PCA Check after Gem-Mapping

Ricky Lim

April 25, 2013

```
Filename: PCACheck.Rnw
Working directory: /TEMP_DDN/users/gfilion/rlim/E14_ColorChromatin/PCACheckGemMapE14
```

1 Load the input table

```
bigTable <- read.delim("input/E14_matBin.bed")
head(bigTable)[1:5]
dim(bigTable)

matTable <- bigTable[, 4:31]
head(matTable)[1:5]
dim(matTable)
```

2 Load the Annotation's file

```
annot <- read.delim("idE14_Fastq_annotation.txt", header = F)
dim(annot)
head(annot)[1:5]

# grep the file numbers
id_ <- sub(".*-(\\d{3}|\\d{3}[ab]).*", "\\1", annot$V10)
print(id_)

# double check for the id mapping from annotation file with the colnames of the matrix
annot_dataset <- paste("X", id_, sep = "") %in% colnames(matTable)

# get the samples given the file no
sample_dataset <- annot$V12[annot_dataset]
head(sample_dataset)[1:5]
length(sample_dataset)

# function to get the file number in the loaded dataset given the sampleName
getProfileId <- function(sampleName) {
    library(stringr)
    # get only the partial match (in case!)
    sample_dataset <- str_extract(sample_dataset, sampleName)
    sample_ <- (sample_dataset == sampleName)
    sample_no <- id_[annot_dataset][sample_]
```

```
    sample_no <- sample_no[!is.na(sample_no)]
    sample_names <- paste("X", sample_no, sep = "")
    return(sample_names)
}

# e.g
getProfileId("Input")
```

3 Assigning NAs

NAs were assigned for rows(genomic coordinates) in which in all profiles they were no reads

```
matTable[which(rowSums(matTable) == 0), ] <- NA
```

4 matTable no NAs

```
matTableNoNA <- matTable[complete.cases(matTable), ]
sum(rowSums(matTableNoNA[, ]) == 0)
head(matTableNoNA)[1:5]
nrow(matTableNoNA)/nrow(matTable)
```

5 Create PCA Object

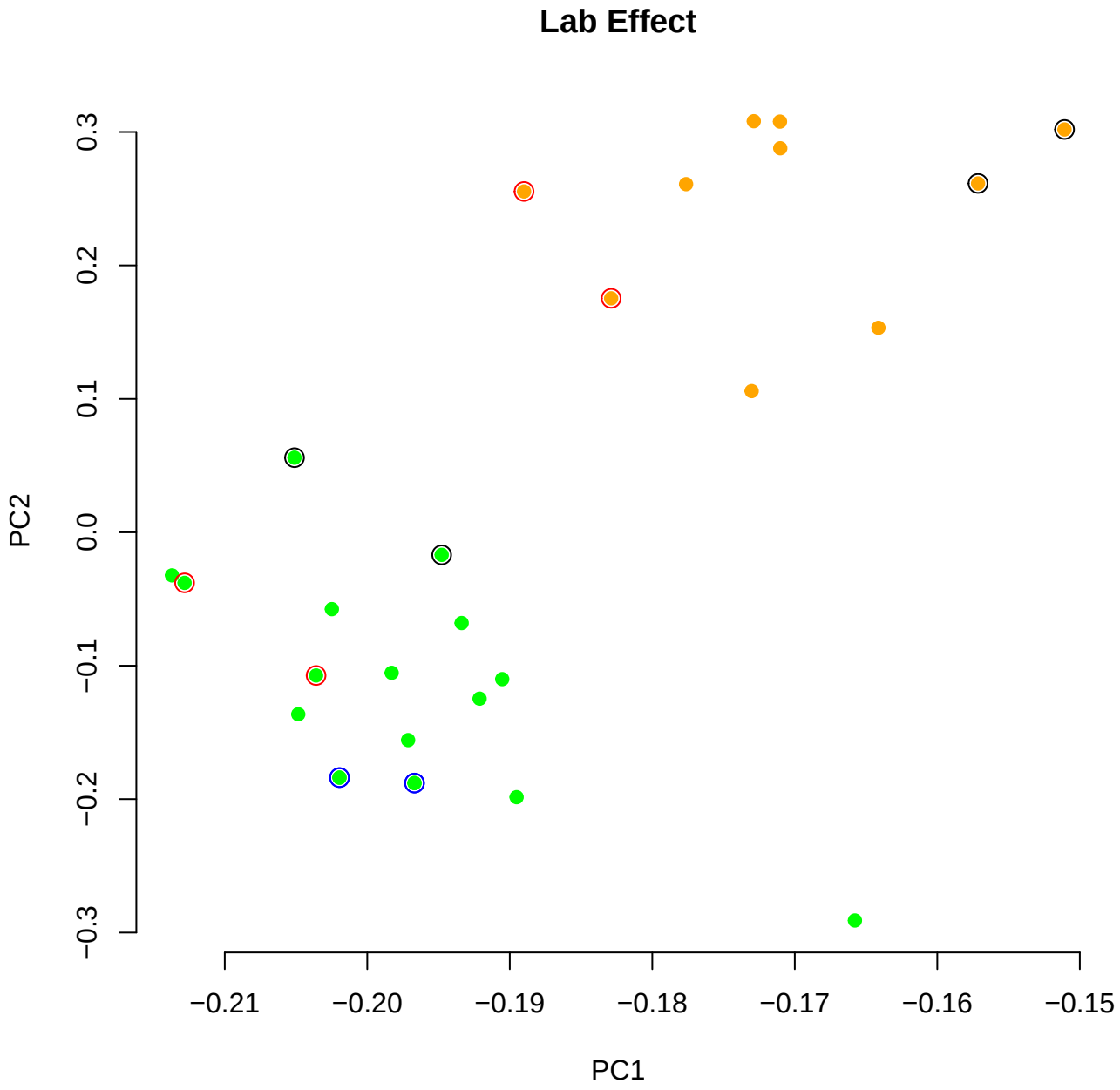
```
log_mat <- log(matTableNoNA + 1)
pca_mat <- prcomp(log_mat, scale. = T)
plot(pca_mat, main = "Scree Plot")
```

