Benchmark comparison

Stelios Batziakas (ampatziakas@gmail.com)

22 August, 2017

The pasaR borrows heavily on the functions of package *micropan* while trying to use functions from tidyverse and optimize code for speed. In this vignette, all functions with same output in both packages are benchmarked. Micropan 1.1.2 was used.

Load Data

Benchmark tests are done using the Mpneumoniae dataset from package *micropan*. It contains genes from 7 genomes in 1210 clusters.

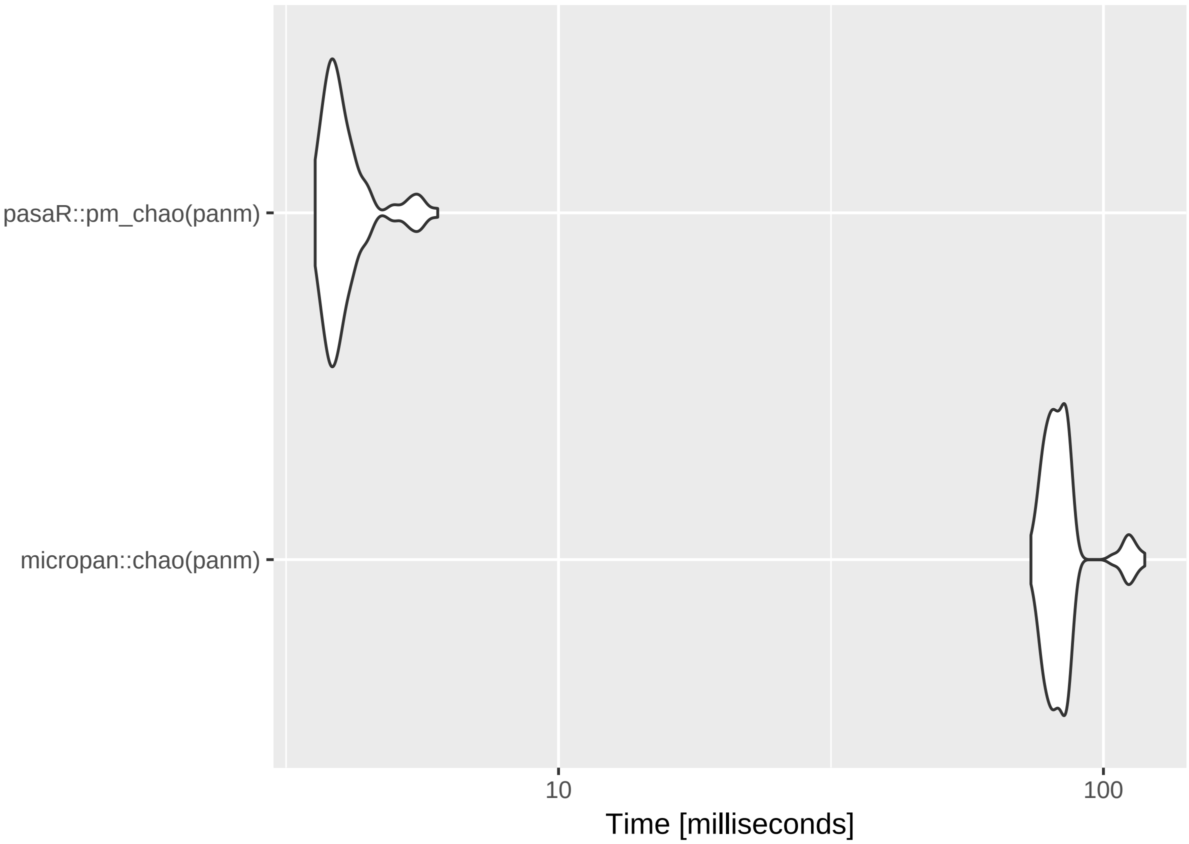
```
data("Mpneumoniae.blast.panmat")
panm<-as.data.frame(Mpneumoniae.blast.panmat)
```

