent))
    )
  )
}

# Prep for data for better visualization
for (i in 1:length(compartment_sample_ids)) {
  sample_specific_sv_compartment_quant <- sv_compartment_quant %>%
    dplyr::filter(id == compartment_sample_ids[i])

  for (j in 1:length(regions_of_sv$region)) {
    sample_and_region_specific_sv_compartment_quant <-
      sample_specific_sv_compartment_quant %>%
        dplyr::filter(region == regions_of_sv$region[j])

    # Add row if there are any unassigned segments to the region
    # Check if total compartment percentage per region is equal to 100
    sv_compartment_quant <- rbind(
      sv_compartment_quant,
      tibble(
        region = regions_of_sv$region[j],
        percent = 100 - round(sum(sample_and_region_specific_sv_compartment_quant$percent),
          digits = 0
        ),
        id = compartment_sample_ids[i],
        compartment = "unassigned"
      )
    )

    # Add row for A compartment if the value is zero
    if (!"A" %in% sample_and_region_specific_sv_compartment_quant$compartment) {
      sv_compartment_quant <- rbind(
        sv_compartment_quant,

```

```

        tibble(
            region = regions_of_sv$region[j],
            percent = 0,
            id = compartment_sample_ids[i],
            compartment = "A"
        )
    )
}

# Add row for B compartment if the value is zero
if (!"B" %in% sample_and_region_specific_sv_compartment_quant$compartment) {
    sv_compartment_quant <- rbind(
        sv_compartment_quant,
        tibble(
            region = regions_of_sv$region[j],
            percent = 0,
            id = compartment_sample_ids[i],
            compartment = "B"
        )
    )
}

sv_compartment_quant$region <- factor(sv_compartment_quant$region)
sv_compartment_quant$id <- factor(sv_compartment_quant$id,
    levels = compartment_sample_ids
)
sv_compartment_quant$compartment <- factor(sv_compartment_quant$compartment)

sv_compartment_grid_plot <- ggplot(sv_compartment_quant) +
    geom_bar(aes(x = "", y = percent, fill = compartment), stat = "identity",
        width = 1, color = "black", position = "dodge") +
    scale_fill_manual(values = c("chartreuse2", "orchid2", "gray")) +
    facet_grid(id ~ region) +
    theme(
        axis.line.x = element_blank(),
        axis.line.y = element_blank(),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_text(size = 8),
        axis.ticks.x = element_blank(),
        legend.text = element_text(size = 12),
        panel.background = element_rect(
            color = "black",
            size = 1
        )
    )
}

# export 8x11 landscape

```

Plot compartment quantification grid plots together in one panel

```
ggdraw() +
  draw_plot(gain_compartment_grid_plot + theme(legend.position = "top"),
            x = 0, y = 0.66, width = 1, height = 0.34) +
  draw_plot(deletion_compartment_grid_plot + theme(legend.position = "none"),
            x = 0, y = 0.33, width = 0.791, height = 0.33) +
  draw_plot(sv_compartment_grid_plot + theme(legend.position = "none"),
            x = 0, y = 0.0, width = 0.58, height = 0.33)
```

compartment A B unassigned



```
# export 9x17 portrait
```

Visualize epigenetic tracks at specific loci

```

abl2_region <- "chr1:179,107,718-179,229,397"

locusEpiPlot <- function(locus, window_buffer = 500, scale_max = F) {
  # Set base of plot, focused to a user-defined locus plus window buffer
  roi <- toGRanges(locus, genome = "hg38")
  roi$ranges$start <- as.integer(roi$ranges$start - window_buffer)
  roi$ranges$width <- as.integer(roi$ranges$width + window_buffer)
  roi_window <- toGRanges(
    genome = "hg38",
    A = tibble(
      chrom = "chr1",
      start = as.character(roi$ranges$start - window_buffer),
      end = as.character(roi$ranges$start + roi$ranges$width + window_buffer - 1)
    )
  )

  # Set path to cell line expression RPKM files
  rna_tracks <- c(
    KMS11 = paste0(base_data_path,
                  "hicbenchOutput/tracks/locusSpecific/kms11_median.tpm.bw"),
    RPMI8226 = paste0(base_data_path,
                      "hicbenchOutput/tracks/locusSpecific/rpmi8226_median.tpm.bw"),
    U266 = paste0(base_data_path,
                  "hicbenchOutput/tracks/locusSpecific/u266_median.tpm.bw"),
    PC = paste0(base_data_path,
                "hicbenchOutput/tracks/locusSpecific/pc_median.tpm.bw")
  )
}

# Start the plot
kp <- plotKaryotype(
  zoom = roi_window,
  genome = "hg38"
)

# Load gene information, then instruct to use merged transcript representation
genes_data <- makeGenesDataFromTxDb(
  txdb = TxDb.Hsapiens.UCSC.hg38.knownGene,
  karyoplot = kp,
  plot.transcripts = TRUE,
  plot.transcripts.structure = TRUE
)
genes_data <- addGeneNames(genes_data)
genes_data <- mergeTranscripts(genes_data)

# Add genomic location scale to plot
kpAddBaseNumbers(kp,
  tick.dist = 20000,
  minor.tick.dist = 5000,
  add.units = TRUE,
  cex = 0.8,
  digits = 6
)

```

```

# Add a bigWig track for each sample expression in RPKM format
kpPlotGenes(kp,
  data = genes_data,
  r0 = 0,
  r1 = 0.20,
  gene.name.cex = 1
)

for (i in seq_len(length(rna_tracks))) {
  bigwig.file <- rna_tracks[i]
  at <- autotrack(i, length(rna_tracks),
    r0 = 0.35,
    r1 = 1,
    margin = 0.1
  )
  kp <- kpPlotBigWig(kp,
    data = bigwig.file,
    ymax = "visible.region",
    r0 = at$r0,
    r1 = at$r1
  )

