01_TNBC_Diagnostics

2022-11-18

Contents

Overview

The document will perform the following diagnostics:

1. Check the unique number and values of the metadata, and constuct a tibble of metadata for each patient
2. Checking the basic properties such as dimensions and column names of the dataset
3. Check if `sample_id` and `patient_id` have the same length, and check if `sample_id` and `ImageNb` have the same values
4. Check if one value in one column is unique to one value in the other two columns
5. Check if `cell_type` and `mm` have the same length, and if true then check if one `cell_type` is unique to one `mm` value
6. Check the names of the `mm` cell type and construct a tibble for the count of different cell types present in each sample
7. Find number of unique cluster number and investigate number of different clusters in each sample with graphs
8. Investigate the range, median and standard deviation of the all spatial coordinates, for each cell type and for each cell type in each sample

Libraries

```
library(readr)
library(dplyr)
library(ggplot2)
library(formattable)
```

Read CSV

Read in the MATLAB revised TNBC output CSV files.

```
TNBC <- readr::read_csv("/Users/henzhwang/Desktop/TNBC_training/MIBI-TNBC_scdata_counts_mm_matlab_revis
```

```
## Rows: 179194 Columns: 64
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): SITE_02, RECURRENCE_LABEL, mm
## dbl (61): sample_id, patient_id, AGE_AT_DX, STAGE, LATERAL, GRADE, Survival_...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Dataset Diagnostics

The aim of this rmarkdown file is to perform disgnostics on the `TNBC` dataset.

Basic Properties

There are 179194 rows and 64 columns in the `TNBC` dataset, 44 of the columns are proteins markers and 8 coulmns are the patients metadata.

```
# dimensions
dim(TNBC)
```

```
## [1] 179194     64
```

```
# names of the columns
names(TNBC)
```

```
##  [1] "sample_id"           "patient_id"          "AGE_AT_DX"
##  [4] "STAGE"               "SITE_02"             "LATERAL"
##  [7] "GRADE"               "RECURRENCE_LABEL"    "Survival_days_capped"
## [10] "cluster_id"          "mm"                  "cell_type"
## [13] "ImageNb"             "cellLabelInImage"    "cellSize"
## [16] "cellRadius"          "centroidX"           "centroidY"
## [19] "majoraxis"           "eccentricity"        "Au"
## [22] "Background"          "betaCatenin"         "Ca"
## [25] "CD11b"               "CD11c"               "CD138"
## [28] "CD16"                "CD20"                "CD209"
## [31] "CD3"                 "CD31"                "CD4"
## [34] "CD45"                "CD45RO"              "CD56"
## [37] "CD63"                "CD68"                "CD8"
## [40] "dsDNA"               "EGFR"                "Fe"
## [43] "FoxP3"               "H3K27me3"            "H3K9ac"
## [46] "HLA_Class_1"         "HLADR"               "IDO"
## [49] "Keratin17"           "Keratin6"            "Ki67"
## [52] "Lag3"                "MPO"                 "Na"
## [55] "P"                   "p53"                 "panKeratin"
## [58] "PD1"                 "PDL1"                "pS6"
## [61] "Si"                  "SMA"                 "Ta"
## [64] "Vimentin"
```

Metadata

There are 8 patient metadata in the dataset, they are `patient_id`, `AGE_AT_DX`, `STAGE`, `SITE_02`, `LATERAL`, `GRADE`, `RECURRENCE_LABEL`, `Survival_days_capped`. We want to investigate on their unique number and metadata information for each patient.

```
# unique number for each metadata
data.frame(patient_id = length(unique(TNBC$patient_id)),
           AGE_AT_DX = length(unique(TNBC$AGE_AT_DX)),
           STAGE = length(unique(TNBC$STAGE)),
           SITE_02 = length(unique(TNBC$SITE_02)),
           LATERAL = length(unique(TNBC$LATERAL)),
           GRADE = length(unique(TNBC$GRADE)),
           RECURRENCE_LABEL = length(unique(TNBC$RECURRENCE_LABEL)),
           Survival_days_capped = length(unique(TNBC$Survival_days_capped)))
```

