PCA_new

Jieqi Tu, jt3098

11/30/2019

Data Manipulation

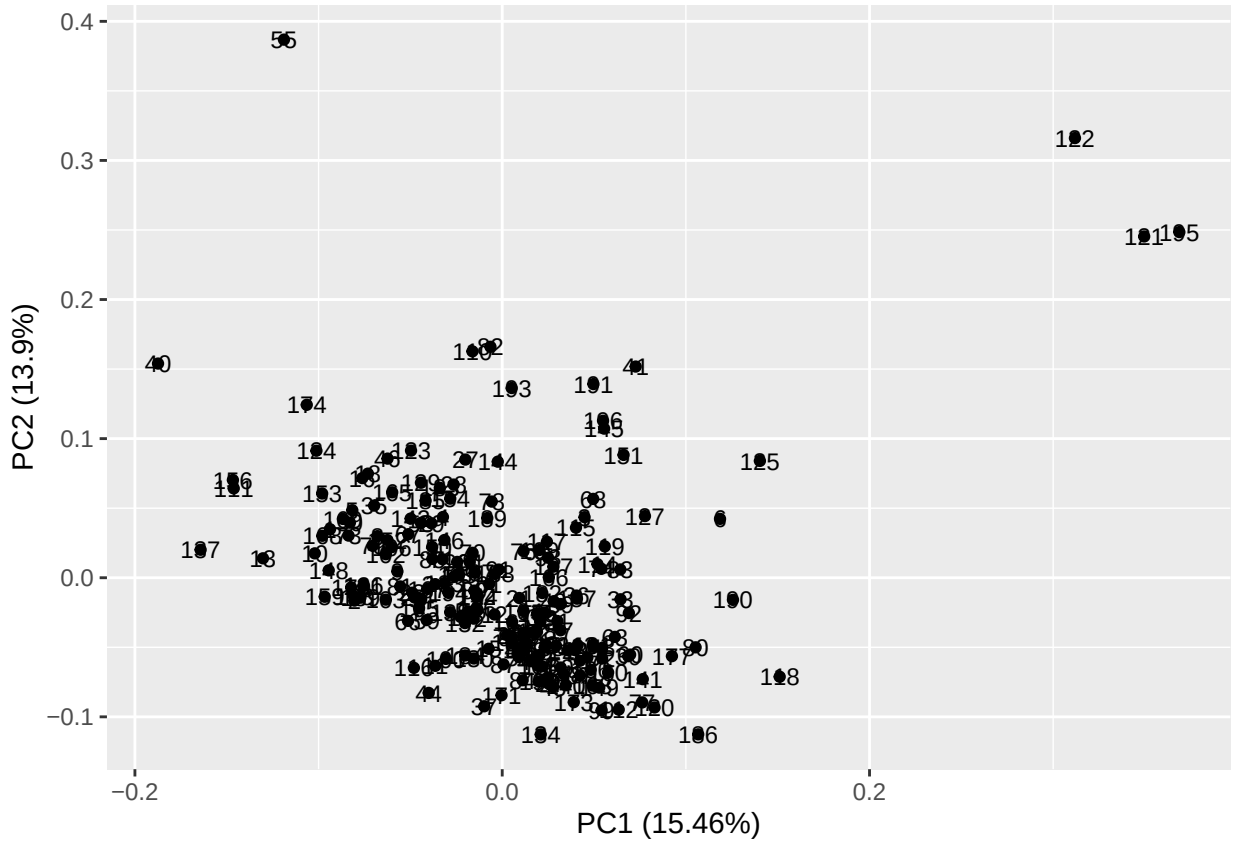
```
# Import three datasets
CL = readxl::read_excel("./ABC_Cord Blood_Metabolomics_CL data_14Nov2019.xlsx") %>% janitor::clean_names
BA = readxl::read_excel("./ABC_Cord Blood_Metabolomics_BA data_14Nov2019.xlsx") %>% janitor::clean_names
PM = readxl::read_excel("./ABC_Cord Blood_Metabolomics_PM data_14Nov2019.xlsx") %>% janitor::clean_names

# Extract only the data part of each dataset
CL_data = CL[11:491]
BA_data = BA[11:266]
PM_data = PM[11:193]
```

```
# Calculate the minimum value of each column
CL_data[CL_data == 0] = NA
BA_data[BA_data == 0] = NA
PM_data[PM_data == 0] = NA
CL_min = sapply(CL_data[1:481], function(x) min(x, na.rm = T))
BA_min = sapply(BA_data[1:256], function(x) min(x, na.rm = T))
PM_min = sapply(PM_data[1:183], function(x) min(x, na.rm = T))
CL_data = CL[11:491]
BA_data = BA[11:266]
PM_data = PM[11:193]
# Convert 0 to half of the minimum value
for(i in 1:481) {
  CL_data[i][CL_data[i]==0] = 0.5*CL_min[i]
}

for(i in 1:256) {
  BA_data[i][BA_data[i]==0] = 0.5*BA_min[i]
}

for(i in 1:183) {
  PM_data[i][PM_data[i]==0] = 0.5*PM_min[i]
}
```