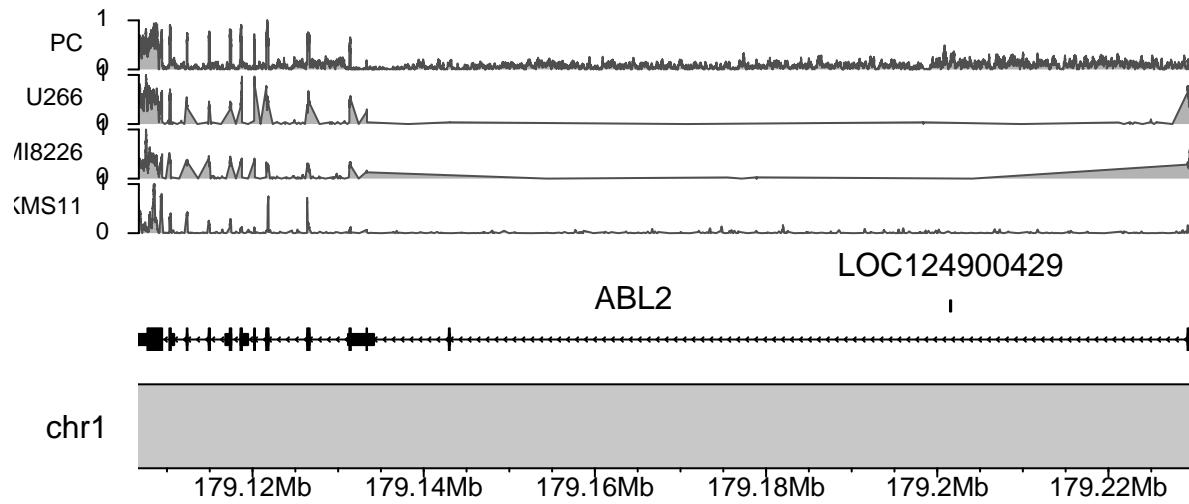
  if (scale_max) {
    computed.ymax <- scale_max
  } else {
    computed.ymax <- ceiling(kp$latest.plot$computed.values$ymax)
  }

  kpAxis(kp,
    ymin = 0,
    ymax = computed.ymax,
    numticks = 2,
    cex = 0.75,
    r0 = at$r0,
    r1 = at$r1
  )
  kpAddLabels(kp,
    labels = names(rna_tracks)[i],
    r0 = at$r0,
    r1 = at$r1,
    cex = 0.75,
    label.margin = 0.045
  )
}

locusEpiPlot(locus = abl2_region, scale_max = 1)

## 2169 genes were dropped because they have exons located on both strands of
## the same reference sequence or on more than one reference sequence, so cannot
## be represented by a single genomic range.
## Use 'single.strand.genes.only=FALSE' to get all the genes in a GRangesList
## object, or use suppressMessages() to suppress this message.

```



Main Analysis - Compartment Quantification in Context of Deletion Hotspots

Read in BED file for regions of deletion hotspots that overlap with B compartments

```
del_hotspot_b_overlap <- read_delim(
  file = paste0(base_data_path,
    "compartmentQuantification/deletionsBCompartmentOverlap.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "id"),
  show_col_types = F
) %>%
  mutate(length = end - start)
```

Calculate the percentage of each deletion hotspot region is within B compartments

```
filter(del_hotspot_b_overlap, id == "D3-B") %>%
  dplyr::select(length) /
  dplyr::filter(del_hotspot_b_overlap, id == "D3") %>%
  dplyr::select(length)
```

length
0.4496457

```

filter(del_hotspot_b_overlap, id == "D4-B") %>%
  dplyr::select(length) /
  dplyr::filter(del_hotspot_b_overlap, id == "D4") %>%
    dplyr::select(length)

```

$$\begin{array}{c} \overline{\text{length}} \\ \hline 1 \end{array}$$

```

(dplyr::filter(del_hotspot_b_overlap, id == "D5-B") %>%
  dplyr::select(length) %>%
  sum()) /
  (dplyr::filter(del_hotspot_b_overlap, id == "D5") %>%
    dplyr::select(length))

```

$$\begin{array}{c} \overline{\text{length}} \\ \hline 0.9031519 \end{array}$$

Main Analysis - Compartment Quantification in Context of Gain(1q)

Read in BED files for chromosome 1 A/B compartments (without gain1q regions) and GISTIC 1q gain regions

```

a_compartments <- read_delim(
  file = paste0(base_data_path,
                "compartmentQuantification/U266-MboI_A_compartments_chr1_minus_gain1q.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end"),
  show_col_types = F
) %>%
  mutate(length = end - start)

b_compartments <- read_delim(
  file = paste0(base_data_path,
                "compartmentQuantification/U266-MboI_B_compartments_chr1_minus_gain1q.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end"),
  show_col_types = F
) %>%
  mutate(length = end - start)

gain2 <- read_delim(
  file = paste0(base_data_path,
                "compartmentQuantification/G2_1q21.1_scored.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "section", "score"),
  show_col_types = F
) %>%
  mutate(length = end - start)

```