6 PCA Labs

```
# order the PCA rotation matrix
lab_PCA <- pca_mat$rotation[order(rownames(pca_mat$rotation)), ]
lab_PCA[c(1, 6, 18, 27), c(1, 2)]
getProfileId("H3K4me1")

# get the lab's info
lab_info <- annot$V1[annot_dataset]
length(rownames(pca_mat$rotation))
lab_info

# plot lab's effect
pdf("figs/lab_effect.pdf", useDingbats = FALSE)
plot(lab_PCA[, 1], lab_PCA[, 2], col = c("orange", "green")[lab_info], pch = 19, xlab = "PC1",
    ylab = "PC2", frame = F, main = "Lab Effect")
```

```
legend(y = 0.3, x = 0.205, pch = 19, cex = 1, col = c("orange", "green"), legend = levels(lab_info),
    box.lwd = 0, box.col = "white", bg = "white")
legend(y = 0.23, x = 0.205, pch = 1, cex = 1, col = c("blue", "red", "black"), legend = c("input",
    "H3K4me1", "H3K4me3"), box.lwd = 0, box.col = "white", bg = "white")

points(lab_PCA[getProfileId("Input"), 1], lab_PCA[getProfileId("Input"), 2], col = "blue",
    cex = 1.5)

points(lab_PCA[getProfileId("H3K4me1"), 1], lab_PCA[getProfileId("H3K4me1"), 2], col = "red",
    cex = 1.5)

points(lab_PCA[getProfileId("H3K4me3"), 1], lab_PCA[getProfileId("H3K4me3"), 2], col = "black",
    cex = 1.5)
dev.off()
```

The profiles from Ping and Synder's lab show two obvious clusters.


Lab Effect
PC2
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.21
−0.20
−0.19
−0.18
−0.17
−0.16
−0.15
PC1

7 Input Correlations

```
cor(matTableNoNA[, getProfileId("Input")])
```

```
##        X011a  X011b  X016a  X016b
## X011a 1.0000 0.9827 1.0000 0.9827
## X011b 0.9827 1.0000 0.9827 1.0000
## X016a 1.0000 0.9827 1.0000 0.9827
## X016b 0.9827 1.0000 0.9827 1.0000
```

Note samples id 016 and 011 are duplicated. In the PCA's figure (above, the loadings plots), these duplicated samples were on top of each other. They were similar samples that have been deposited in GEO database twice!.

