Validation of Cell-Based Quantifications via RNA-Seq Deconvolution

July 26, 2020

Read in HIF Data

```
given_date <- "12022019"
brca_date <- "02232020"
lung_date <- "10242019"
tissue_names <- c("BRCA","STAD","LUAD","LUSC","SKCM")
c(sprintf("val_data/brca_%s.rds", brca_date),
  sprintf("val_data/stad_%s.rds", given_date),
  sprintf("val_data/luad_%s.rds", lung_date),
  sprintf("val_data/lusc_%s.rds", lung_date),
  sprintf("val_data/skcm_%s.rds", given_date)
) %>%
  sprintf(given_date) %>% lapply(readRDS) %>% setNames(tissue_names) -> all_input
do.call("rbind", all_input %>%
          lapply(function(df) {
            output <- df$all$feature
            colnames(output) <- gsub(" *_HE", "", colnames(output))
            colnames(output) <- gsub(" *_FFPE", "", colnames(output))
            return(output)
          }) %>% setNames(tissue_names)) %>% as.data.frame -> all_features
metadata_counts <- sapply(all_input, function(df) {
  df$all$metadata %>% colnames
}) %>% unlist %>% table
common_metadata <- names(metadata_counts)[metadata_counts >= length(tissue_names)]
do.call("rbind",
        lapply(tissue_names, function(tissue) {
          output <- all_input[[tissue]]$all$metadata %>%
            select(all_of(common_metadata)) %>% data.frame(tissue)
        }) %>% setNames(tissue_names)) -> all_metadata
```