```

gain3 <- read_delim(
  file = paste0(base_data_path,
                "compartmentQuantification/G3_1q21.3_scored.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "section", "score"),
  show_col_types = F
) %>%
  mutate(length = end - start)

gain4 <- read_delim(
  file = paste0(base_data_path,
                "compartmentQuantification/G4_1q22_scored.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "section", "score"),
  show_col_types = F
) %>%
  mutate(length = end - start)

gain5 <- read_delim(
  file = paste0(base_data_path,
                "compartmentQuantification/G5_1q23.3_scored.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "section", "score"),
  show_col_types = F
) %>%
  mutate(length = end - start)

gain6 <- read_delim(
  file = paste0(base_data_path,
                "compartmentQuantification/G6_1q24.2_scored.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "section", "score"),
  show_col_types = F
) %>%
  mutate(length = end - start)

gain7 <- read_delim(
  file = paste0(base_data_path,
                "compartmentQuantification/G7_1q25.2_scored.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "section", "score"),
  show_col_types = F
) %>%
  mutate(length = end - start)

```

Separately aggregate data for whole GISTIC gain regions and all per A/B compartment

```

aggregated_whole_gain_regions <- rbind(
  gain2[1, ],
  gain3[1, ],
  gain4[1, ],
  gain5[1, ],
  gain6[1, ],

```

```

    gain7[1, ]
)

aggregated_gain_region_compartments <- rbind(
  gain2[-1, ],
  gain3[-1, ],
  gain4[-1, ],
  gain5[-1, ],
  gain6[-1, ],
  gain7[-1, ]
)

```

Calculate raw length of A/B compartments and find proportion in relation to whole chromosome

```

chrom1_telomere_length <- 10000 * 2
chrom1_centromere_length <- 124932724 - 122026459
chrom1_whole_length <- 248956422 -
  chrom1_telomere_length -
  chrom1_centromere_length -
  sum(aggregated_whole_gain_regions$length)

a_compartment_length <- sum(a_compartments$length)
b_compartment_length <- sum(b_compartments$length) - chrom1_centromere_length

chrom1_a_compartment_proportion <- a_compartment_length / chrom1_whole_length
chrom1_b_compartment_proportion <- b_compartment_length / chrom1_whole_length
chrom1_unassigned_proportion <- (chrom1_whole_length -
  a_compartment_length -
  b_compartment_length) / chrom1_whole_length

```

Calculate compartment proportion of each GISTIC 1q gain region

```

gain_region_a_compartment_proportion <- aggregated_gain_region_compartments %>%
  dplyr::filter(section == "a_compartment") %>%
  dplyr::select(length) %>%
  sum() / sum(aggregated_whole_gain_regions$length)

gain_region_b_compartment_proportion <- aggregated_gain_region_compartments %>%
  dplyr::filter(section == "b_compartment") %>%
  dplyr::select(length) %>%
  sum() / sum(aggregated_whole_gain_regions$length)

gain_region_unassigned_compartment_proportion <- aggregated_gain_region_compartments %>%
  dplyr::filter(section == "unassigned") %>%
  dplyr::select(length) %>%
  sum() / sum(aggregated_whole_gain_regions$length)

```

Calculate Chi-squared test statistic of A/B compartments in 1q gain regions vs whole chromosome

```

# Using total length of A compartment and length in base pairs
# prop.test(x = c(a_compartment_length, 3785218),
#            n = c(chrom1_whole_length, sum(aggregated_whole_gain_regions$length)))

```

```

# Using 100kb bins
prop.test(
  x = c(a_compartment_length / 100000, 3785218 / 100000),
  n = c(chrom1_whole_length / 100000, sum(aggregated_whole_gain_regions$length) / 100000)
)

##
## 2-sample test for equality of proportions with continuity correction
##
## data: c(a_compartment_length/100000, 3785218/100000) out of c(chrom1_whole_length/100000, sum(aggregat
## X-squared = 11.044, df = 1, p-value = 0.0008895
## alternative hypothesis: two.sided
## 95 percent confidence interval:
## -0.3686745 -0.1020363
## sample estimates:
## prop 1    prop 2
## 0.4540409 0.6893963

```

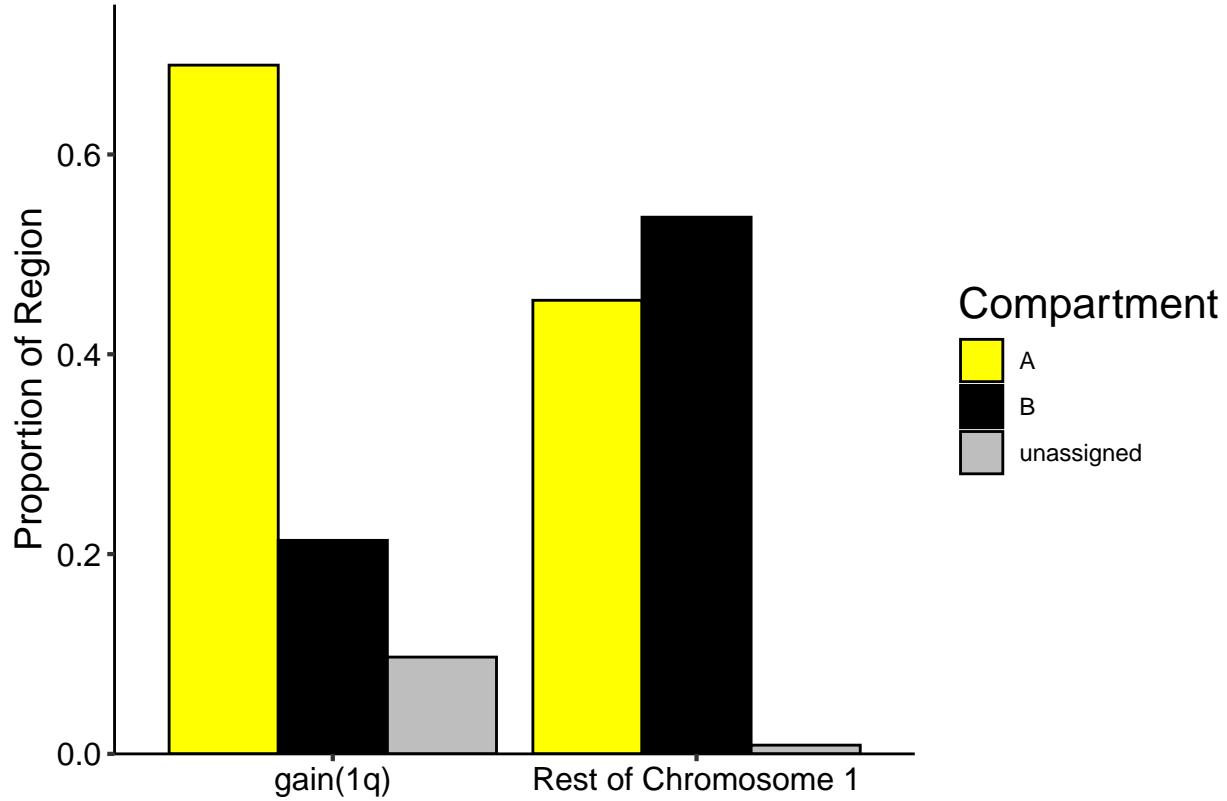
Graphical representation of compartment proportion across chromosome 1 and 1q

