# Compare two classifications

## Zuguang Gu (z.gu@dkfz.de (mailto:z.gu@dkfz.de))

### 2021-12-17

In this supplementary file, we demonstrate how to calculate overall agreement for two classifications. Here we use two classifications by cola and Seurat on the PBMC scRNASeq dataset.

```
tb = readRDS("cola_Seurat_classification.rds")
head(tb)
```

```
##   cola_class Seurat_class
## 1         01            2
## 2         03            3
## 3         01            2
## 4       0212            1
## 5        042            6
## 6         01            2
```

The contingency table on the two classifications:

```
table(tb)
```

```
##           Seurat_class
## cola_class   0   1   2   3   4   5   6   7   8
##       01   692   0 447   2  50   0   0   1   1
##       0211   0   3   0   0   0 139   0   0   1
##       0212   0  90   0   0   0  17   0  22   0
##       022    0 386   0   0   0   6   0   7   1
##       03     0   1   0 342   0   0   0   1   0
##       041   19   0  21   0 208   0   4   0   0
##       042    0   0   4   0  21   0 140   1  11
```

Next we define a function that calculates overlap coeffcient for every pair of classes in the two classifications:

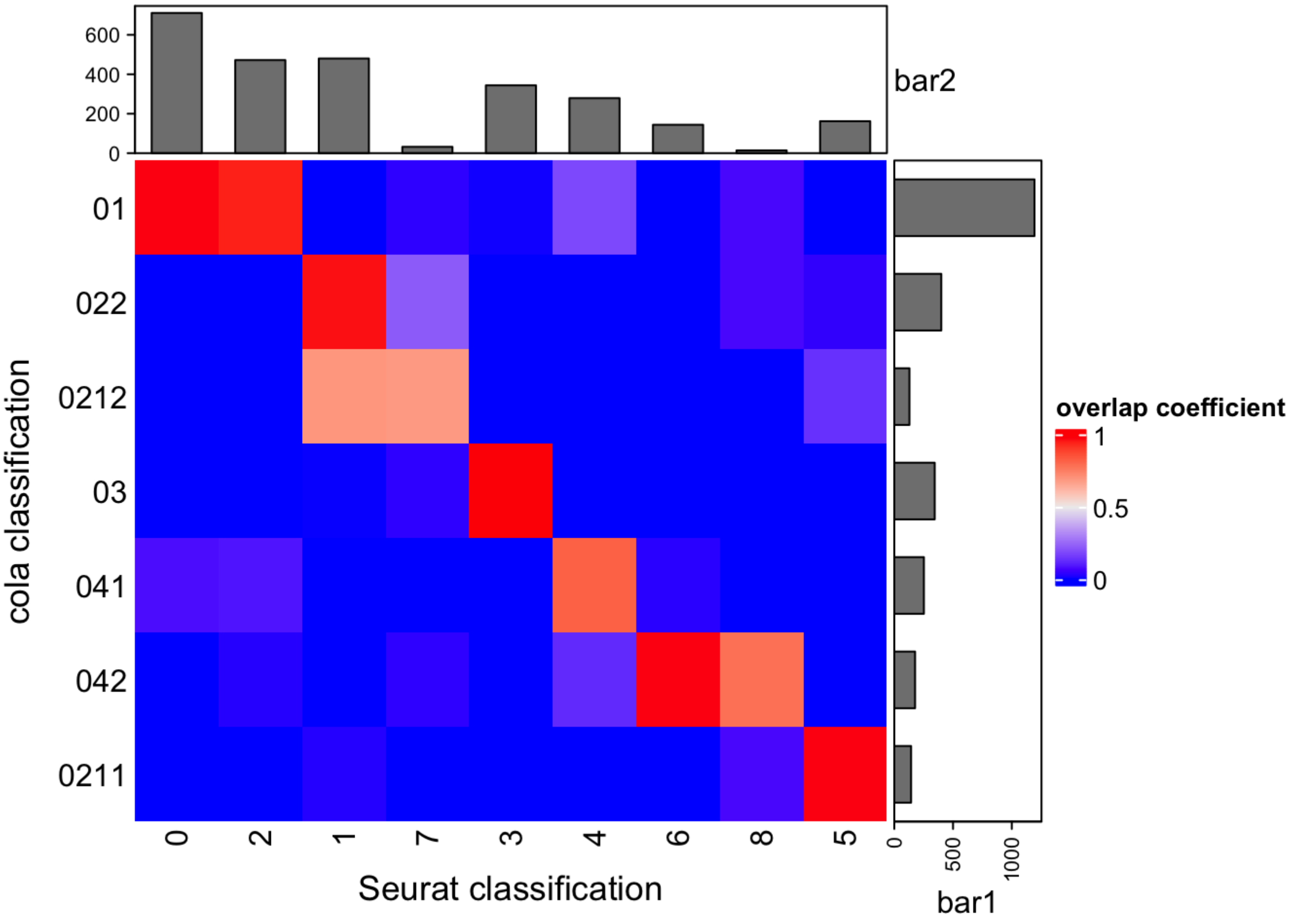
```
overlap_coefficient = function(x, y) {
    le1 = unique(x)
    le2 = unique(y)

    om = matrix(nrow = length(le1), ncol = length(le2))
    dimnames(om) = list(le1, le2)
    for(a in le1) {
        for(b in le2) {
            om[a, b] = sum(x == a & y == b)/min(sum(x == a), sum(y == b))
        }
    }
    om
}
m = overlap_coefficient(tb$cola_class, tb$Seurat_class)
```