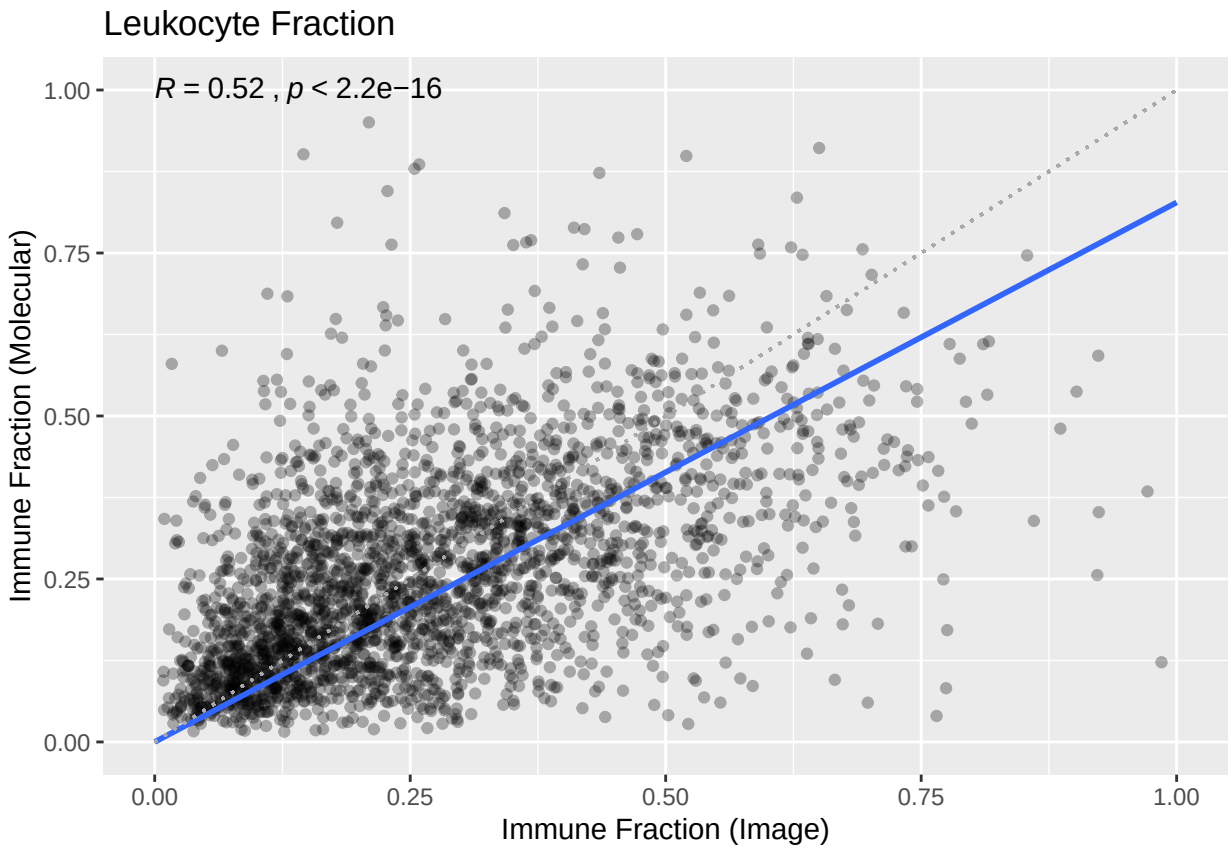
Data Processing of Feature Combinations

```
mmc2 <- openxlsx::read.xlsx("val_data/mmc2.xlsx") %>% as.data.frame()
id_list <- unique(intersect(all_metadata$case, mmc2$TCGA.Participant.Barcode))

mmc2 <- mmc2[, !duplicated(colnames(mmc2))] %>%
  filter(TCGA.Participant.Barcode %in% id_list) %>%
  mutate(case = TCGA.Participant.Barcode) %>%
  arrange(case)

all_features <- all_features %>%
```

```
  data.frame(case = all_metadata$case) %>%
  filter(case %in% id_list) %>% arrange(case)
all_metadata <- all_metadata %>%
  filter(case %in% id_list) %>% arrange(case)
mmc2 <- merge(mmc2, all_metadata, by='case')

dat <- data.frame(all_features, mmc2)
tissues <- c("STROMA","TUMOR", "EPITHELIAL","ESI_0080")
cells <- c("CANCER","FIBROBLAST","LYMPHOCYTE","PLASMA","MACROPHAGE")


tumor_cols <- colnames(dat) %in% sprintf("TOTAL.%s.CELLS.IN.TUMOR", cells)
lymphocyte_cols <- colnames(dat) %in% sprintf("TOTAL.LYMPHOCYTE.CELLS.IN.%s", tissues)
plasma_cols <- colnames(dat) %in% sprintf("TOTAL.PLASMA.CELLS.IN.%s", tissues)
macrophage_cols <- colnames(dat) %in% sprintf("TOTAL.MACROPHAGE.CELLS.IN.%s", tissues)
fibroblast_cols <- colnames(dat) %in% sprintf("TOTAL.FIBROBLAST.CELLS.IN.%s", tissues)
cancer_cols <- colnames(dat) %in% sprintf("TOTAL.CANCER.CELLS.IN.%s", tissues)

dat <- dat %>%
  mutate(LYMPHOCYTE_CELL_COUNT = dat[, lymphocyte_cols] %>% rowSums) %>%
  mutate(PLASMA_CELL_COUNT = dat[, plasma_cols] %>% rowSums) %>%
  mutate(MACROPHAGE_CELL_COUNT = dat[, macrophage_cols] %>% rowSums) %>%
  mutate(FIBROBLAST_CELL_COUNT = dat[, fibroblast_cols] %>% rowSums) %>%
  mutate(CANCER_CELL_COUNT = dat[, cancer_cols] %>% rowSums) %>%
  mutate(TUMOR_CELL_COUNT = dat[, tumor_cols] %>% rowSums) %>%
  mutate(IMMUNE_CELL_COUNT = LYMPHOCYTE_CELL_COUNT +
           PLASMA_CELL_COUNT + MACROPHAGE_CELL_COUNT) %>%
  mutate(TOTAL_CELL_COUNT = IMMUNE_CELL_COUNT +
           CANCER_CELL_COUNT + FIBROBLAST_CELL_COUNT) %>%
  mutate(IMMUNE_FRACTION = IMMUNE_CELL_COUNT/TOTAL_CELL_COUNT) %>%
  mutate(TOTAL_AREA = AREA.MM2.OF.TUMOR.IN.TISSUE + AREA.MM2.OF.TUMOR.IN.TISSUE) %>%
  mutate(STROMAL_FRACTION = AREA.MM2.OF.STROMA.IN.TISSUE/AREA.MM2.OF.TUMOR.IN.TISSUE) %>%
  mutate(TUMOR_FRACTION = AREA.MM2.OF.EPITHELIAL.IN.TISSUE/AREA.MM2.OF.TUMOR.IN.TISSUE)
```