```

df <- tibble(
  "region" = c(
    "gain(1q)", "gain(1q)", "gain(1q)",
    "Rest of Chromosome 1", "Rest of Chromosome 1", "Rest of Chromosome 1"
  ),
  "compartment" = c("A", "B", "unassigned", "A", "B", "unassigned"),
  "proportion" = c(
    gain_region_a_compartment_proportion,
    gain_region_b_compartment_proportion,
    gain_region_unassigned_compartment_proportion,
    chrom1_a_compartment_proportion,
    chrom1_b_compartment_proportion,
    chrom1_unassigned_proportion
  )
)

ggplot(df) +
  geom_col(aes(x = as.factor(region), y = proportion, fill = compartment),
           position = "dodge",
           color = "black")
) +
  scale_fill_discrete("Compartment", type = c("#FFFF01", "#000000", "gray")) +
  scale_y_continuous(expand = c(0, 0), breaks = seq(0, 0.75, 0.2), limits = c(0, 0.75)) +
  xlab("") +
  ylab("Proportion of Region")

```



Calculate Mann Whitney U test statistic for compartment scores in 1q gain regions vs whole chromosome

```
chrom1_compartment_scores <- read_delim(
  file = paste0(
    base_data_path,
    "compartmentQuantification/U266-MboI_compartments_chr1_minus_gain1q.scores.bedGraph"),
  delim = "\t",
  skip = 1,
  col_names = c("chrom", "start", "end", "score"),
  show_col_types = F
)
```

Graphical representation of compartment score across chromosome 1 and 1q

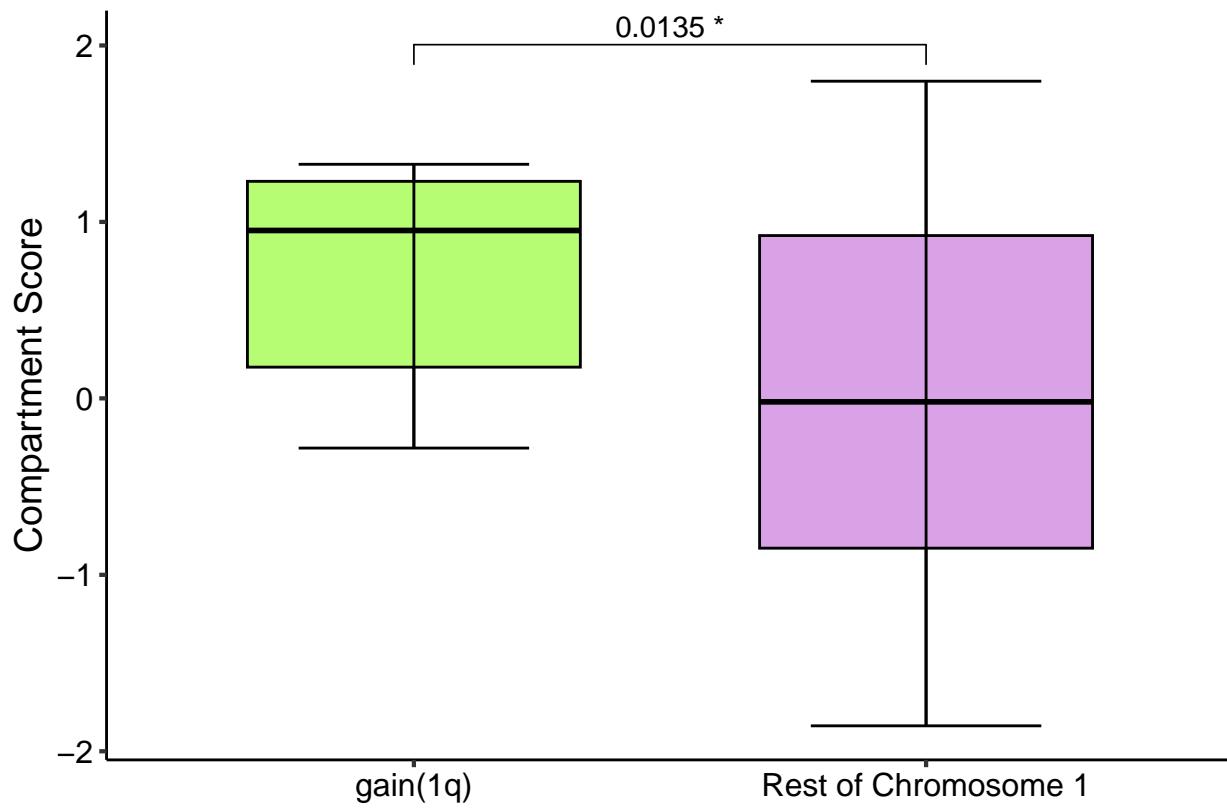
```
df2 <- rbind(
  aggregated_gain_region_compartments %>%
    dplyr::mutate("region" = "gain(1q)") %>%
    dplyr::filter(score != ".") %>%
    dplyr::select(score, region),
  chrom1_compartment_scores %>%
    dplyr::mutate("region" = "Rest of Chromosome 1") %>%
    dplyr::select(score, region)
)
df2$region <- factor(df2$region)
df2$score <- as.numeric(df2$score)
```