```
##   patient_id AGE_AT_DX STAGE SITE_02 LATERAL GRADE RECURRENCE_LABEL
## 1         39        28     8       7       2     5                2
##   Survival_days_capped
## 1                   38
```

```
# unique values
# patient_id
print("patien_id")
```

```
## [1] "patien_id"
```

```
data.frame(patient_id = unique(TNBC$patient_id)) %>% t()
```

```
##             [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10] [,11]
## patient_id 30824 30805 30812 30838 30865 30847 30846 30783 30781 30782 30753
##            [,12] [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22]
## patient_id 30770 30766 30744 30742 30734 30786 30739 30762 30785 30789 30843
##            [,23] [,24] [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33]
## patient_id 30868 30866 30821 30827 30818 30823 30851 30853 30799 30854 30732
##            [,34] [,35] [,36] [,37] [,38] [,39]
## patient_id 30771 30738 30765 30860 30740 30754
```

```
# AGE_AT_DX
print("AGE_AT_DX")
```

```
## [1] "AGE_AT_DX"
```

```
data.frame(AGE_AT_DX = unique(TNBC$AGE_AT_DX)) %>% t()
```

```
##           [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## AGE_AT_DX   77   67   42   41   64   53   26   79   60    38    31    37    39
##           [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## AGE_AT_DX    45    50    35    52    68    36    49    43    75    63    54
##           [,25] [,26] [,27] [,28]
## AGE_AT_DX    59    62    91    48
```

```r
# STAGE
print("STAGE")
```

```
## [1] "STAGE"
```

```r
data.frame(STAGE = unique(TNBC$STAGE)) %>% t()
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## STAGE   33   32   21   22   11   10   31   40
```

```r
# SITE_02
print("SITE_02")
```

```
## [1] "SITE_02"
```

```r
data.frame(SITE_02 = unique(TNBC$SITE_02)) %>% t()
```

```
##         [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]
## SITE_02 "C504" "C509" "C505" "C508" "C503" "C502" "C501"
```

```r
# LATERAL
print("LATERAL")
```

```
## [1] "LATERAL"
```

```r
data.frame(LATERAL = unique(TNBC$LATERAL)) %>% t()
```

```
##         [,1] [,2]
## LATERAL    2    1
```

```r
# GRADE
print("GRADE")
```

```
## [1] "GRADE"
```

```r
data.frame(GRADE = unique(TNBC$GRADE)) %>% t()
```

```
##       [,1] [,2] [,3] [,4] [,5]
## GRADE    1    3    9    2    4
```

```r
# RECURRENCE_LABEL
print("RECURRENCE_LABEL")
```

```
## [1] "RECURRENCE_LABEL"
```

```
data.frame(RECURRENCE_LABEL = unique(TNBC$RECURRENCE_LABEL)) %>% t()
```

```
##                   [,1]       [,2]
## RECURRENCE_LABEL "POSITIVE" "NEGATIVE"
```

```
# Survival_days_capped
print("Survival_days_capped")
```

```
## [1] "Survival_days_capped"
```

```
data.frame(Survival_days_capped = unique(TNBC$Survival_days_capped)) %>% t()
```

```
##                       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
## Survival_days_capped 2612  745 3130 2523 1683 2275  946 3767 3822  3774  4353
##                       [,12] [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20]
## Survival_days_capped  1072  4145   530  2842  5063  3725  4761    91  1319
##                       [,21] [,22] [,23] [,24] [,25] [,26] [,27] [,28] [,29]
## Survival_days_capped  2438  1568  1738  2832  2759  3063  2853  2096  3573
##                       [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37] [,38]
## Survival_days_capped  3355   584   635   194  4785  3658  1009  1754  4430
```

```
# metadata information for each sample_id
TNBC %>%
  group_by(sample_id) %>%
  summarise(n_1 = length(unique(AGE_AT_DX)),
            AGE_AT_DX = unique(AGE_AT_DX),
            n_2 = length(unique(STAGE)),
            STAGE = unique(STAGE),
            n_3 = length(unique(SITE_02)),
            SITE_02 = unique(SITE_02),
            n_4 = length(unique(LATERAL)),
            LATERAL = unique(LATERAL),
            n_5 = length(unique(GRADE)),
            GRADE = unique(GRADE),
            n_6 = length(unique(RECURRENCE_LABEL)),
            RECURRENCE_LABEL = unique(RECURRENCE_LABEL),
            n_7 = length(unique(Survival_days_capped)),
            Survival_days_capped = unique(Survival_days_capped))
```

