PCA Check after Gem-Mapping

Ricky Lim

April 25, 2013

```
Filename: PCACheck.Rnw
Working directory: /TEMP_DDN/users/gfilion/rlim/H1_ColorChromatin/PCACheckGemMapH1
```

1 Load the input table

```
bigTable <- read.delim("input/H1_matBin.bed")
head(bigTable)
dim(bigTable)

matTable <- bigTable[, 4:162]
head(matTable)
dim(matTable)
```

2 Load the Annotation's file

```
annot <- read.delim("idH1_Fastq_annotation.txt", header = F)
dim(annot)
head(annot)[10:13]
tail(annot)[10:13]

# grep the file numbers
id_ <- sub(".*-(\\d{3}|\\d{3}[ab]).*", "\\1", annot$V11)
print(id_)
length(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$V13[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")
getProfileId("CtBP2")
```

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
dim(matTable)
```

4 matTable no NAs

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

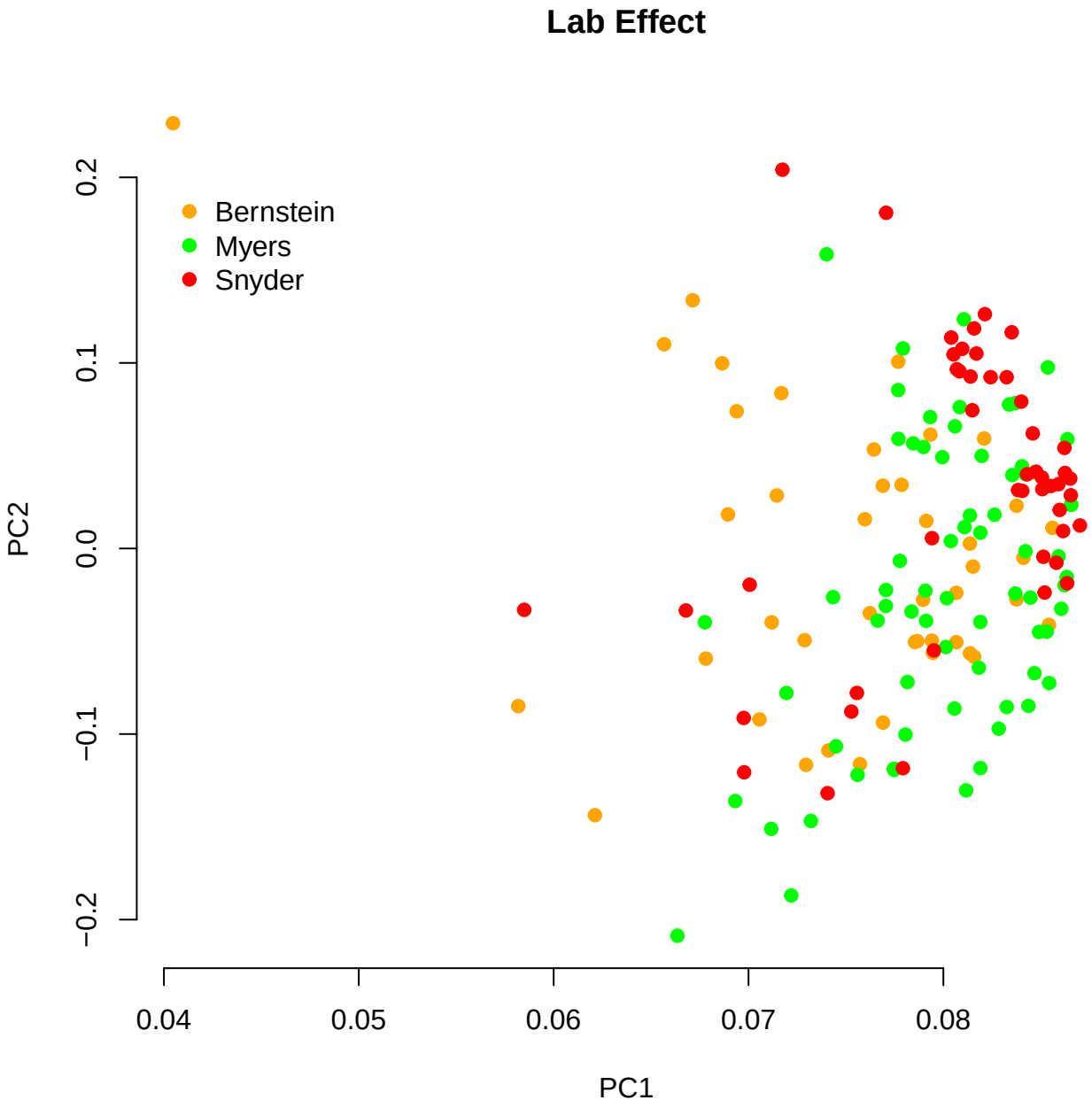
5 Create PCA Object