```

compartment_score_data_1q_stat_test <- df2 %>%
  rstatix::wilcox_test(
    formula = score ~ region,
    alternative = "greater",
    p.adjust.method = "fdr"
  ) %>%
  rstatix::add_significance() %>%
  rstatix::add_xy_position(x = "region")

ggplot(df2) +
  geom_boxplot(aes(x = region, y = as.numeric(score)),
    width = 0.65,
    fill = c("lawngreen", "mediumorchid"),
    color = "black",
    alpha = 0.55,
    outlier.shape = NA
  ) +
  stat_boxplot(aes(x = region, y = as.numeric(score)),
    geom = "errorbar",
    width = 0.45
  ) +
  stat_pvalue_manual(
    data = compartment_score_data_1q_stat_test,
    label = "{p} {p.signif}",
    bracket.nudge.y = 0.15
  ) +
  xlab("") +
  ylab("Compartment Score")

```



Main Analysis - Intersection of Super-Enhancers with Hotspots

Read in master list of genome-wide super-enhancers and BED file of intersection with gain and deletion hotspots

```
master_list_super_enhancers <- read_delim(
  file = paste0(base_data_path,
                "originalSuperEnhancersAnalysis/U266_rose2_super_enhancers.txt"),
  delim = "\t",
  skip = 1,
  col_names = T,
  show_col_types = F
) %>%
  arrange(desc(U266.sort.bam)) %>%
  mutate(rank = seq(1, 963, 1))

hotspot_super_enhancers <- read_delim(
  file = paste0(
    base_data_path,
    "originalSuperEnhancersAnalysis/U266_rose2_super_enhancers_intersect.gains.deletions.bed"),
  delim = "\t",
  col_names = F,
  show_col_types = F
)
```

```

chrom_sizes <- read_delim(
  file = paste0(base_data_path,
    "originalSuperEnhancersAnalysis/hg38ChromSizes.txt"),
  delim = "\t",
  col_names = c("CHROM", "LENGTH"),
  show_col_types = F
)[1:23, ]

```

Calculate and visualize the number of super-enhancers per 5Mb of each chromosome

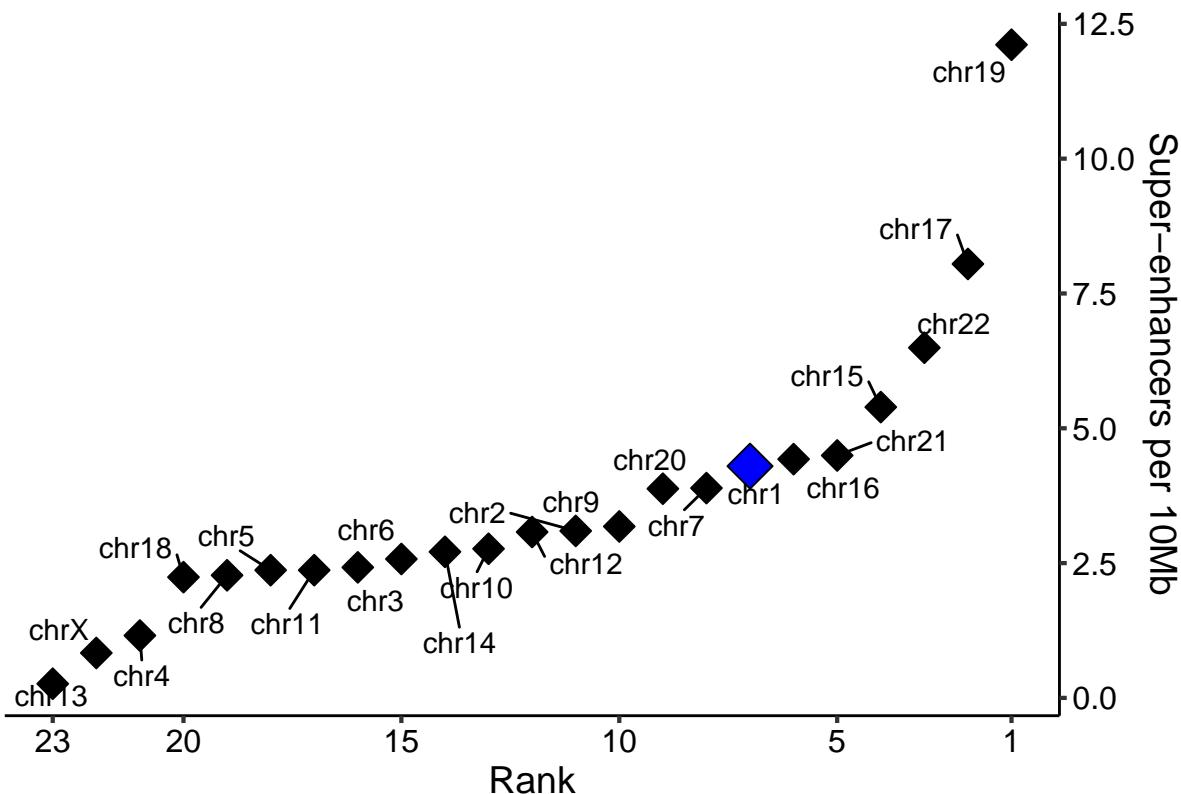
```