```
## # A tibble: 39 x 15
##    sample_id   n_1 AGE_AT_DX   n_2 STAGE   n_3 SITE_02   n_4 LATERAL   n_5 GRADE
##        <dbl> <int>     <dbl> <int> <dbl> <int> <chr>    <int>   <dbl> <int> <dbl>
##  1         1     1        77     1    33     1 C504         1       2     1     1
##  2         2     1        67     1    32     1 C509         1       2     1     3
##  3         3     1        42     1    21     1 C509         1       2     1     3
##  4         4     1        41     1    22     1 C505         1       2     1     3
##  5         5     1        64     1    11     1 C508         1       1     1     3
##  6         6     1        53     1    10     1 C508         1       1     1     3
##  7         7     1        62     1    32     1 C509         1       2     1     2
##  8         8     1        26     1    21     1 C503         1       1     1     3
##  9         9     1        79     1    21     1 C504         1       2     1     3
```

```
## 10        10     1        60     1    22     1 C502        1       2     1     3
## # ... with 29 more rows, and 4 more variables: n_6 <int>,
## #   RECURRENCE_LABEL <chr>, n_7 <int>, Survival_days_capped <dbl>
```

Uniquness of patient information

There are total 39 patients in the dataset with their corresponding 39 images. The original study from Keren et al. in 2018 have 41 patients however the images for patient 22 and 38 failed the quality check and were left out in the analysis.

The following code chunk will check the uniquness of `sample_id`, `patient_id` and `ImageNb`.

```
# unique values
unique_sample <- unique(TNBC$sample_id)
unique_patient <- unique(TNBC$patient_id)
unique_image <- unique(TNBC$ImageNb)

# if sample_id and patient_id have same length
dplyr::setequal(unique_sample, unique_patient)
```

```
## [1] FALSE
```

```
# if sample_id and ImageNb have same values
all.equal(unique_sample, unique_image)
```

```
## [1] TRUE
```

```
# since true, if one sample_id is unique to one patient_id (also one ImageNb)
TNBC %>%
  group_by(sample_id) %>%
  summarise(cell_count = n(),
            num_unique_image = length(unique(ImageNb)), ImageNb = unique(ImageNb), #Image 22 and 38 are
            num_unique_patient = length(unique(patient_id)), patient_id = unique(patient_id))
```

```
## # A tibble: 39 x 6
##    sample_id cell_count num_unique_image ImageNb num_unique_patient patient_id
##        <dbl>      <int>            <int>   <dbl>              <int>      <dbl>
##  1         1       5199                1       1                  1      30824
##  2         2       3033                1       2                  1      30805
##  3         3       5671                1       3                  1      30812
##  4         4       5381                1       4                  1      30838
##  5         5       4252                1       5                  1      30865
##  6         6       4894                1       6                  1      30847
##  7         7       3308                1       7                  1      30854
##  8         8       3786                1       8                  1      30846
##  9         9       5105                1       9                  1      30783
## 10        10       4066                1      10                  1      30781
## # ... with 29 more rows
```

```
## Conclusion: Each patient_id is unique to one patient_id and one ImageNb(the same)
```

Uniqueness of cell type

The columns `cell_type` and `mm` are the cell types for each cells, where `cell_type` is the factor level number and `mm` is the names of the cell_type. There are 16 different cell types in the dataset.

The following code chunk will perform the diagnostics of these two columns.