```
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 <- as.data.frame(lab_PCA)
head(lab_PCA)
lab_PCA[140:159, 1:5]
row.names(lab_PCA) <- rownames(lab_PCA)


# get the lab's info
```

```
lab_info <- annot$V1[annot_dataset]
lab_info

# plot lab's effect without inputs
pdf("figs/lab_effect.pdf", useDingbats = FALSE)
plot(lab_PCA[, 1], lab_PCA[, 2], col = c("orange", "green", "red")[lab_info], pch = 19, xlab = "PC1",
    ylab = "PC2", frame = F, main = "Lab Effect")
legend(y = 0.2, x = 0.04, pch = 19, cex = 1, col = c("orange", "green", "red"), legend = levels(lab_info
    box.lwd = 0, box.col = "white", bg = "white")
dev.off()

# plot lab's effect with inputs
pdf("figs/Inputslab_effect.pdf", useDingbats = FALSE)
plot(lab_PCA[, 1], lab_PCA[, 2], col = c("orange", "green", "red")[lab_info], pch = 19, xlab = "PC1",
    ylab = "PC2", frame = F, main = "Lab Effect")
All_Input <- c(getProfileId("RevXlinkChromatin"), getProfileId("Input"))
points(lab_PCA[All_Input, 1], lab_PCA[All_Input, 2], col = "blue", cex = 1.5)

points(lab_PCA[All_Input, 1], lab_PCA[All_Input, 2], col = "blue", cex = 1.5)
legend(y = 0.2, x = 0.04, pch = 19, cex = 1, col = c("orange", "green", "red"), legend = levels(lab_info
    box.lwd = 0, box.col = "white", bg = "white")