per_chromosome_se <- master_list_super_enhancers %>%
  dplyr::group_by(CHROM) %>%
  dplyr::count(isSuper) %>%
  dplyr::select(CHROM, n)

se_per_10Mb <- left_join(
  x = chrom_sizes,
  y = per_chromosome_se,
  by = "CHROM"
) %>%
  dplyr::mutate(SEper10MB = (n / LENGTH) * 10^7) %>%
  dplyr::arrange(desc(SEper10MB))

ggplot(se_per_10Mb) +
  geom_point(aes(x = seq(1, 23, 1), y = SEper10MB),
    shape = 23,
    fill = case_when(
      se_per_10Mb$CHROM == "chr1" ~ "blue",
      !se_per_10Mb$CHROM == "chr1" ~ "black"
    ),
    size = case_when(
      se_per_10Mb$CHROM == "chr1" ~ 6,
      !se_per_10Mb$CHROM == "chr1" ~ 4
    )
  ) +
  geom_text_repel(
    data = se_per_10Mb,
    aes(
      x = seq(1, 23, 1),
      y = SEper10MB,
      label = CHROM
    ),
    force = 100
  ) +
  scale_x_reverse(
    limits = c(23, 1),
    breaks = c(23, seq(20, 5, -5), 1)
  ) +
  scale_y_continuous(position = "right") +
  theme(plot.margin = margin(0.5, 0.5, 0.5, 0.5, "cm")) +
  xlab("Rank") +
  ylab("Super-enhancers per 10Mb")

```



Generate rank plot of all genome-wide SEs

```

rose_se_hotspot_intersect_list <-
  master_list_super_enhancers[master_list_super_enhancers$REGION_ID %in%
                                hotspot_super_enhancers$X4, ][, c(1:7, 12, 15)]

write_delim(
  x = rose_se_hotspot_intersect_list,
  file = paste0(base_data_path,
                "originalSuperEnhancersAnalysis/hotspotRoseIntersect.txt"),
  delim = "\t",
  col_names = T
)

super_enhancer_labels <- master_list_super_enhancers %>%
  dplyr::filter(REGION_ID %in% hotspot_super_enhancers$X4) %>%
  dplyr::select(REGION_ID, rank, U266.sort.bam) %>%
  dplyr::mutate(label = c(
    "(12)TENT5C", "(25)MCL1", "(35)BTG2", "(93)", "(197)CDKN2C,FAF1",
    "(226)SLAMF7", "(231)NBPF20", "(281)", "(346)NOTCH2NL", "(357)", "(362)",
    "(396)SEC22B", "(417)", "(526)", "(539)", "(596)DIPK1A",
    "(751)SLAMF1", "(917)RALGPS2", "(926)NADK"
  ))
}