Visualization and Significance Testing

```
cor_method <- "pearson"

#Immune fraction:
immune_fraction_comp <- dat %>%
  ggplot(aes(x=IMMUNE_FRACTION, y=Leukocyte.Fraction)) +
  geom_point(alpha = 0.3) + ggtitle("Leukocyte Fraction") +
  geom_smooth(method = "lm", se=FALSE, fullrange=TRUE, formula=y~x-1) +
  geom_segment(x = 0, y = 0, xend = 1, yend = 1, linetype="dotted", color = "darkgray") +
  xlab("Immune Fraction (Image)") + ylab("Immune Fraction (Molecular)") +
  xlim(0,1) + ylim(0,1) +
  stat_cor(method = cor_method, label.x = 0, label.y = 1)
immune_fraction_comp %>% plot
```




```
cor.test(dat$IMMUNE_FRACTION, dat$Leukocyte.Fraction, use = "pairwise.complete.obs")
```

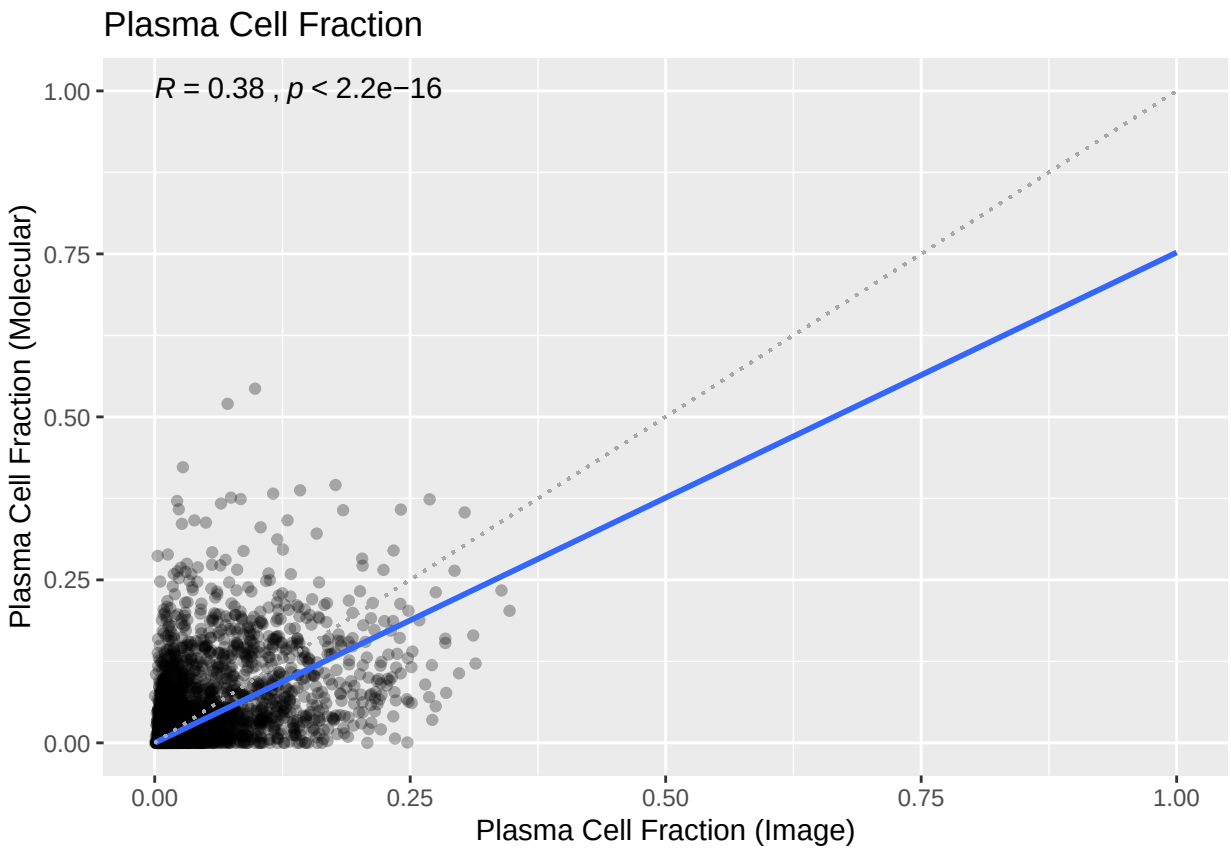
```
##
##  Pearson's product-moment correlation
##
## data:  dat$IMMUNE_FRACTION and dat$Leukocyte.Fraction
## t = 31.032, df = 2547, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4950290 0.5513975
## sample estimates:
##       cor
## 0.5237864
```

```
#Plasma Cells:
plasma_cell_comp <- dat %>%
  ggplot(aes(x=PLASMA_CELL_COUNT/TOTAL_CELL_COUNT, y=Plasma.Cells)) +
  geom_point(alpha = 0.3) + ggtitle("Plasma Cell Fraction") +
  geom_smooth(method = "lm", se=FALSE, fullrange=TRUE, formula=y~x-1) +
  geom_segment(x = 0, y = 0, xend = 1, yend = 1, linetype="dotted", color = "darkgray") +
  xlab("Plasma Cell Fraction (Image)") + ylab("Plasma Cell Fraction (Molecular)") +
  xlim(0,1) + ylim(0,1) +
  stat_cor(method = cor_method, label.x = 0, label.y = 1)
cor.test(dat$PLASMA_CELL_COUNT/dat$TOTAL_CELL_COUNT, dat$Plasma.Cells,
         use = "pairwise.complete.obs")
```

```
##
##  Pearson's product-moment correlation
```

```
##
## data:  dat$PLASMA_CELL_COUNT/dat$TOTAL_CELL_COUNT and dat$Plasma.Cells
## t = 20.459, df = 2473, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3462574 0.4136662
## sample estimates:
##       cor
## 0.3804671
```

```
plasma_cell_comp
```