8 PCA pairs

Check the figures of PCA pairs. The aim is to check if there is a chromosomal duplication or deletion (chromosomal abberation) that creates the clusters between Ping and Snyder labs.

```
# chromosomes pairs pairing from the longest to the shortest, chr1 chr20, chr2 chr10, ...
c1 <- str_c("chr", 1:10)
c2 <- str_c("chr", 20:11)
cpairs <- cbind(c1, c2)

chr_bigTable <- bigTable[, c(1, 4:31)]
head(chr_bigTable)
dim(chr_bigTable)

# put NA if all the columns in a row contain only zero reads
chr_bigTable[which(rowSums(chr_bigTable[, 2:29]) == 0), 2:29] <- NA

# remove these NAs
chr_bigTable <- chr_bigTable[complete.cases(chr_bigTable), ]
head(chr_bigTable)
dim(chr_bigTable)
sum(rowSums(chr_bigTable[, 2:29]) == 0)

# create the matrixes of pairs
results <- list()
for (i in 1:nrow(cpairs)) {
    pair1 <- chr_bigTable[chr_bigTable$chr == cpairs[i, 1], 2:29]
    pair2 <- chr_bigTable[chr_bigTable$chr == cpairs[i, 2], 2:29]
    pair <- rbind(pair1, pair2)
    pairname <- paste(cpairs[i, 1], cpairs[i, 2], sep = "")
    # assign(pairname, pair)
    results[[pairname]] <- rbind(pair1, pair2)
}

# plot PCA on these pair matrices
for (i in names(results)) {
    log_mat <- log(results[[i]] + 1)
    pca_mat <- prcomp(log_mat, scale. = T)
```

```
    lab_PCA <- pca_mat$rotation[order(rownames(pca_mat$rotation)), ]
    pic_f <- paste("figs", i, sep = "/")
    pdf(paste(pic_f, ".pdf", sep = ""), useDingbats = FALSE)
    plot(lab_PCA[, 1], lab_PCA[, 2], col = c("red", "green")[lab_info], pch = 19, xlab = "PC1",
        ylab = "PC2", frame = F, main = i)
    dev.off()
}
```

The artefacts clusters between Ping and Snyder's lab seem not to be affected by single chromosomals duplication or insertion. This suggests for a possible genome-wide bias between Ping and Snyder's profiles.

8.1 Genomic Loci that might cause bias

```
chr1chr20Log_mat <- log(results[["chr1chr20"]] + 1)
head(chr1chr20Log_mat)
dim(chr1chr20Log_mat)
pca_chr1chr20 <- prcomp(chr1chr20Log_mat, scale. = T)
names(pca_chr1chr20)
head(pca_chr1chr20$x)

# plot(pca_chr1chr20$x[,1], pca_chr1chr20$x[,2], xlab='PC1', ylab='PC2',
# main='Chr1Chr20', cex=2) identify(pca_chr1chr20$x) rownames that are clustered together
# exclusively at the bottom left corner
rownames_diff <- c(2521, 2522, 6207, 6426, 8206, 8207, 8755, 8756, 13345, 13346, 28616, 28617,
    30445, 30841, 34689, 35305, 43541, 43542)

chr1chr20 <- results[["chr1chr20"]]
head(chr1chr20)
head(bigTable)

getProfileId("H3K4me1")
coordinates <- bigTable[, c(1:30)]
selected_coordinates <- coordinates[rownames_diff, ]
head(selected_coordinates)

# map this to hg19 library(ggbio) library(rtracklayer) source('data.frame2GRanges.R')
# select_ <- data.frame2GRanges(selected_coordinates) head(select_)

# library(BSgenome.Mmusculus.UCSC.mm10) chr.len = seqlengths(Mmusculus) exclude
# chromosomes with suffix '_' , 'M', 'Het', 'extra'. chr.len =
# chr.len[grep('_|M|U|Het|extra', names(chr.len), invert = T)]

# select_ = keepSeqlevels(select_, names(chr.len)) seqlevels(select_) = names(chr.len)
# seqlengths(select_) = (chr.len) p <- autoplot(select_, layout = 'karyogram')
```