ggplot(master_list_super_enhancers) +
  geom_point(

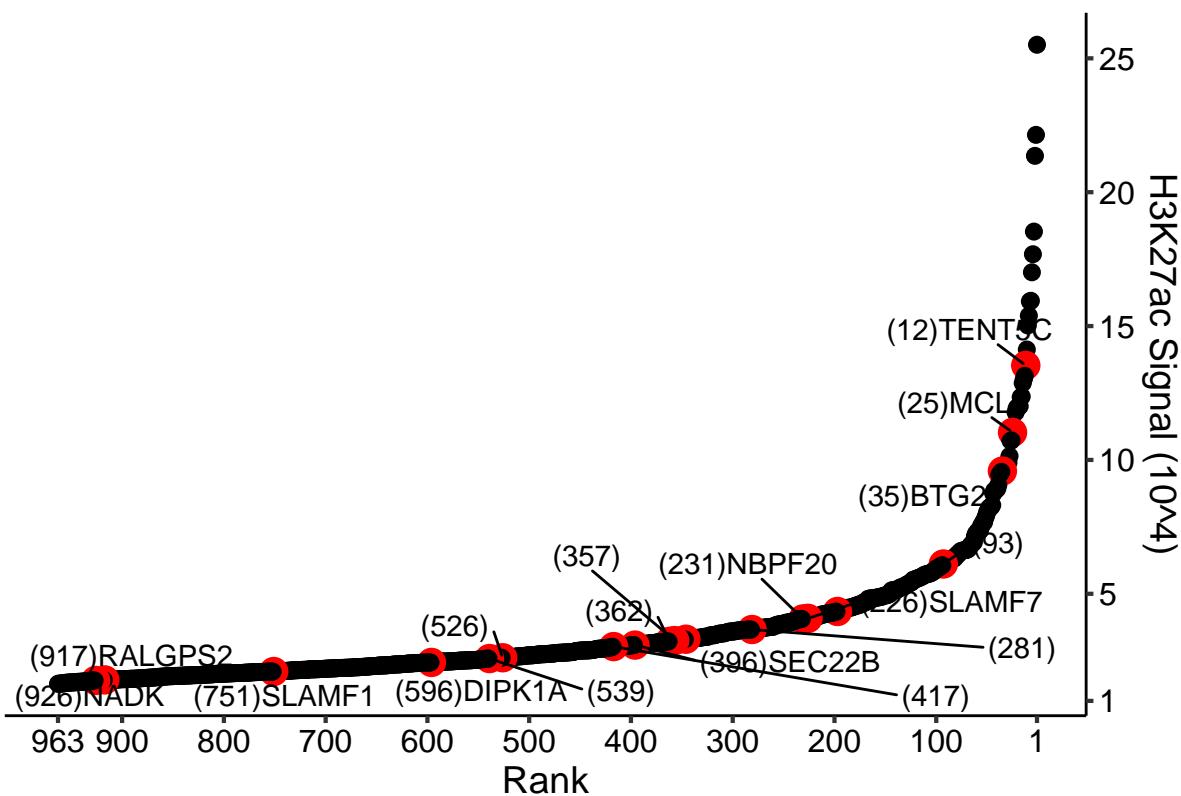
```

```

aes(
  x = rank,
  y = sort(U266.sort.bam / 10^4, decreasing = T)
),
color = case_when(
  master_list_super_enhancers$REGION_ID %in% hotspot_super_enhancers$X4 ~ "red",
  !master_list_super_enhancers$REGION_ID %in% hotspot_super_enhancers$X4 ~ "black"
),
size = case_when(
  master_list_super_enhancers$REGION_ID %in% hotspot_super_enhancers$X4 ~ 4.5,
  !master_list_super_enhancers$REGION_ID %in% hotspot_super_enhancers$X4 ~ 2.5
)
) +
geom_text_repel(
  data = super_enhancer_labels,
  aes(
    x = rank,
    y = sort(U266.sort.bam / 10^4, decreasing = T),
    label = label
  ),
  force = 100
) +
scale_x_reverse(
  limits = c(length(master_list_super_enhancers$enhancerRank) + 4, 1),
  breaks = c(length(master_list_super_enhancers$enhancerRank), seq(900, 100, -100), 1)
) +
scale_y_continuous(
  position = "right",
  breaks = c(1, seq(5, 25, 5))
) +
theme(plot.margin = margin(0.5, 0.5, 0.5, 0.5, "cm")) +
xlab("Rank") +
ylab("H3K27ac Signal (10^4)")

## Warning: ggrepel: 2 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps

```



Main Analysis - Histone Signal Enrichment in Hotspots

Read in BED files that contain mean histone signal across 1kb windows for hotspots and rest of chr1

```

gains_h3k27ac_1kb <- read_delim(
  file = paste0(
    base_data_path,
    "originalSuperEnhancersAnalysis/gainsAcompartments.meanH3K27ac.1kb.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "signal"),
  col_types = "ciidi"
) %>%
  dplyr::mutate(region = "Gain Hotspots") %>%
  dplyr::filter(!is.na(signal))

gains_h3k4me1_1kb <- read_delim(
  file = paste0(
    base_data_path,
    "originalSuperEnhancersAnalysis/gainsAcompartments.meanH3K4me1.1kb.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "signal"),
  col_types = "ciidi"
) %>%
  dplyr::mutate(region = "Gain Hotspots") %>%

```

```

dplyr::filter(!is.na(signal))

deletions_h3k27ac_1kb <- read_delim(
  file = paste0(
    base_data_path,
    "originalSuperEnhancersAnalysis/deletionsAcompartments.meanH3K27ac.1kb.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "signal"),
  col_types = "ciidi"
) %>%
  dplyr::mutate(region = "Deletion Hotspots") %>%
  dplyr::filter(!is.na(signal))

deletions_h3k4me1_1kb <- read_delim(
  file = paste0(
    base_data_path,
    "originalSuperEnhancersAnalysis/deletionsAcompartments.meanH3K4me1.1kb.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "signal"),
  col_types = "ciidi"
) %>%
  dplyr::mutate(region = "Deletion Hotspots") %>%
  dplyr::filter(!is.na(signal))

a_compartment_h3k27ac_1kb <- read_delim(
  file = paste0(
    base_data_path,
    "originalSuperEnhancersAnalysis/Acompartments.nogains.nodeletions.meanH3K27ac.1kb.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "signal"),
  col_types = "ciidi"
) %>%
  dplyr::mutate(region = "Rest of Chromosome 1") %>%
  dplyr::filter(!is.na(signal))

a_compartment_h3k4me1_1kb <- read_delim(
  file = paste0(
    base_data_path,
    "originalSuperEnhancersAnalysis/Acompartments.nogains.nodeletions.meanH3K4me1.1kb.bed"),
  delim = "\t",
  col_names = c("chrom", "start", "end", "signal"),
  col_types = "ciidi"
) %>%
  dplyr::mutate(region = "Rest of Chromosome 1") %>%
  dplyr::filter(!is.na(signal))

```

Visualize the mean signal distribution between the gain/deletion hotspots and the background A compartments of rest of chr1

```

h3k27ac_stat_test <- rbind(
  gains_h3k27ac_1kb,
  deletions_h3k27ac_1kb,
  slice_sample(a_compartment_h3k27ac_1kb,

```

