

Qusage: Speeding up qusage in RcppArmadillo

Timothy J. Triche, Jr, Anthony R. Colombo

11 March, 2016

Contents

1	SpeedSage Intro	1
1.1	changes Armadillo C++	1

1 SpeedSage Intro

qusage is published software that is slow for large runs, SpeedSage corrects for speed and efficiency at large orders. there is Bottlenecking of Functions Qusage can improve the speed of its algorithm by minimizing the cost of computaiton.

1.1 changes Armadillo C++

trading NA flexibility slows down qusage runs, but having the user input no NAs enforcing good input, this speeds up calcIndividualExpressions, as well as using C++ libraries.

```
library(inline)
library(microbenchmark)
library(Rcpp)
```

```
##
## Attaching package: 'Rcpp'

## The following object is masked from 'package:inline':
##
##   registerPlugin
```

```
library(parallel)
library(speedSage)
```

```
## Loading required package: limma
```

```
library(qusage)
```

```
##
## Attaching package: 'qusage'

## The following objects are masked from 'package:speedSage':
##
##   aggregateGeneSet, calcBayesCI, getXcoords, read.gmt
```

```

library(ggplot2)
eset<-system.file("extdata","eset.RData",package="speedSage")
load(eset)
labels<-c(rep("t0",134),rep("t1",134))
contrast<-"t1-t0"
colnames(eset)<-c(rep("t0",134),rep("t1",134))
fileISG<-system.file("extdata","c2.cgp.v5.1.symbols.gmt",package="speedSage")
ISG.geneSet<-read.gmt(fileISG)
geneSets<-ISG.geneSet[grepl("DER_IFN_GAMMA_RESPONSE_UP",names(ISG.geneSet))]

#cpp functions
sourceCpp(file="/home/anthonycolombo/Documents/qusage/qusage_repos/qusage_speed/R/sigmaArm.cpp")
sourceCpp(file="/home/anthonycolombo/Documents/qusage/qusage_repos/qusage_speed/R/sigmaSingle.cpp")
sourceCpp(file="/home/anthonycolombo/Documents/qusage/qusage_repos/qusage_speed/R/bayesEstimation.cpp")
sourceCpp(file="/home/anthonycolombo/Documents/qusage/qusage_repos/qusage_speed/R/notbayesEstimation.cpp")
sourceCpp(file="/home/anthonycolombo/Documents/qusage/qusage_repos/qusage_speed/R/calcVIFarm.cpp")
sourceCpp(file="/home/anthonycolombo/Documents/qusage/qusage_repos/qusage_speed/R/calcVIFarmalt.cpp")
sourceCpp(file="/home/anthonycolombo/Documents/qusage/qusage_repos/qusage_speed/R/calcVIFarm_nosdalphaa.cpp")

pairVector<-NULL
var.equal<-FALSE
filter.genes<-FALSE
n.points<-2^12

#setting up calcVif call objects
source("/home/anthonycolombo/Documents/qusage/qusage_repos/qusage_speed/R/qusageArm.R")
useAllData<-TRUE
useCAMERA<-FALSE
##### QuSage::qusage default calc
defaultSage<-qusage(eset,labels,contrast,geneSets,var.equal=FALSE)

## Calculating gene-by-gene comparisons...

## Found more than one class "QSarray" in cache; using the first, from namespace 'speedSage'

## Done.
## Aggregating gene data for gene sets.Done.
## Calculating variance inflation factors...Done.

mySage<-qusageArm(eset,labels,contrast,geneSets,var.equal=FALSE)

## Calculating gene-by-gene comparisons...Done.
## Aggregating gene data for gene sets.Done.
## Calculating variance inflation factors...Done.

library(speedSage)
useCAMERA<-FALSE
sourceCpp(file="/home/anthonycolombo/Documents/qusage/qusage_repos/qusage_speed/R/calcVIFarmalt.cpp")
sourceCpp(file="/home/anthonycolombo/Documents/qusage/qusage_repos/qusage_speed/R/calcVIFarm_nosdalphaa.cpp")