PCA for CL data

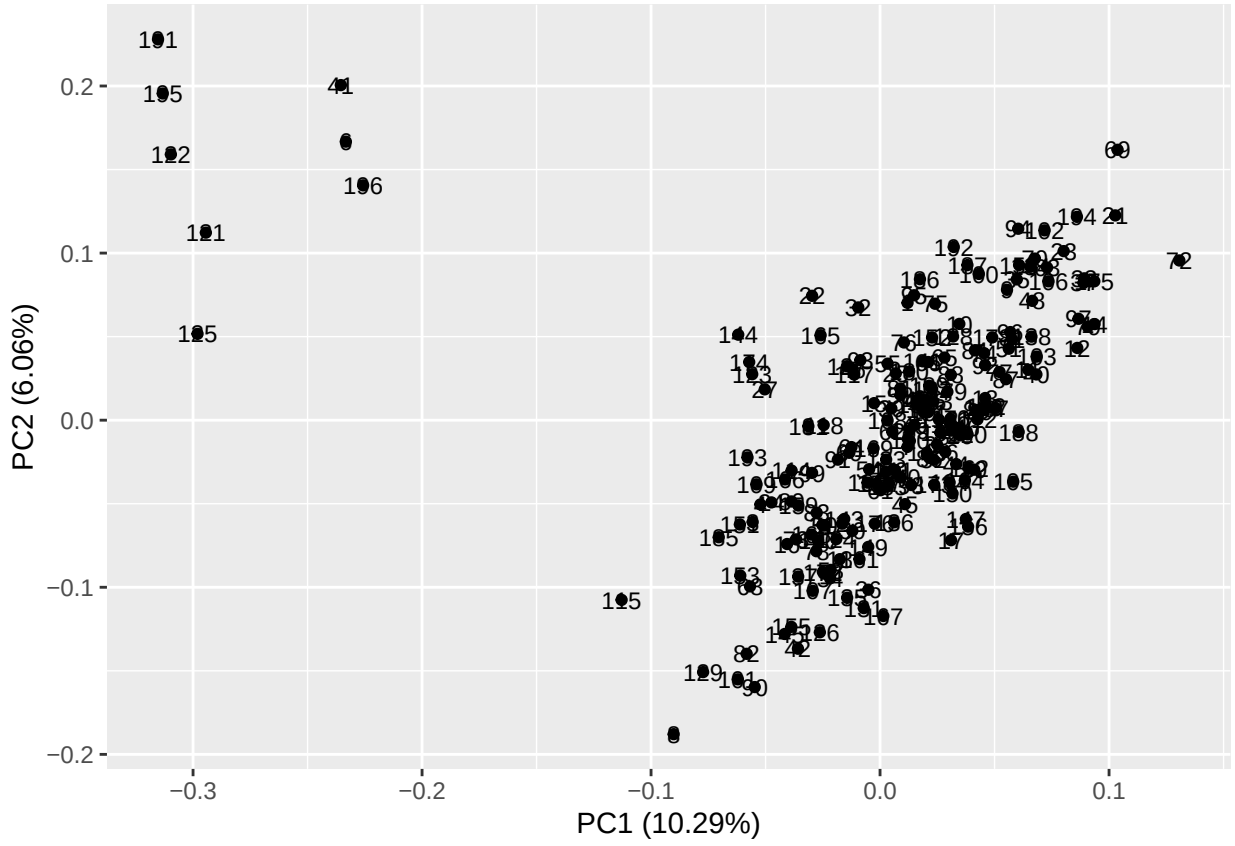
```
# Calculate z-scores
mean_CL = mean_control = sapply(CL_data[1:481], function(x) mean(x))
sd_CL = sapply(CL_data[1:481], function(x) sd(x))
CL_data_z = CL_data
for(i in 1:481) {
  CL_data_z[i] = (CL_data[i] - mean_CL[i])/sd_CL[i]
}

CL_pca = prcomp(CL_data_z[c(1:481)], center = F, scale. = F)
```

```
# Plot PC1/PC2
library(ggfortify)
autoplot(CL_pca, label = T, label.size = 3)
```