```
# unique values
unique_celltype <- unique(TNBC$cell_type)
unique_mm <- unique(TNBC$mm)

# if cell_type and mm have the same number
dplyr::setequal(length(unique_celltype), length(unique_mm))
```

```
## [1] TRUE
```

```
# since true, if cell_type and mm are unique to each other
TNBC %>%
  group_by(mm) %>%
  summarise(cell_count = n(),
            num_unique_cellType = length(unique(cell_type)), cell_type = unique(cell_type))
```

```
## # A tibble: 16 x 4
##    mm           cell_count num_unique_cellType cell_type
##    <chr>             <int>               <int>     <dbl>
##  1 B                 17084                   1         5
##  2 CD3 T              1135                   1         4
##  3 CD4 T              9918                   1         2
##  4 CD8 T             13376                   1         3
##  5 DC                 2381                   1        10
##  6 DC/Mono            1280                   1        11
##  7 Endothelial         279                   1        13
##  8 Epithelial        31871                   1        14
##  9 Mac                5552                   1         7
## 10 Mesenchymal       27698                   1        15
## 11 Mono/Neu            835                   1         9
## 12 Neu                1365                   1         8
## 13 NK                  285                   1         6
## 14 Other             56603                   1        16
## 15 Other immune       8669                   1        12
## 16 T reg               863                   1         1
```

```
## Conclusion: cell_type and mm are unique to each other(they are redunant)
```

```
# names of the cell types
unique_mm
```

```
##  [1] "B"            "CD3 T"        "CD4 T"        "CD8 T"        "DC"
##  [6] "DC/Mono"      "Endothelial"  "Epithelial"   "Mac"          "Mesenchymal"
## [11] "Mono/Neu"     "Neu"          "NK"           "Other"        "Other immune"
## [16] "T reg"
```

```
# tibble of number of samples each cell type present
TNBC %>%
  group_by(mm) %>%
  summarise(cell_count = n(), NumOfSamples = n_distinct(sample_id))
```

```
## # A tibble: 16 x 3
##    mm           cell_count NumOfSamples
##    <chr>             <int>        <int>
##  1 B                 17084           31
##  2 CD3 T              1135           22
##  3 CD4 T              9918           36
##  4 CD8 T             13376           36
##  5 DC                 2381           34
##  6 DC/Mono            1280           28
##  7 Endothelial         279           25
##  8 Epithelial        31871           39
##  9 Mac                5552           38
## 10 Mesenchymal       27698           39
## 11 Mono/Neu            835           36
## 12 Neu                1365           36
## 13 NK                  285           26
## 14 Other             56603           39
## 15 Other immune       8669           36
## 16 T reg               863           22
```

Cluster_id

Cells are clustered into different cell clusters by applying the technique FlowSOM, and it has 113 cluster for all cells in the `TNBC` dataset.

```
# unique cluster_id
unique_clusterId <- sort(unique(TNBC$cluster_id))
length(unique_clusterId)
```

```
## [1] 113
```

```
unique_clusterId
```

```
##   [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##  [19]  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##  [37]  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##  [55]  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##  [73]  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##  [91]  91  92  93  94  95  96  97  98  99 100 101 113 114 128 129 143 157 158
## [109] 159 172 175 187 211
```

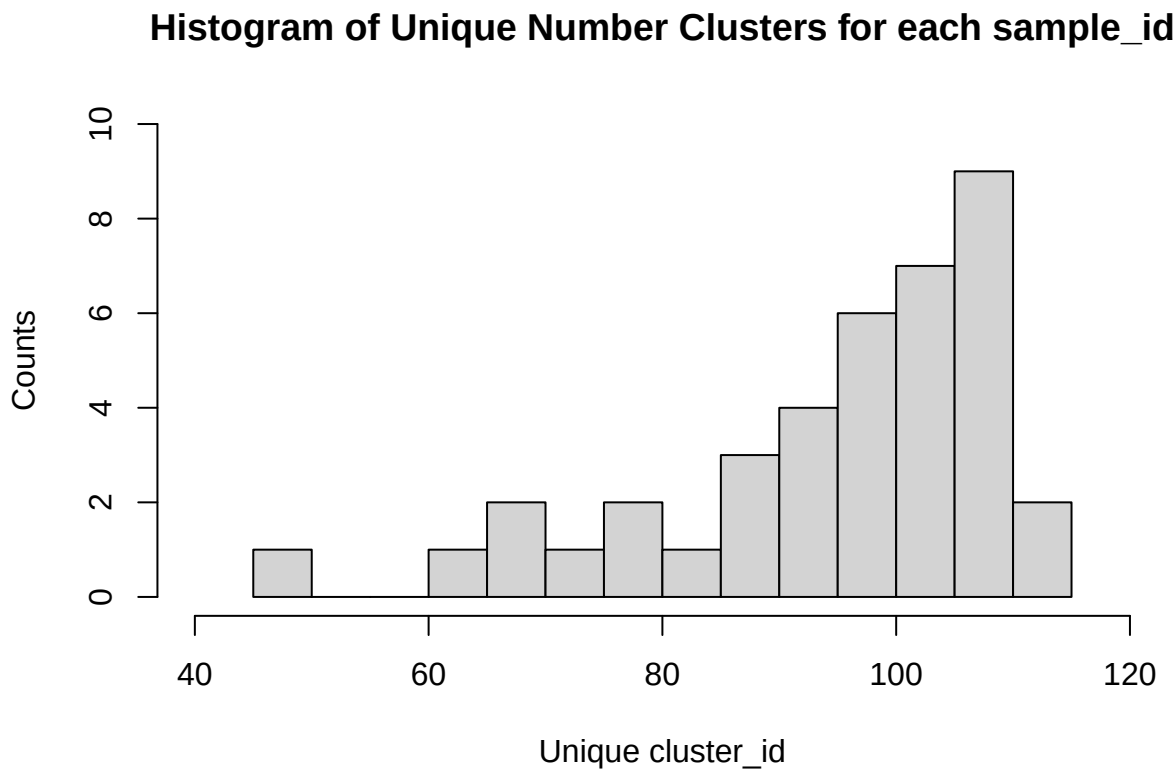
The following tibble and graphs show number of `cluster_id` presents in each sample.