Chao estimator Comparison

Speed comparison for the Chao estimator computing.

```
chao_comparison<-microbenchmark(micropan::chao(panm),pasaR::pm_chao(panm))
```

expr	min	lq	mean	median	uq	max	neval
micropan::chao(panm)	73.61	79.08	84.39	82.18	85.82	119.1	100
pasaR::pm_chao(panm)	3.576	3.785	4.077	3.923	4.159	6.001	100

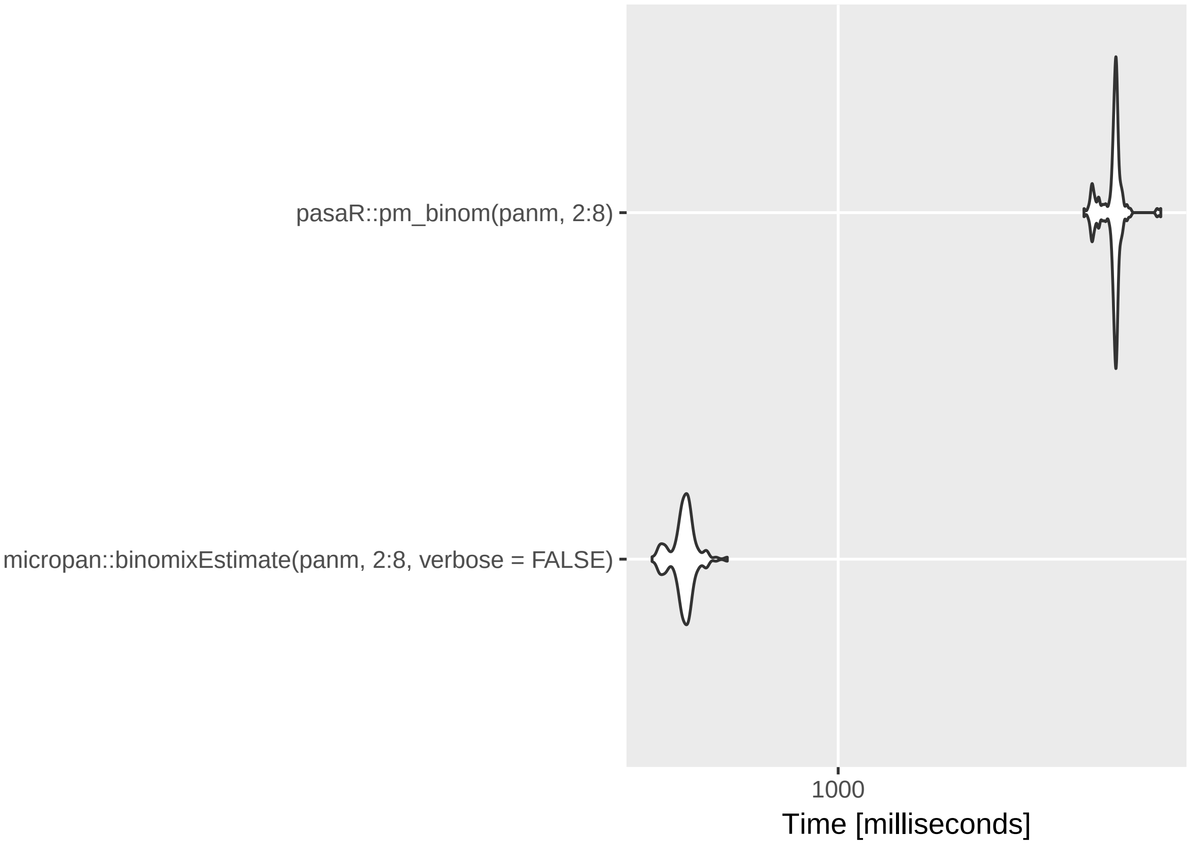


Binomial mixture models

Speed comparison for binomial mixture model fitting, searching between 2 to 8 underlying components. The binomial mixture model is the only function that performs faster is the *micropan* package. This is caused by the difference in micropan:::binomixMachine and pasar:::binomixMachine internal functions specifics, ie. the controls of the model optimizer. Micropan allows for 300 max iterations in order to discover the optimal solution. However was not sufficient for large datasets (clusters> 100k) where it was observed that the BIC criterion value was fixated in "ranges" of components. In order to fix this maximum iterations now are 200 * K where K is the number of supposed underlying components of the mixture.

```
binomial_comparison<-microbenchmark(micropan::binomixEstimate(panm,2:8,verbose=FALSE),
                                    pasaR::pm_binom(panm,2:8))
```

expr	min	lq	mean	median	uq	max	neval
micropan::binomixEstimate(panm, 2:8, verbose = FALSE)	648	690.1	698.1	699.9	707	772.2	100
pasaR::pm_binom(panm, 2:8)	1774	1893	1898	1909	1915	2122	100