We manually reorder `m` and visualize it via a heatmap. The two barplot annotations shows number of samples in each class.

```
library(ComplexHeatmap)

m = m[c("01", "022", "0212", "03", "041", "042", "0211"),
      c("0", "2", "1", "7", "3", "4", "6", "8", "5")]
t1 = as.vector(table(tb$cola_class)[rownames(m)])
t2 = as.vector(table(tb$Seurat_class)[colnames(m)])
Heatmap(m, name = "overlap coefficient",
    right_annotation = rowAnnotation(bar1 = anno_barplot(t1, width = unit(2, "cm"))),
    top_annotation = HeatmapAnnotation(bar2 = anno_barplot(t2, height = unit(2, "cm"
))),
    row_title = "cola classification", row_names_side = "left",
    column_title = "Seurat classification", column_title_side = "bottom",
    cluster_rows = FALSE, cluster_columns = FALSE)
```



Generally speaking, if cells with high overlap coefficents locate on the diagonal of the heatmap, it means the two classifications highly agrees.

For every class in cola classification (which is on rows of `m`), the agreement to Seurat classification is defined as the maximal overlap coeffcient to all classes in Seurat.

```
library(matrixStats)
v = rowMaxs(m)
v
```

```
## [1] 0.9732771 0.9650000 0.6976744 0.9941860 0.8253968 0.9722222 0.9720280
```

Finally, the overlap classification agreement of cola to Seurat is defined as the mean agreement of each class to Seurat weighted by the class size.

```
size = table(tb$cola_class)
size = size[rownames(m)]
sum(v*size)/sum(size)
```

```
## [1] 0.9470064
```

The process can be wrapped into a function `overall_classification_agreement()`:

```
overall_classification_agreement = function(x, y) {
    m = overlap_coefficient(x, y)

    size = table(x)
    size = size[rownames(m)]
    v = rowMaxs(m)
    sum(v*size)/sum(size)
}
```

We can test the overlap classification agreement of cola to Seurat and of Seurat to cola. The two values are not the same, but are highly similar.

```
overall_classification_agreement(tb$cola_class, tb$Seurat_class)
```