```
# cluster_id for each sample_id
TNBC_clusterId <- TNBC %>%
  group_by(sample_id) %>%
```

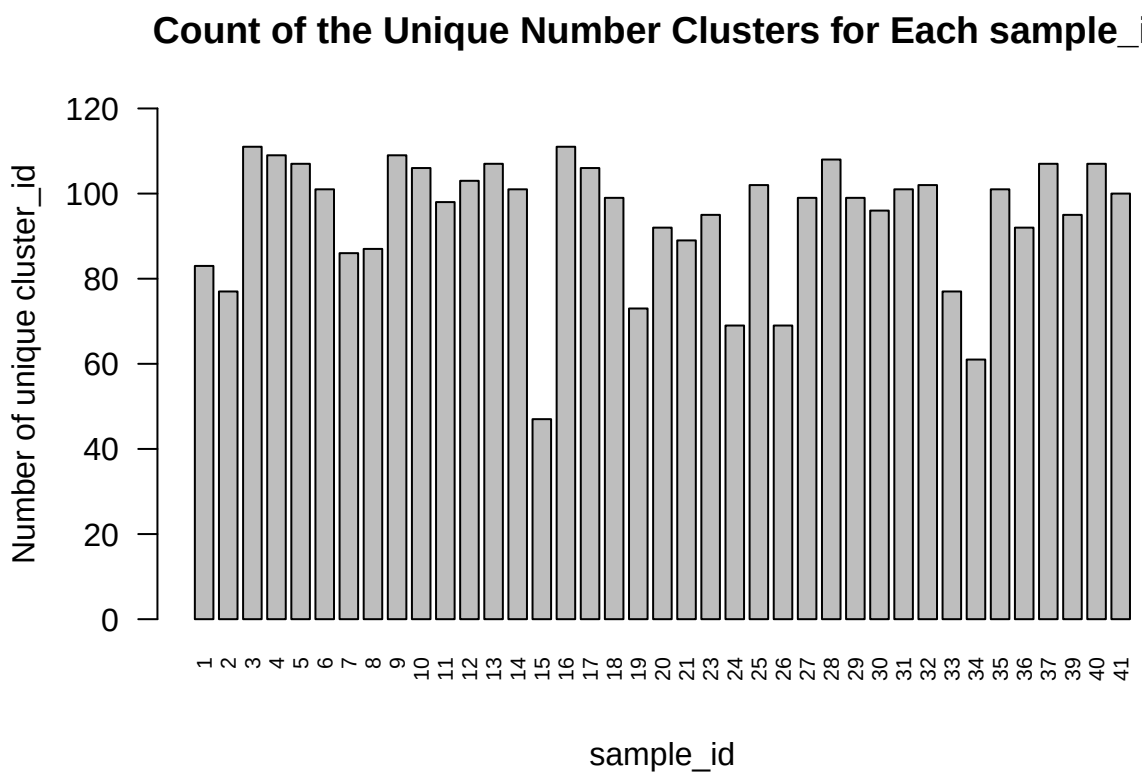
```
  summarise(cell_count = n(),
            num_unique_cluster = length(unique(cluster_id)))
TNBC_clusterId
```

