QC

Siwanart Ma

2024-12-21

Initial setting up

import the cleaned data which NA values and duplicates are removed

```
df = read.csv("/scratch_tmp/grp/msc_appbio/DCDM_group6/outputs/tables/merged_data_clean.csv")
print(head(df))
```

```
##       analysis_id gene_accession_id gene_symbol mouse_strain  mouse_life_stage
## 1 0003b50qbwu5484       MGI:2444120       Spag4        C57BL       Early adult
## 2 0003klt8412603i       MGI:2385619      Klhl12        C57BL Middle aged adult
## 3 00052qirbv4o3tm       MGI:1098826        Coq4        C57BL       Early adult
## 4 00062gd65s37385       MGI:1917237      Spaca9        C57BL       Early adult
## 5 0007p3a8djq350z       MGI:2448580       Myo3b        C57BL       Early adult
## 6 0007psb08ytc999       MGI:2155705      P2ry14        C57BL        Late adult
##         parameter_id
## 1  ESLIM_005_001_702
## 2 KMPCLA_CSD_008_001
## 3  M-G-P_006_001_026
## 4   IMPC_IMM_033_002
## 5  M-G-P_016_001_004
## 6   IMPC_GEO_009_002
##                                                parameter_name     pvalue
## 1                                          Lean/Body weight 0.70844160
## 2                                 Coat - color pattern - back 0.42109680
## 3                                          Fusion of vertebrae 0.08293449
## 4 NaÃ\u0083Â¯ve CD8+ T cells - % of live leukocytes (Panel A) 0.30371250
## 5                                             Haematocrit 0.17170630
## 6                                              Hemorrhage 0.85469090
```

Import SOP Read sop

```
sop = read.csv("/scratch_tmp/grp/msc_appbio/DCDM_group6/metadata/IMPC_SOP.csv")
sop
```

```
##           dataField dataType minValue maxValue
## 1       analysis_id   String       15       15
## 2 gene_accession_id   String        9       11
## 3       gene_symbol   String        1       13
## 4      mouse_strain   String        3        5
## 5  mouse_life_stage   String        4       17
## 6      parameter_id   String       15       18
## 7    parameter_name   String        2       74
## 8            pvalue    Float        0        1
##                                   exampleValues
## 1                                 0a1yei0604s8275
```

```
## 2                                           MGI:2679336
## 3                                                 Satb2
## 4                                                 C57BL
## 5                                                 E15.5
## 6                                      IMPC_GRS_009_001
## 7 Forelimb and hindlimb grip strength measurement mean
## 8                                             0.8004226
##
## 1                                                                        Unique 15 character string containing al
## 2 Unique alphanumeric character string that is used to unambiguously identify a particular record in
## 3                                                                                                    Offic
## 4                                                                                       Name of mouse stra
## 5            Developmental stage of mice being analysed. Values are E12.5; E15.5; E18.5; E9.5; Early
## 6
## 7
## 8
```

Set the ranges

```
allFields = sop[,1]
allMins = sop[,3]
allMaxs = sop[,4]
print(allFields)
```

```
## [1] "analysis_id"       "gene_accession_id" "gene_symbol"
## [4] "mouse_strain"      "mouse_life_stage"  "parameter_id"
## [7] "parameter_name"    "pvalue"
```

```
print(allMins)
```

```
## [1] 15  9  1  3  4 15  2  0
```

```
print(allMaxs)
```

```
## [1] 15 11 13  5 17 18 74  1
```

QC all data

```
# Load the progress bar library
library(progress)

# Initialize counters for overall QC failures
overall_string_failure_counter <- 0
overall_pvalue_failure_counter <- 0

# Initialize a column to count the number of QC failures per row
df$qc_failure_count <- 0
df$qc_pvalue_failure_count <- 0

# Loop through all fields in SOP
for (i in 1:length(allFields)) {
  # Get the current field name
  field_name <- as.character(allFields[i])

  # Create a progress bar for the rows in the merged dataframe
  pb <- progress_bar$new(total = nrow(df), format = paste0("Checking ", field_name, " [:bar] :percent :e

  # Initialize counters for QC failures for this field
```

```
string_failure_counter <- 0
pvalue_failure_counter <- 0

# Check if the field's data type is "String"
if (sop$dataType[i] == "String") {
  # Loop through rows of the merged dataframe
  for (row in 1:nrow(df)) {
    # Extract the current row
    row_data <- df[row, ]

    # Fetch the analysis ID (assuming column name is `analysis_id`)
    analysis_id <- if ("analysis_id" %in% colnames(df)) as.character(row_data$analysis_id) else "Unkn

    # Perform QC check for the current string field
    if (field_name %in% colnames(df)) {
      field_value <- as.character(row_data[[field_name]])
      field_length <- nchar(field_value)

      if (is.na(field_value) || field_length < allMins[i] || field_length > allMaxs[i]) {
        print(paste("Row", row, "with Analysis ID", analysis_id,
                    "has", field_name, "value", field_value,
                    "with", field_length, "characters and failed QC"))
        cat("\n")
        string_failure_counter <- string_failure_counter + 1
        df$qc_failure_count[row] <- df$qc_failure_count[row] + 1
      }
    }
    # Update the progress bar
    pb$tick()
  }
  # Print summary for this string field
  print(paste("Field:", field_name, "- Total String QC failures:", string_failure_counter))
  overall_string_failure_counter <- overall_string_failure_counter + string_failure_counter
} else {
  # Loop through rows for non-string fields to check p-value
  for (row in 1:nrow(df)) {
    # Extract the current row
    row_data <- df[row, ]

    # Fetch the analysis ID (assuming column name is `analysis_id`)
    analysis_id <- if ("analysis_id" %in% colnames(df)) as.character(row_data$analysis_id) else "Unkn

    # Perform QC check for numeric p-value fields
    if (field_name %in% colnames(df)) {
      field_value <- as.numeric(row_data[[field_name]])

      if (is.na(field_value) || (!is.na(field_value) && (field_value < allMins[i] || field_value > al
        #print(paste("Row", row, "with Analysis ID", analysis_id,
                    #"has", field_name, field_value, "and failed QC for p-value range"))
        df$qc_pvalue_failure_count[row] <- 1

        pvalue_failure_counter <- pvalue_failure_counter + 1
      }
```

```
      }
      # Update the progress bar
      pb$tick()
    }
    # Print summary for this numeric field
    print(paste("Field:", field_name, "- Total p-value QC failures:", pvalue_failure_counter))
    overall_pvalue_failure_counter <- overall_pvalue_failure_counter + pvalue_failure_counter
  }
}
```

```
## [1] "Field: analysis_id - Total String QC failures: 0"
## [1] "Field: gene_accession_id - Total String QC failures: 0"
## [1] "Field: gene_symbol - Total String QC failures: 0"
## [1] "Field: mouse_strain - Total String QC failures: 0"
## [1] "Field: mouse_life_stage - Total String QC failures: 0"
## [1] "Field: parameter_id - Total String QC failures: 0"
## [1] "Field: parameter_name - Total String QC failures: 0"
## [1] "Field: pvalue - Total p-value QC failures: 1707"
```

```
# Print overall summary of QC results
cat("\n") # Adds a blank line
```

```
print(paste("Total String QC failures across all fields:", overall_string_failure_counter))
```

```
## [1] "Total String QC failures across all fields: 0"
```

```
print(paste("Total p-value QC failures across all fields:", overall_pvalue_failure_counter))
```

```
## [1] "Total p-value QC failures across all fields: 1707"
```

```
print(paste("Total rows processed:", nrow(df)))
```

```
## [1] "Total rows processed: 170617"
```

```
cat("\n")
```