legend(y = 0.1, x = 0.04, pch = 1, cex = 1, col = c("blue"), legend = c("input"), box.lwd = 0,
    box.col = "white", bg = "white")
text(lab_PCA[All_Input, 1], lab_PCA[All_Input, 2], All_Input, pos = 1)
dev.off()
```


Lab Effect
Bernstein
Myers
Snyder
PC2
0.2
0.1
0.0
−0.1
−0.2
0.04
0.05
0.06
0.07
0.08
PC1

X111 and X109, were on top of one another, suggesting possible duplicated samples.

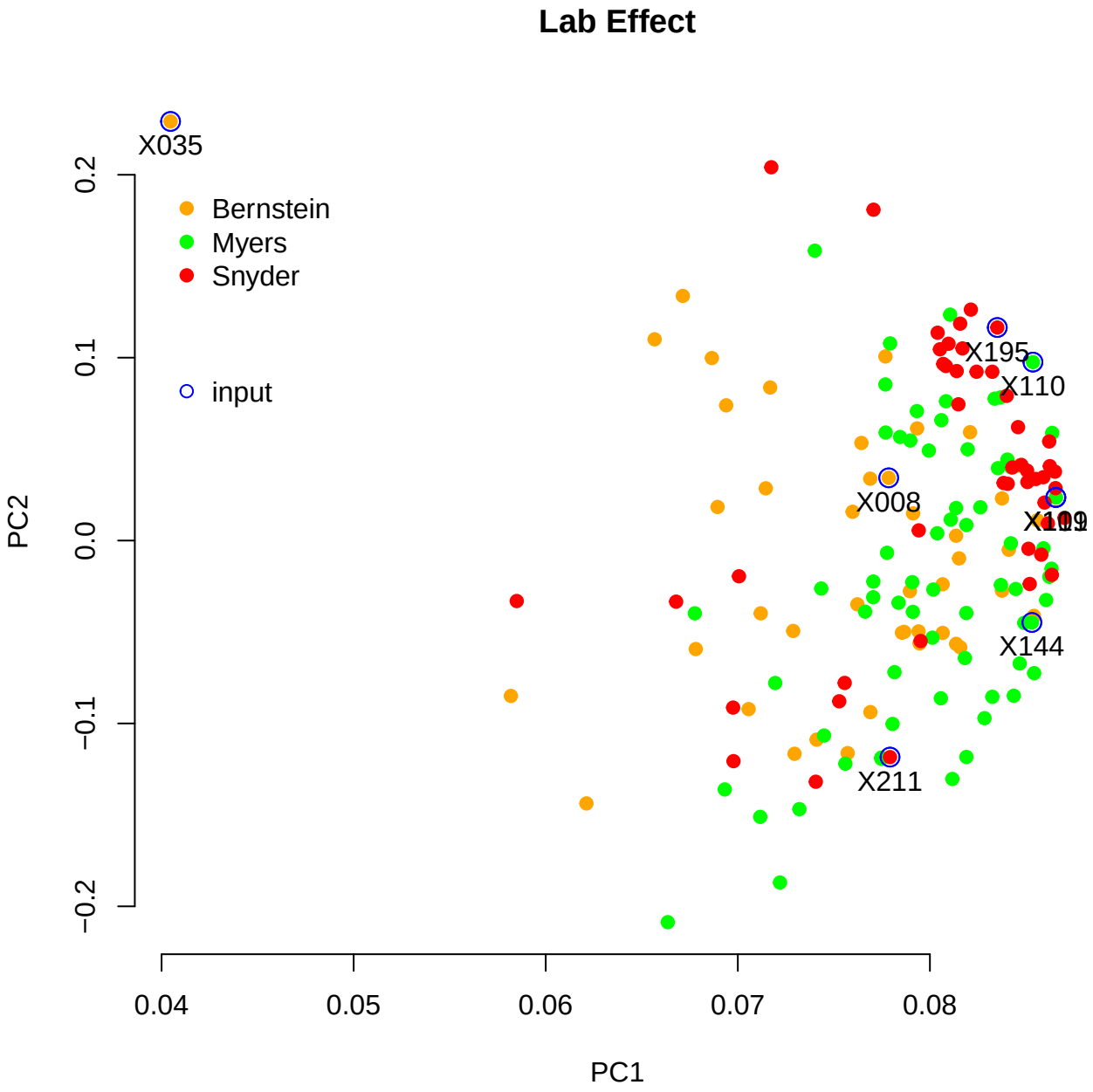
6.1 Duplicated Samples

```
cor(matTableNoNA[, "X109"], matTableNoNA[, "X111"], method = "spearman")

## [1] 1
```

X111 and X109, were duplicated. These samples that were uploaded two times in ENCODE were similar.

7 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] stats     graphics  grDevices datasets  utils     methods   base
##
## other attached packages:
## [1] Cairo_1.5-1       codetools_0.2-8   knitr_1.2         stringr_0.6
## [5] vimcom_0.9-8      setwidth_1.0-3    cacheSweave_0.6-1 stashR_0.3-5
## [9] filehash_2.2-1
##
## loaded via a namespace (and not attached):
## [1] digest_0.5.2   evaluate_0.4.3 formatR_0.7    plyr_1.7.1     tools_2.15.0
```


Lab Effect
X035
Bernstein
Myers
Snyder
input
X195
X110
X008
X144
X211
PC2
PC1
0.2
0.1
0.0
−0.1
−0.2
0.04
0.05
0.06
0.07
0.08