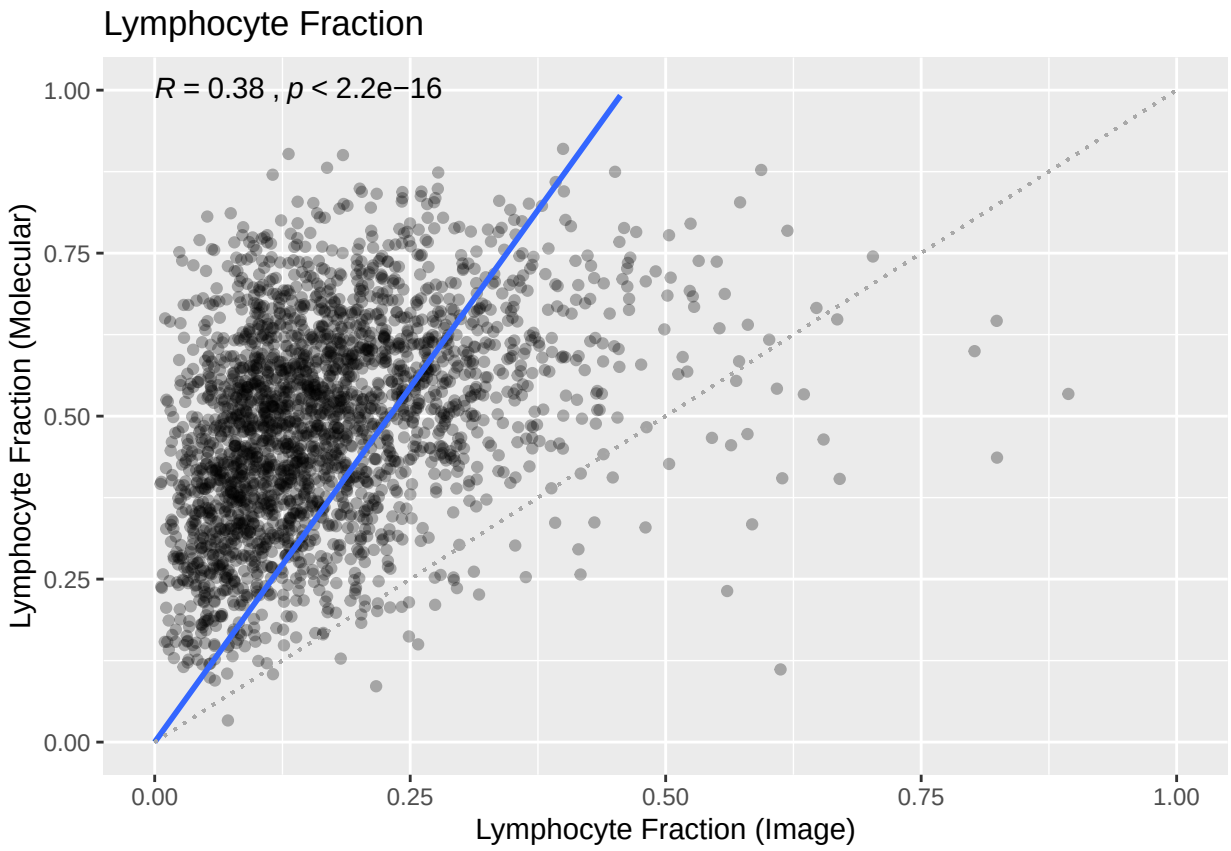



```
#Lymphocytes
lymphocyte_comp <- dat %>%
  ggplot(aes(x=LYMPHOCYTE_CELL_COUNT/TOTAL_CELL_COUNT, y=Lymphocytes)) +
  geom_point(alpha = 0.3) + ggtitle("Lymphocyte Fraction") +
  geom_smooth(method = "lm", se=FALSE, fullrange=TRUE, formula=y~x-1) +
  geom_segment(x = 0, y = 0, xend = 1, yend = 1, linetype="dotted", color = "darkgray") +
  xlab("Lymphocyte Fraction (Image)") + ylab("Lymphocyte Fraction (Molecular)") +
  xlim(0,1) + ylim(0,1) +
  stat_cor(method = cor_method, label.x = 0, label.y = 1)
cor.test(dat$LYMPHOCYTE_CELL_COUNT/dat$TOTAL_CELL_COUNT, dat$Lymphocytes,
         use = "pairwise.complete.obs")
```