Data that failed QC

```
df_failQC = df[df$qc_pvalue_failure_count==1,]
print(head(df_failQC))
```

```
##         analysis_id gene_accession_id gene_symbol mouse_strain mouse_life_stage
## 16  000o7ub5mtl2j33        MGI:104742       Kcnj3        C57BL      Early adult
## 29  0016341kad903gb       MGI:1351627        Pdhx        C57BL      Early adult
## 62  0026n3bpem35653       MGI:1201792       Pde1a        C57BL      Early adult
## 81  0033u0t6k567llr       MGI:1861453      Actl6a        C57BL             E9.5
## 131 0050e1nh1kt4l03       MGI:2183102       Sardh        C57BL      Early adult
## 139 0055k7w4a7v6q4k         MGI:88294     Cacna1s        C53BL      Early adult
##          parameter_id                           parameter_name   pvalue
## 16  JAXLA_DXA_008_001                        Lean/Body weight 1.451129
## 29   IMPC_HEM_033_001             Monocyte differential count 1.293499
## 62   IMPC_XRY_012_001                                   Pelvis 1.214774
## 81  ESLIM_007_001_008               Periphery permanence time 1.074841
## 131  IMPC_GRS_008_001 Forelimb grip strength measurement mean 1.093216
## 139  IMPC_GEP_005_002           Responsive to tactile stimuli 1.346702
##     qc_failure_count qc_pvalue_failure_count
## 16                 0                      1
## 29                 0                      1
```

```
## 62                 0                       1
## 81                 0                       1
## 131                0                       1
## 139                0                       1
```

Data that passed QC

```
df_passQC = df[df$qc_pvalue_failure_count==0,]
print(head(df_passQC))
```

```
##       analysis_id gene_accession_id gene_symbol mouse_strain  mouse_life_stage
## 1 0003b50qbwu5484       MGI:2444120       Spag4        C57BL       Early adult
## 2 0003klt8412603i       MGI:2385619      Klhl12        C57BL Middle aged adult
## 3 00052qirbv4o3tm       MGI:1098826        Coq4        C57BL       Early adult
## 4 00062gd65s37385       MGI:1917237      Spaca9        C57BL       Early adult
## 5 0007p3a8djq350z       MGI:2448580       Myo3b        C57BL       Early adult
## 6 0007psb08ytc999       MGI:2155705      P2ry14        C57BL        Late adult
##         parameter_id
## 1  ESLIM_005_001_702
## 2 KMPCLA_CSD_008_001
## 3  M-G-P_006_001_026
## 4   IMPC_IMM_033_002
## 5  M-G-P_016_001_004
## 6   IMPC_GEO_009_002
##                                             parameter_name     pvalue
## 1                                         Lean/Body weight 0.70844160
## 2                                Coat - color pattern - back 0.42109680
## 3                                        Fusion of vertebrae 0.08293449
## 4 NaÃ\u0083Â¯ve CD8+ T cells - % of live leukocytes (Panel A) 0.30371250
## 5                                                Haematocrit 0.17170630
## 6                                                 Hemorrhage 0.85469090
##   qc_failure_count qc_pvalue_failure_count
## 1                0                       0
## 2                0                       0
## 3                0                       0
## 4                0                       0
## 5                0                       0
## 6                0                       0
```

Export the data

```
write.csv(df_failQC, "/scratch_tmp/grp/msc_appbio/DCDM_group6/outputs/tables/failQC.csv", row.names = F
write.csv(df_passQC, "/scratch_tmp/grp/msc_appbio/DCDM_group6/outputs/tables/passQC.csv", row.names = F
```