IMC Tumor cell classification report

Lijia

2025-03-25

Introduction

In this project, we use the METABRIC breast cancer IMC dataset to annotate cells in our in-house data based on predefined marker expression. We begin by categorizing METABRIC dataset cell types into tumor cells and other cells. To identify tumor-associated markers, we apply a Wilcoxon test to select markers with significantly higher expression in tumor cells. Using these markers, we then train a Random Forest model to classify tumor cells in our in-house dataset, enabling accurate and reproducible annotation. From this study, we also aim to determine the minimum set of protein markers required to characterize breast cancer tumor cells. These markers can help distinguish tumor cells across various breast cancer IMC datasets.

Basic data exploration

The data comes from a subset of the widely-known METABRIC breast cancer cohort: Imaging Mass Cytometry and Multiplatform Genomics Define the Phenogenomic Landscape of Breast Cancer. In total, we downloaded 10 samples, which include 7 major cell types, 19169 cells and 39 markers.

```r
## load data
imc.sub <- readRDS("../out/metabric_sub.rds")
## dimension of data
print(dim(imc.sub))
```

```
[1]    39 19169
```

```r
## Data normalisation
imc.sub <- normalizeCells(
  cells = imc.sub,
  markers = row.names(imc.sub),
  assayIn = "intensities",
  assayOut = "normIntensities",
  imageID = "metabricId",
  transformation = "asinh",
  method = c("trim99","minMax","PC1")
)
```

Cell types per sample

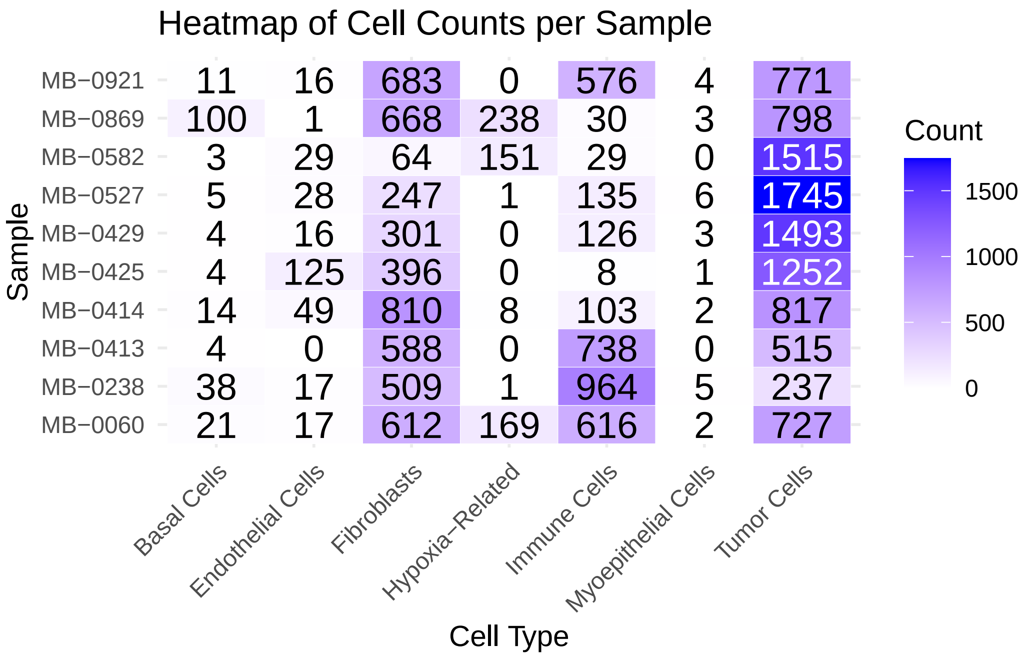
Here, we create a table to show the distribution of cell types per sample. Tumor cells constitute the majority in each sample, while other cell types are less abundant, particularly Myoepithelial Cells and Hypoxia-Related cells.

```r
# Count number of cells per sample and cell type
cell_counts <- colData(imc.sub) |>
  as.data.frame() |>
  dplyr::count(metabricId, high_level_category) |>
  pivot_wider(names_from = high_level_category,
              values_from = n, values_fill = 0)  # Fill missing values with 0

# View the transformed table by heatmap

cell_counts_long <- cell_counts |>
  pivot_longer(cols = -metabricId, names_to = "Cell_Type", values_to = "Count")

ggplot(cell_counts_long, aes(x = Cell_Type, y = metabricId, fill = Count)) +
  geom_tile(color = "white") +
  geom_text(aes(label = Count, color = Count > 1000), size = 5) +
  scale_color_manual(values = c("FALSE" = "black", "TRUE" = "white"), guide = "none") +
  scale_fill_gradient(low = "white", high = "blue") +  # Light to dark blue
  labs(title = "Heatmap of Cell Counts per Sample", x = "Cell Type", y = "Sample") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Marker list