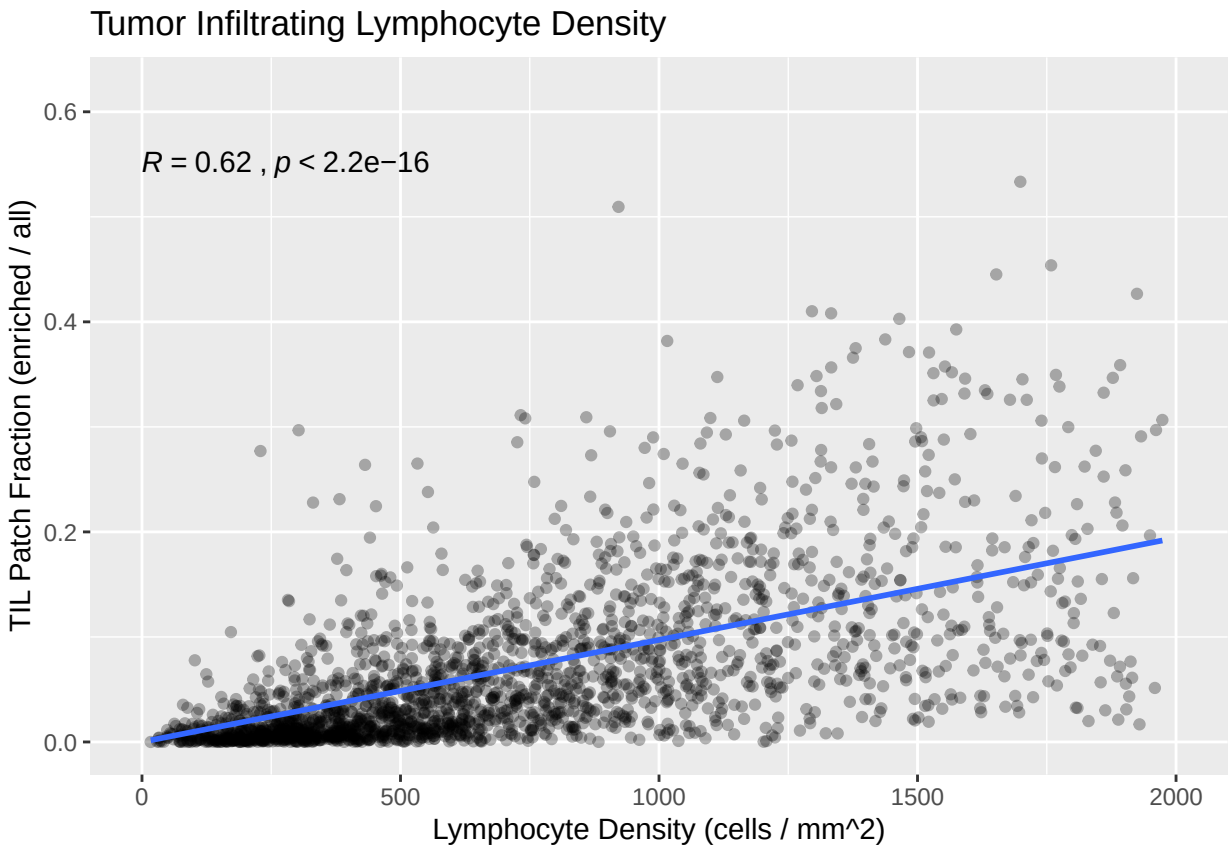
```
##
##  Pearson's product-moment correlation
##
## data:  dat$LYMPHOCYTE_CELL_COUNT/dat$TOTAL_CELL_COUNT and dat$Lymphocytes
```

```
## t = 20.548, df = 2473, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3477050 0.4150295
## sample estimates:
##       cor
## 0.3818737
```

```
lymphocyte_comp
```




```
#TIL/Lymphocyte density in tumor
TIL_lymphocyte_density_comp <- dat %>%
  ggplot(aes(x=DENSITY.LYMPHOCYTE.CELLS.IN.TUMOR, y=TIL.Regional.Fraction/100)) +
  geom_point(alpha = 0.3) + ggtitle("Tumor Infiltrating Lymphocyte Density") +
  geom_smooth(method = "lm", se=FALSE, formula=y~x-1) +
  xlim(c(0,2000)) + xlab("Lymphocyte Density (cells / mm^2)") + ylab("TIL Patch Fraction (enriched / al
  stat_cor(method = cor_method, label.x.npc = "left", label.y = 0.55)
TIL_lymphocyte_density_comp
```