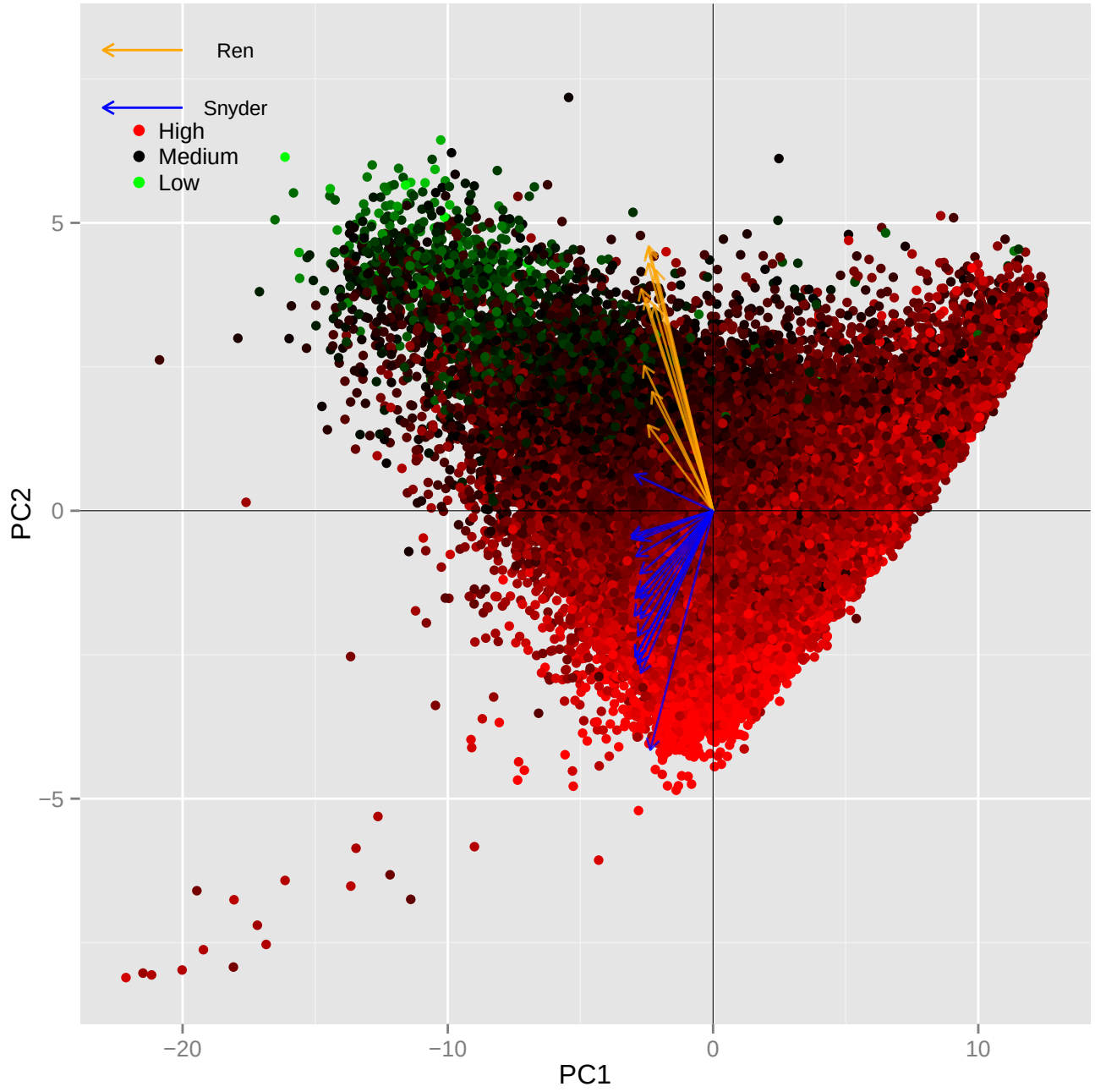
8.2 Check for the Sequence Content

Sequence content being checked was AT and low-complexity.

```
# map the coordinate of big table with the coordinate of the sequence contents/bin
head(bigTable)
chr1_big <- bigTable[bigTable$chr == "chr1", ]
nrow(chr1_big)
head(chr1_big)

# seqContent: AT and LowComplexity
chr1_seqContent <- read.delim("chr1_seqContent.bed", header = FALSE)
head(chr1_seqContent)
colnames(chr1_seqContent) <- c("chr", "start", "end", "AT", "LowComplexity")
chr1_seqCheck <- cbind(chr1_big, chr1_seqContent[, c("AT", "LowComplexity")])
dim(chr1_seqCheck)
head(chr1_seqCheck)
chr1_seqCheck[, 4:31][which(rowSums(chr1_seqCheck[, 4:31]) == 0), ] <- NA
chr1_seqCheck <- chr1_seqCheck[complete.cases(chr1_seqCheck), ]

# construct pca
AT_content <- matrix(chr1_seqCheck[, 32])
LowSeq_content <- matrix(chr1_seqCheck[, 33])
rownames(AT_content) <- rownames(chr1_seqCheck)
head(AT_content)
rownames(LowSeq_content) <- rownames(chr1_seqCheck)
head(LowSeq_content)

log_matSeq <- log(chr1_seqCheck[, c(4:31)] + 1)
pca_matSeq <- prcomp(log_matSeq, scale. = T)
names(pca_matSeq)
pca_matSeqScores <- cbind(pca_matSeq$x, AT_content, LowSeq_content)
head(pca_matSeqScores)
dim(pca_matSeqScores)
rownames(pca_matSeq$rotation) <- lab_info

# biplot PCAs for AT-content biplot
# function:http://stackoverflow.com/questions/6578355/plotting-pca-biplot-with-ggplot2
col_gradientAT <- colorRampPalette(c("green", "green", "green", "black", "red", "red", "red"))(1024)
lab_info

PCbiplot <- function(PC, x = "PC1", y = "PC2") {
    # PC being a prcomp object
    data <- data.frame(obsnames = row.names(PC$x), PC$x)
    # pca_matSeqScores col 29: AT-content
    plot <- ggplot(data, aes_string(x = x, y = y)) + geom_point(aes(label = obsnames), color = col_gradi
        29] * 1024])
    plot <- plot + geom_hline(aes(0), size = 0.2) + geom_vline(aes(0), size = 0.2)
    datapc <- data.frame(varnames = rownames(PC$rotation), PC$rotation)
    mult <- min((max(data[, y]) - min(data[, y])/(max(datapc[, y]) - min(datapc[, y]))), (max(data[,