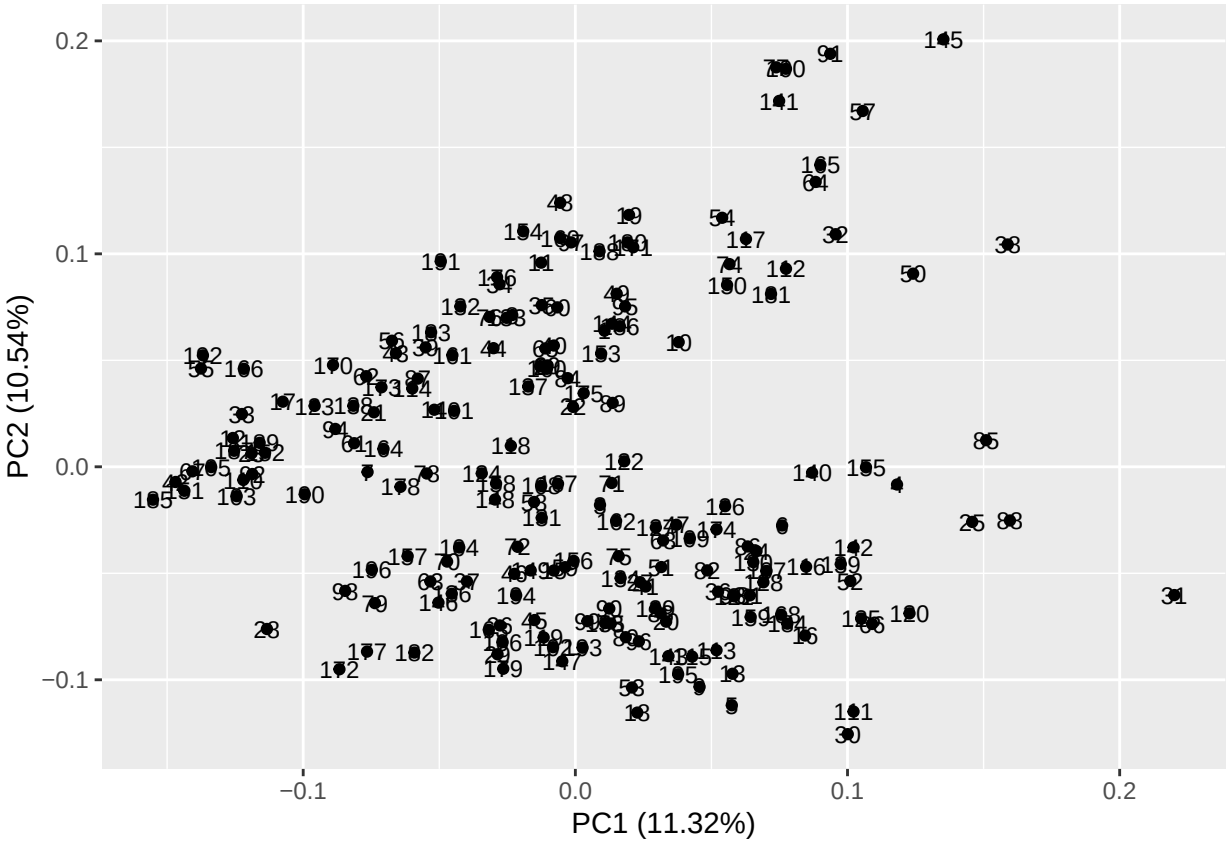
PCA for BA data

```
# Calculate z-scores
mean_BA = mean_control = sapply(BA_data[1:256], function(x) mean(x))
sd_BA = sapply(BA_data[1:256], function(x) sd(x))
BA_data_z = BA_data
for(i in 1:256) {
  BA_data_z[i] = (BA_data[i] - mean_BA[i])/sd_BA[i]
}

BA_pca = prcomp(BA_data_z[c(1:256)], center = F, scale. = F)

# Plot PC1/PC2
library(ggfortify)
autoplot(BA_pca, label = T, label.size = 3)
```



PCA for PM data

```
# Calculate z-scores
mean_PM = mean_control = sapply(PM_data[1:183], function(x) mean(x))
sd_PM = sapply(PM_data[1:183], function(x) sd(x))
PM_data_z = PM_data
for(i in 1:183) {
  PM_data_z[i] = (PM_data[i] - mean_PM[i])/sd_PM[i]
}

PM_pca = prcomp(PM_data_z[c(1:183)], center = F, scale. = F)

# Plot PC1/PC2
library(ggfortify)
autoplot(PM_pca, label = T, label.size = 3)
```


PC2 (10.54%)
PC1 (11.32%)
0.2
0.1
0.0
−0.1
−0.1
0.0
0.1
0.2