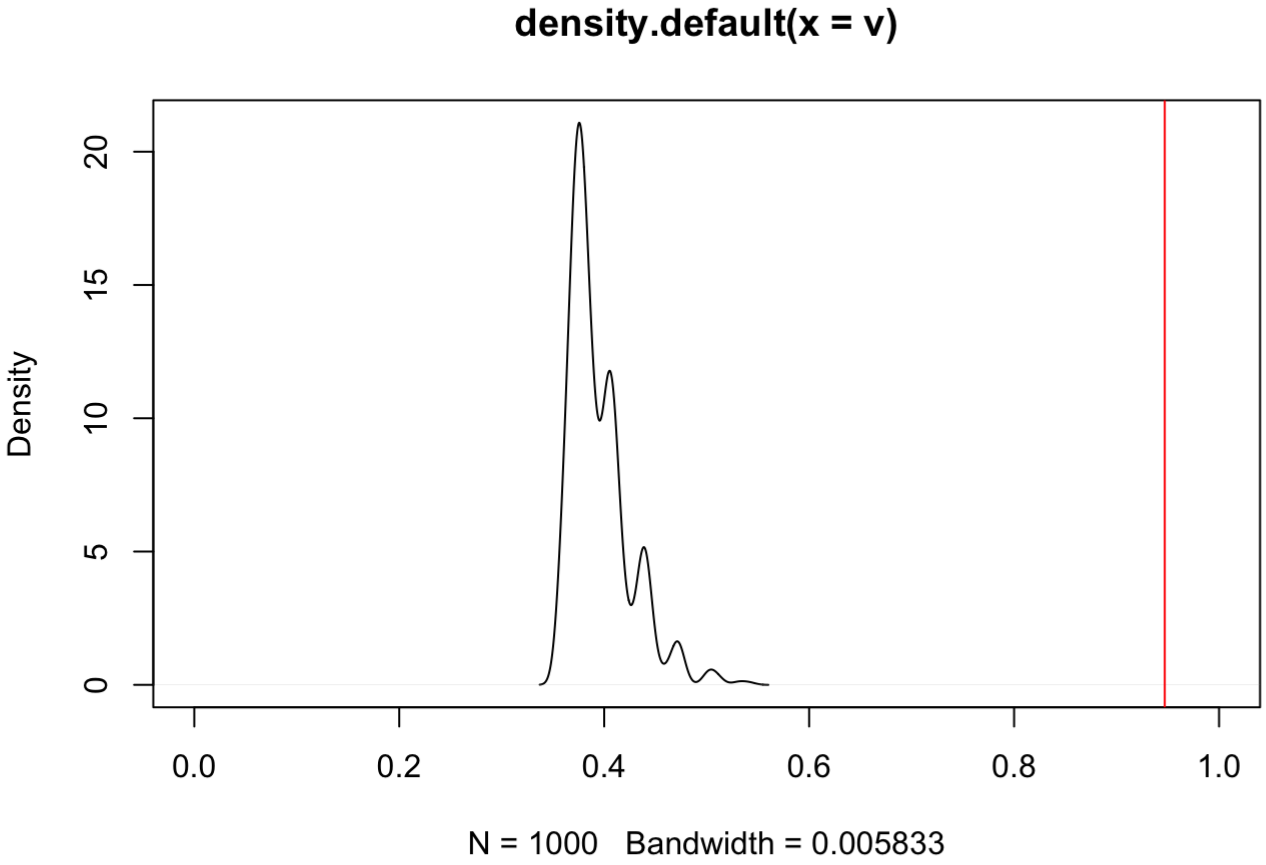
```
## [1] 0.9470064
```

```
overall_classification_agreement(tb$Seurat_class, tb$cola_class)
```

```
## [1] 0.9495657
```

We can randomly permute the Seurat classification to get a null distribution of the overlap classification agreement.

```
set.seed(123)
v = replicate(1000,
    overall_classification_agreement(tb$cola_class, sample(tb$Seurat_class, nrow(t
b)))
)
plot(density(v), xlim = c(0, 1))
abline(v = 0.9470064, col = "red")
```


density.default(x = v)
Density
0
5
10
15
20
0.0
0.2
0.4
0.6
0.8
1.0
N = 1000 Bandwidth = 0.005833