```

        n = 5000
    )
)
h3k27ac_stat_test$region <- factor(h3k27ac_stat_test$region)
h3k27ac_stat_test$signal <- log2(h3k27ac_stat_test$signal + 1)
h3k27ac_stat_test$pvalue <- h3k27ac_stat_test %>%
  rstatix::wilcox_test(
    formula = signal ~ region,
    alternative = "two.sided",
    p.adjust.method = "BH",
    mu = 0,
    paired = FALSE
) %>%
  rstatix::add_significance() %>%
  rstatix::add_xy_position(x = "region")

h3k27ac_boxplot <- ggplot(data = h3k27ac_stat_test) +
  geom_jitter(aes(x = region, y = signal),
              color = "black",
              size = 0.55,
              alpha = 0.45,
              width = 0.15
) +
  geom_boxplot(aes(x = region, y = signal),
               width = 0.65,
               fill = c("#0C3FE8", "#FA160C", "lightgrey"),
               color = "black",
               alpha = 0.55,
               outlier.shape = NA
) +
  stat_boxplot(aes(x = region, y = signal),
               geom = "errorbar",
               width = 0.45
) +
  stat_pvalue_manual(
    data = h3k27ac_stat_test$pvalue,
    label = "{p.adj.signif}",
    bracket.nudge.y = 0.5
) +
  xlab("") +
  ylab("log2(signal + 1)") +
  ggtitle("H3K27ac") +
  theme(plot.title = element_text(hjust = 0.5))

h3k4me1_stat_test <- rbind(
  gains_h3k4me1_1kb,
  deletions_h3k4me1_1kb,
  slice_sample(a_compartment_h3k4me1_1kb,
               n = 5000
  )
)
h3k4me1_stat_test$region <- factor(h3k4me1_stat_test$region)

```

```

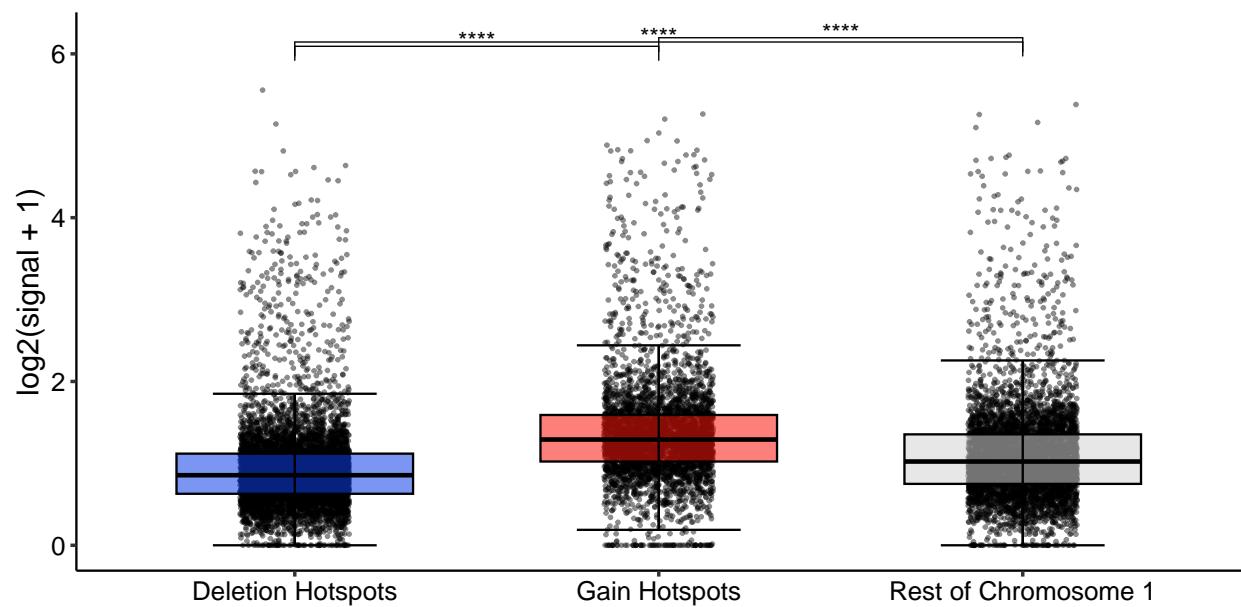
h3k4me1_stat_test$signal <- log2(h3k4me1_stat_test$signal + 1)
h3k4me1_stat_test_pvalue <- h3k4me1_stat_test %>%
  rstatix::wilcox_test(
    formula = signal ~ region,
    alternative = "two.sided",
    p.adjust.method = "BH",
    mu = 0,
    paired = FALSE
  ) %>%
  rstatix::add_significance() %>%
  rstatix::add_xy_position(x = "region")

h3k4me1_boxplot <- ggplot(data = h3k4me1_stat_test) +
  geom_jitter(aes(x = region, y = signal),
    color = "black",
    size = 0.55,
    alpha = 0.45,
    width = 0.15
  ) +
  geom_boxplot(aes(x = region, y = signal),
    width = 0.65,
    fill = c("#0C3FE8", "#FA160C", "lightgrey"),
    color = "black",
    alpha = 0.55,
    outlier.shape = NA
  ) +
  stat_boxplot(aes(x = region, y = signal),
    geom = "errorbar",
    width = 0.45
  ) +
  stat_pvalue_manual(
    data = h3k4me1_stat_test_pvalue,
    label = "{p.adj.signif}",
    bracket.nudge.y = 0.25
  ) +
  xlab("") +
  ylab("log2(signal + 1)") +
  ggtitle("H3K4me1") +
  theme(plot.title = element_text(hjust = 0.5))

ggarrange(
  plotlist = list(h3k27ac_boxplot, h3k4me1_boxplot),
  ncol = 1,
  nrow = 2
)

```

H3K27ac



H3K4me1