```


[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

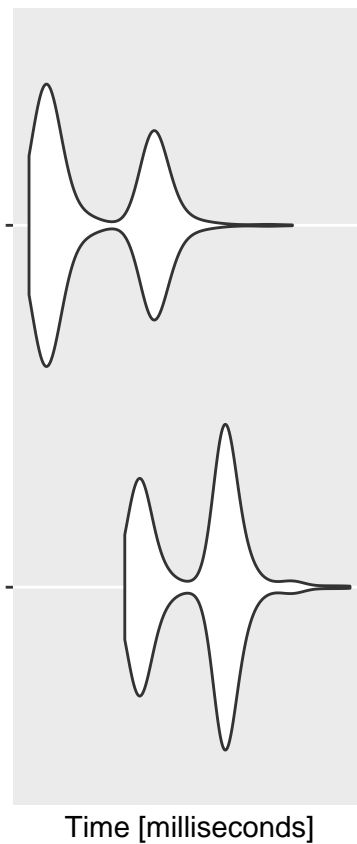
[illegible]

```
## Calculating gene-by-gene comparisons...Done.
## Aggregating gene data for gene sets.Done.
## Calculating variance inflation factors...Done.
## Calculating gene-by-gene comparisons...Done.
## Aggregating gene data for gene sets.Done.
## Calculating variance inflation factors...Done.
## Calculating gene-by-gene comparisons...Done.
## Aggregating gene data for gene sets.Done.
## Calculating variance inflation factors...Done.
## Calculating gene-by-gene comparisons...Done.
## Aggregating gene data for gene sets.Done.
## Calculating variance inflation factors...Done.
```

```
autoplot(speedUp)
```

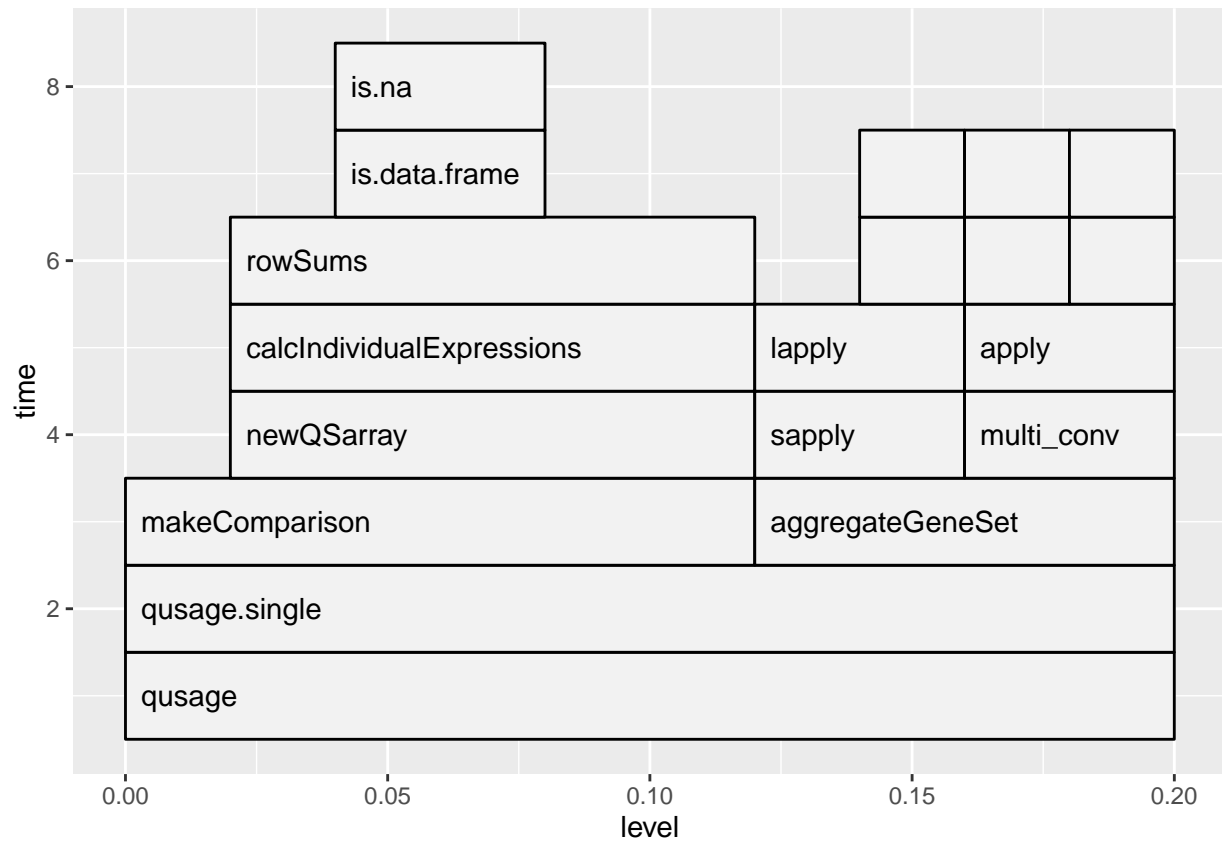
```
mySage <- qusageArm(eset, labels, contrast, geneSets, var.equal = FALSE)
```

```
defaultSage <- qusage(eset, labels, contrast, geneSets, var.equal = FALSE)
```



```
library(profr)
defaultX<-profr(qusage(eset,labels,contrast,geneSets,var.equal=FALSE))
myX<-profr(qusageArm(eset,labels,contrast,geneSets,var.equal=FALSE))

##default qusage
ggplot(defaultX)
```



```
##with armadillo plot
ggplot(myX)
```