```
## # A tibble: 39 x 3
##    sample_id cell_count num_unique_cluster
##        <dbl>      <int>              <int>
##  1         1       5199                 83
##  2         2       3033                 77
##  3         3       5671                111
##  4         4       5381                109
##  5         5       4252                107
##  6         6       4894                101
##  7         7       3308                 86
##  8         8       3786                 87
##  9         9       5105                109
## 10        10       4066                106
## # ... with 29 more rows
```

```
# Histogram
hist(TNBC_clusterId$num_unique_cluster,
     main = "Histogram of Unique Number Clusters for each sample_id",
     xlab = "Unique cluster_id", ylab = "Counts",
     breaks = 20, xlim = c(40, 120), ylim = c(0, 10))
```



```r
# Barplot
barplot(TNBC_clusterId$num_unique_cluster ~ TNBC_clusterId$sample_id,
        main = "Count of the Unique Number Clusters for Each sample_id",
        xlab = "sample_id", ylab = "Number of unique cluster_id",
        ylim = c(0, 120), width = 1.5, space = 0.3,
        las = 2, cex.names = 0.7) #las=1 for upright x names
```



Spatial Information

The two columns, `centroidX` and `centroidY` contain the spatial coordinates x and y for cells in each sample.

All spatial coordiantes We first want to investigate the range of all the spatial coordinates by finding their range, median and standard deviation.

The range of all spatial coordinate x is $(4.512, 2043.474)$, the median is 1054.774 and standard deviation is 577.792; the range of all spatial coordinates y is $(3.514, 2044.483)$, the median is is 1055.1165 and standard deviation is 580.025.

```r
# subset spatial coordinates
centroidX <- TNBC$centroidX
centroidY <- TNBC$centroidY

# Range for the whole dataset
range_wholeX <- c(min(centroidX), max(centroidX), median(centroidX), sd(centroidX))
range_wholeX
```

```
## [1]    4.511905 2043.474000 1054.774000  577.792180
```

```
range_wholeY <- c(min(centroidY), max(centroidY), median(centroidY), sd(centroidY))
range_wholeY
```

```
## [1]    3.514286 2044.483000 1055.116500  580.024631
```

Each cell types We now want to investigate the spatial coordinates by cell types in `TNBC` dataset. The following tibbles include the counts for each cell type along with the range, median and standard deviation of their spatial coordinates in all samples.

```
# centroidX
TNBC %>%
  group_by(mm) %>%
  summarise(count = n(),
            min = min(centroidX), max = max(centroidX),
            median = median(centroidX), sd = sd(centroidX)) %>%
  knitr::kable()
```

mm	count	min	max	median	sd
B	17084	4.582417	2043.392	1068.6680	541.1996
CD3 T	1135	6.784946	2043.073	1424.7880	500.7704
CD4 T	9918	4.610000	2043.281	1323.7545	559.5474
CD8 T	13376	4.688073	2043.297	1110.1715	570.1150
DC	2381	4.566929	2043.363	948.9395	645.9324
DC/Mono	1280	8.995781	2043.230	1308.6955	499.2984
Endothelial	279	28.724440	2022.824	1309.8260	535.6648
Epithelial	31871	4.567416	2043.380	1118.1760	566.4550
Mac	5552	4.686047	2043.274	1194.2770	559.2432
Mesenchymal	27698	4.555556	2043.347	1028.6410	575.8002
Mono/Neu	835	10.019080	2042.503	1266.4960	546.5867
Neu	1365	8.695842	2042.541	1153.5900	569.1121
NK	285	15.819670	2029.213	1343.4280	523.0047
Other	56603	4.511905	2043.474	914.0930	587.5303
Other immune	8669	4.641791	2043.379	1056.2430	572.1199
T reg	863	9.922190	2039.853	1233.3820	511.7799