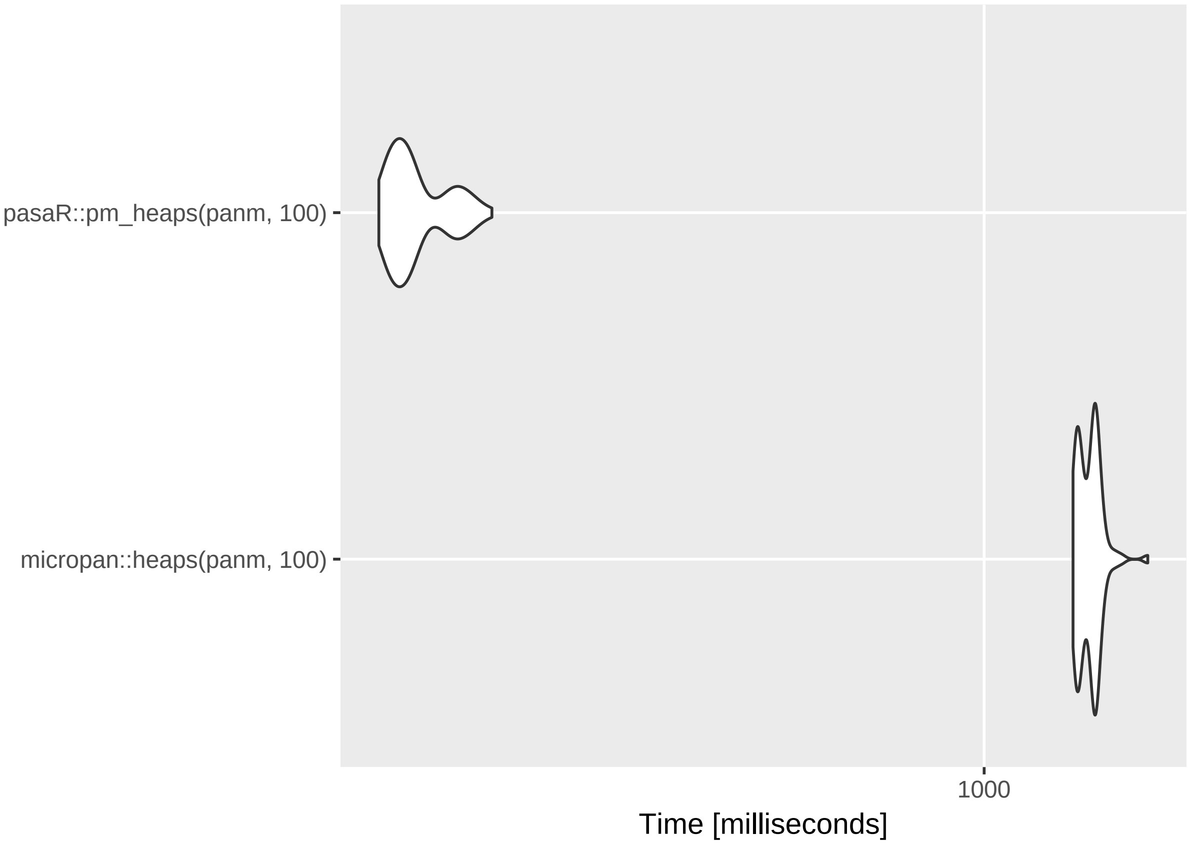


Heaps Law Fit

Speed comparison for power law fitting with 100 permutations.

```
heaps_comparison<- microbenchmark(micropan::heaps(panm,100),pasaR::pm_heaps(panm,100))
```

expr	min	lq	mean	median	uq	max	neval
micropan::binomixEstimate(panm, 2:8, verbose = FALSE)	648	690.1	698.1	699.9	707	772.2	100
pasaR::pm__binom(panm, 2:8)	1774	1893	1898	1909	1915	2122	100