        x]) - min(data[, x])/(max(datapc[, x]) - min(datapc[, x]))))
    datapc <- transform(datapc, v1 = 0.7 * mult * (get(x)), v2 = 0.7 * mult * (get(y)))
    plot <- plot + geom_segment(data = datapc, aes(x = 0, y = 0, xend = v1, yend = v2), arrow = arrow(le
        "cm")), alpha = 0.75, color = c("orange", "blue")[lab_info])
    plot
}
```

```
# create biplot

# color gradient for low, medium, to high
library(ggplot2)
library(grid)
pdf("figs/AT_content.pdf", useDingbats = FALSE)
p <- PCbiplot(pca_matSeq)
plot.new()
P <- p + annotate("segment", x = -20, xend = -23, y = 8, yend = 8, size = 0.5, arrow = arrow(length = ur
    "cm")), color = "orange") + annotate("text", x = -18, y = 8, label = "Ren", size = 3) +
    annotate("segment", x = -20, xend = -23, y = 7, yend = 7, size = 0.5, arrow = arrow(length = unit(0.
        "cm")), color = "blue") + annotate("text", x = -18, y = 7, label = "Snyder", size = 3)
P
legend("topleft", pch = 19, cex = 0.8, col = c("red", "black", "green"), legend = c("High",
    "Medium", "Low"), box.lwd = 0, box.col = "transparent", bg = "transparent")

dev.off()
```

From PC2, the biplot shows that there is a gradient separation in the AT-content between Ren and Snyder's labs, as shown in figure below.


Ren
Snyder
High
Medium
Low
5
0
−5
PC2
−20
−10
0
10
PC1

9 Metainfo

```
sessionInfo()

## R version 2.15.0 (2012-03-30)
## Platform: x86_64-unknown-linux-gnu (64-bit)
##
## locale:
## [1] C
##
## attached base packages:
## [1] grid      stats     graphics  grDevices datasets  utils     methods   base
##
## other attached packages:
##  [1] stringr_0.6       ggplot2_0.9.3     codetools_0.2-8   Cairo_1.5-1
##  [5] knitr_1.2         vimcom_0.9-8      setwidth_1.0-3    cacheSweave_0.6-1
##  [9] stashR_0.3-5      filehash_2.2-1
##
## loaded via a namespace (and not attached):
##  [1] MASS_7.3-17        RColorBrewer_1.0-5 colorspace_1.1-1   dichromat_1.2-4
##  [5] digest_0.5.2       evaluate_0.4.3     formatR_0.7        gtable_0.1.2
##  [9] labeling_0.1       munsell_0.3        plyr_1.7.1         proto_0.3-9.2
## [13] reshape2_1.2.1     scales_0.2.3       tools_2.15.0
```