```
# centroidY
TNBC %>%
  group_by(mm) %>%
  summarise(count = n(),
            min = min(centroidY), max = max(centroidY),
            median = median(centroidY), sd = sd(centroidY)) %>%
  knitr::kable()
```

mm	count	min	max	median	sd
B	17084	4.600000	2044.441	1143.6310	577.0482
CD3 T	1135	14.928100	2043.847	1070.0920	479.8067
CD4 T	9918	3.855556	2044.359	1106.5920	553.1205

mm	count	min	max	median	sd
CD8 T	13376	4.039474	2044.379	1095.8965	574.7819
DC	2381	5.198718	2043.463	1122.3520	618.9403
DC/Mono	1280	3.650000	2043.193	1003.7210	534.9473
Endothelial	279	20.383230	2010.723	901.5644	493.5723
Epithelial	31871	4.493421	2044.347	949.2707	558.0874
Mac	5552	5.909605	2043.214	977.1770	569.4235
Mesenchymal	27698	3.777778	2044.483	979.0533	580.2001
Mono/Neu	835	10.988020	2044.450	1257.1100	531.7334
Neu	1365	7.002710	2042.952	1213.8160	538.2672
NK	285	18.128130	2042.485	1040.5380	540.3429
Other	56603	3.514286	2044.471	1121.4110	594.5656
Other immune	8669	4.250000	2044.372	1041.3730	587.8098
T reg	863	10.049020	2038.496	973.4490	525.8779

Each cell type in each sample We next want to group the `TNBC` data by sample number and cell type, and investigate the spatial coordinates. The following tibbles include `sample_id` and cell type information along with the range, median and standard deviation of their spatial coordinates in all samples.

```
## centroidX
TNBC %>%
  group_by(sample_id, mm) %>%
  summarise(n = n(),
            min = min(centroidX), max = max(centroidX),
            median = median(centroidX), sd = sd(centroidX))
```

```
## # A tibble: 523 x 7
## # Groups:   sample_id [39]
##    sample_id mm              n     min   max median    sd
##        <dbl> <chr>       <int>   <dbl> <dbl>  <dbl> <dbl>
##  1         1 B             734   34.6  2022.  1062.  321.
##  2         1 CD4 T         152  100.   2043.  1566.  516.
##  3         1 CD8 T         147    9.66 2041.  1147.  615.
##  4         1 DC              1  238.    238.   238.   NA
##  5         1 Epithelial     20   44.7  2042.  1742.  584.
##  6         1 Mac            10  347.   2003.  1520.  528.
##  7         1 Mesenchymal   289   36.4  2042.  1074.  518.
##  8         1 Mono/Neu        2 1540.   1938.  1739.  281.
##  9         1 Neu             2 1159.   1572.  1365.  292.
## 10         1 NK              4  610.   1141.   644.  255.
## # ... with 513 more rows
```

```
## centroidY
TNBC %>%
  group_by(sample_id, mm) %>%
  summarise(n = n(),
            min = min(centroidY), max = max(centroidY),
            median = median(centroidY), sd = sd(centroidY))
```

```
## # A tibble: 523 x 7
## # Groups:   sample_id [39]
```

```
##    sample_id mm              n    min   max median    sd
##        <dbl> <chr>       <int>  <dbl> <dbl>  <dbl> <dbl>
##  1         1 B             734   6.73 2043.  1793.  525.
##  2         1 CD4 T         152  12.7  2000.  1005.  635.
##  3         1 CD8 T         147  12.1  1998.  1074.  624.
##  4         1 DC              1 412.    412.   412.   NA
##  5         1 Epithelial     20   6.44  818.   240.  284.
##  6         1 Mac            10 418.   1767.  1100.  461.
##  7         1 Mesenchymal   289   8.05 2038.   596.  538.
##  8         1 Mono/Neu        2 170.   1415.   793.  880.
##  9         1 Neu             2 988.   1686.  1337.  494.
## 10         1 NK              4 301.   1086.   872.  349.
## # ... with 513 more rows
```