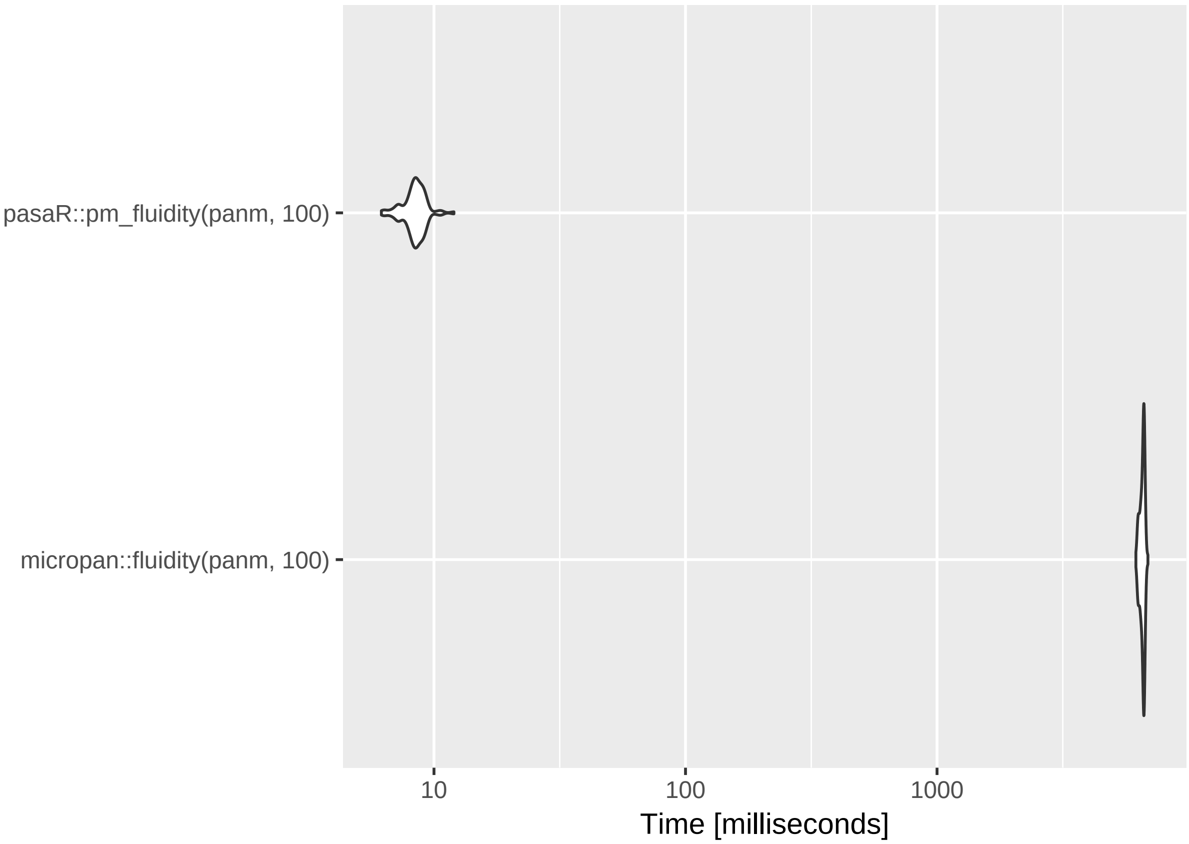


Fluidity

Speed comparison for fluidity computation with resampling, with 100 permutations.

```
fluidity_comparison<-microbenchmark(micropan::fluidity(panm,100),pasaR::pm_fluidity(panm,100))
```

expr	min	lq	mean	median	uq	max	neval
micropan::fluidity(panm, 100)	6175	6453	6556	6606	6655	6893	100
pasaR::pm_fluidity(panm, 100)	6.179	8.058	8.429	8.461	8.948	12	100


pasaR::pm_fluidity(panm, 100)
micropan::fluidity(panm, 100)
10
100
1000
Time [milliseconds]