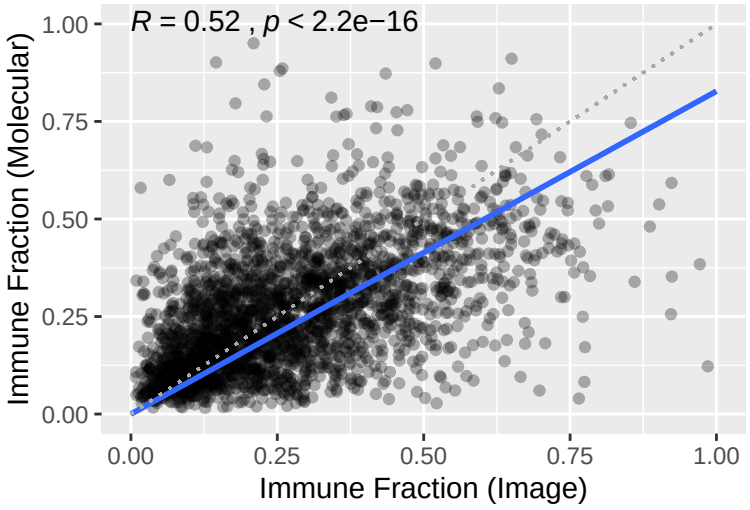

```
cor.test(dat$TIL.Regional.Fraction, dat$DENSITY.LYMPHOCYTE.CELLS.IN.TUMOR,
         use = "pairwise.complete.obs")
```

```
##
##  Pearson's product-moment correlation
##
## data:  dat$TIL.Regional.Fraction and dat$DENSITY.LYMPHOCYTE.CELLS.IN.TUMOR
## t = 33.795, df = 2320, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5464515 0.6009984
## sample estimates:
##       cor
## 0.5743622
```

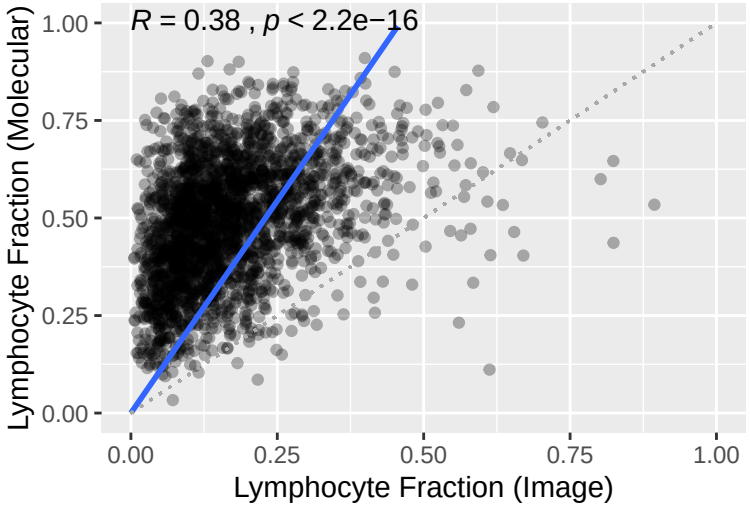
Final Output

```
plot_grid(immune_fraction_comp,
          lymphocyte_comp,
          plasma_cell_comp,
          labels = letters[1:3],
          ncol = 1, nrow = 3)
```

a Leukocyte Fraction



b Lymphocyte Fraction



c Plasma Cell Fraction