```r
row.names(imc.sub)
```

```
 [1] "HH3_total"       "CK19"            "CK8_18"          "Twist"
 [5] "CD68"            "CK14"            "SMA"             "Vimentin"
 [9] "c_Myc"           "HER2"            "CD3"             "HH3_ph"
[13] "Erk1_2"          "Slug"            "ER"              "PR"
[17] "p53"             "CD44"            "EpCAM"           "CD45"
[21] "GATA3"           "CD20"            "Beta_catenin"    "CAIX"
[25] "E_cadherin"      "Ki67"            "EGFR"            "pS6"
[29] "Sox9"            "vWF_CD31"        "pmTOR"           "CK7"
[33] "panCK"           "c_PARP_c_Casp3"  "DNA1"            "DNA2"
[37] "H3K27me3"        "CK5"             "Fibronectin"
```

Identifying Tumor-Positive Markers

We apply a Wilcoxon test to select markers with significantly higher expression in tumor cells. With an adjusted p-value < 0.01, we identified 12 protein markers that are significantly overexpressed in tumor cells.

```r
# Extract marker intensity data
intensity_data <- assay(imc.sub, "normIntensities")
metadata <- as.data.frame(colData(imc.sub))
# Define tumor and non-tumor cells
tumor_cells <- metadata$high_level_category == "Tumor Cells"
non_tumor_cells <- !tumor_cells

# Apply Wilcoxon test for each marker
# return the Pvalue, and mean fold change
wilcox_results <- apply(intensity_data, 1, function(marker) {
  test <- wilcox.test(marker[tumor_cells], marker[non_tumor_cells], alternative = "greater")
  return(data.frame(p_value = test$p.value,
                    FC_tumor_other = mean(marker[tumor_cells])/mean(marker[non_tumor_cells]))
})

## Create data frame
wilcox_df <- do.call(rbind,wilcox_results) |> as.data.frame() |>
  mutate(adj_p_value = p.adjust(p_value, method = "BH")) |>  # Adjust for multiple testing
  rownames_to_column(var = "Marker")

significant_markers <- wilcox_df |> filter(adj_p_value < 0.01)
tumor_markers <- significant_markers$Marker
print(significant_markers[order(significant_markers$adj_p_value),])
```

```
       Marker       p_value FC_tumor_other   adj_p_value
5       GATA3  0.000000e+00      1.4472834  0.000000e+00
11      panCK  0.000000e+00      1.3523464  0.000000e+00
1      CK8_18 9.112173e-291      1.4163716 1.184582e-289
3          PR 6.669770e-137      1.1055615 6.503026e-136
8        Ki67  3.130944e-66      2.1494561  2.442137e-65
7  E_cadherin  3.921108e-57      1.0392327  2.548720e-56
2        HER2  3.064716e-33      1.0682032  1.707485e-32
10        CK7  1.329599e-19      1.1367598  6.481794e-19
9        Sox9  1.666557e-19      1.0838761  7.221747e-19
4         p53  7.652464e-10      1.0429684  2.984461e-09
6        CD20  6.066256e-04      0.9917175  2.028140e-03
12        CK5  6.240430e-04      1.0278901  2.028140e-03
```

Predicting Tumor Cells in In-House Breast Cancer IMC Data

Next, we aim to evaluate whether the 12 protein markers have predictive power to distinguish tumor cells from other cells.

We trained a Random Forest model using the METABRIC dataset and applied it to predict tumor cells in our in-house breast cancer IMC samples. Since we previously manually annotated the in-house samples, we can assess the model's prediction accuracy using these 12 markers.

We combined Basal Cells, Endothelial Cells, Fibroblasts, Hypoxia-Related, Immune Cells, and Myoepithelial Cells into a single "other cells" category and developed a binary classification model to distinguish tumor cells from other cells.

```r
imc.train <- imc.sub[tumor_markers,]

train_data <- as.data.frame(t(assay(imc.train, "normIntensities"))) |>
  mutate(cell_type = ifelse(colData(imc.train)$high_level_category == "Tumor Cells", "Tumor"


train_data$cell_type <- as.factor(train_data$cell_type)

X_train <- train_data |> select(-cell_type)

Y_train <- train_data$cell_type


set.seed(123)
cv_control <- trainControl(method = "cv", number = 5, savePredictions = "final")

# Define tuning grid (optional)
tune_grid <- expand.grid(mtry = sqrt(ncol(X_train)))  # Default sqrt(p) for RF

###   Train the Random Forest Model with CV
rf_model_cv <- train(
  x = X_train,
  y = Y_train,
  method = "rf",  # Random Forest
  trControl = cv_control,
  tuneGrid = tune_grid,
  ntree = 500  # Number of trees
)

# Print CV Results
print(rf_model_cv)
```

```
Random Forest

19169 samples
   12 predictor
    2 classes: 'Other', 'Tumor'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 15335, 15335, 15335, 15336, 15335
Resampling results:

  Accuracy   Kappa
  0.8935781  0.7867343

Tuning parameter 'mtry' was held constant at a value of 3.464102
```

```r
### Extract the Best Model
best_rf_model <- rf_model_cv$finalModel
```

Feature importance score

We show the feature importance scores from the Random Forest model. `Ki67`, `GATA3`, `CK8_18`, `panCK`, `PR` are the top 5 most important protein markers for identifying breast cancer tumor cells.

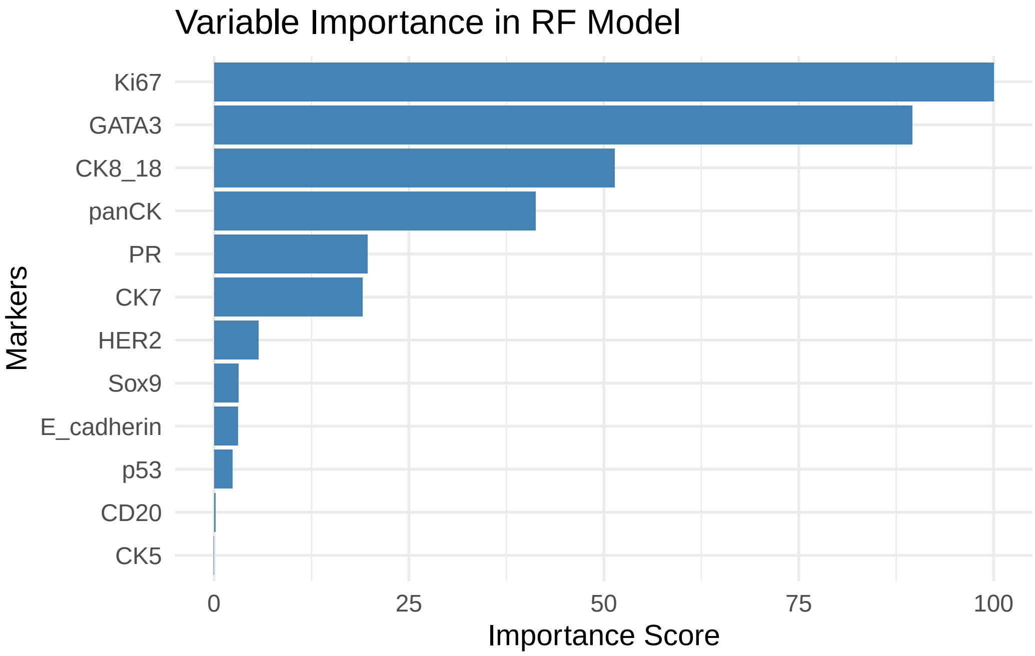
```r
# Extract variable importance
var_importance <- varImp(rf_model_cv)

# Convert to a tidy data frame
importance_df <- as.data.frame(var_importance$importance)

# Add marker names as a column
importance_df$Marker <- rownames(importance_df)

# Sort by importance (descending)
importance_df <- importance_df |> arrange(desc(Overall))


ggplot(importance_df, aes(x = reorder(Marker, Overall), y = Overall)) +
  geom_bar(stat = "identity", fill = "steelblue") +  # Bar plot
  coord_flip() +  # Flip axes for better readability
  labs(title = "Variable Importance in RF Model",
       x = "Markers",
       y = "Importance Score") +
  theme_minimal()
```



Predicting Tumor Cells

`imc.test` is our in-house data, it also includes 10 samples.

```
## read test data
imc.test <- readRDS("../out/imc_test.rds")

## data normalisation
imc.test <- normalizeCells(
  cells = imc.test,
  markers = row.names(imc.test),
  assayIn = "intensities",
  assayOut = "normIntensities",
  imageID = "metabricId",
  transformation = "asinh",
  method = c("trim99","minMax","PC1")
)


test_data <- as.data.frame(t(assay(imc.test[tumor_markers,], "normIntensities"))) |>
  mutate(cell_type = ifelse(colData(imc.test)$high_level_category == "Tumor Cells", "Tumor",

X_test <- test_data |> select(-cell_type)
Y_test <- test_data$cell_type  # True labels

test_predictions <- predict(best_rf_model, X_test)
```

Confusion matrix

The prediction results show an overall accuracy of 0.8151, with a sensitivity of 0.5955 and a specificity of 0.9272.

```
conf_matrix <- confusionMatrix(test_predictions, as.factor(Y_test))

print(conf_matrix)
```

```
Confusion Matrix and Statistics

          Reference
Prediction Other Tumor
     Other  3786   908
     Tumor  2572 11556

               Accuracy : 0.8151
                 95% CI : (0.8095, 0.8206)
    No Information Rate : 0.6622
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5584

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.5955
            Specificity : 0.9272
         Pos Pred Value : 0.8066
         Neg Pred Value : 0.8180
```

```
           Prevalence : 0.3378
       Detection Rate : 0.2011
 Detection Prevalence : 0.2494
    Balanced Accuracy : 0.7613

     'Positive' Class : Other
```