ARI benchmark

Hector Roux de Bézieux

16 September , 2019

Contents

Data

We use the nuclei dataset from Dune:

```
data("nuclei", package = "Dune")
```

ARI function

This ARI function is exactly the same as the one from `mclust`. We jsut break it into two parts for analysis: computing the confusion matrix, and then computing the ARI on the matrix.

```
confusionMatrix <- function(x, y) {
    x <- as.vector(x)
    y <- as.vector(y)
    if (length(x) != length(y))
        stop("arguments must be vectors of the same length")
    tab <- table(x, y)
    return(tab)
}

computeARI <- function(tab) {
  if (all(dim(tab) == c(1, 1))) return(1)
  a <- sum(choose(tab, 2))
  b <- sum(choose(rowSums(tab), 2)) - a
  c <- sum(choose(colSums(tab), 2)) - a
  d <- choose(sum(tab), 2) - a - b - c
  ARI <- (a - (a + b) * (a + c)/(a + b + c + d)) /
    ((a + b + a + c)/2 - (a + b) * (a + c)/(a + b + c + d))
  return(ARI)
}

x <- nuclei$SC3
y <- nuclei$Monocle
tab <- confusionMatrix(x, y)

bench <- microbenchmark(confusionMatrix(x, y), computeARI(tab), times = 200)
knitr::kable(summary(bench))
```

expr	min	lq	mean	median	uq	max	neval	cld
confusionMatrix(x, y)	462.808	561.7975	692.721	595.3205	696.859	4903.381	200	a

expr	min	lq	mean	median	uq	max	neval	cld
computeARI(tab)	43.021	51.1020	291.599	67.4430	87.811	43564.498	200	a