

Qusage: Speeding in Parallel

Timothy J. Triche, Jr, Anthony R. Colombo, Harold Pimentel

08 February, 2016

Contents

1	SpeedSage Intro	1
1.1	changes calcIndividualExpressionsC	1
2	Individual Expression Function	1

1 SpeedSage Intro

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

1.1 changes calcIndividualExpressionsC

trading NA flexibility slows down qusage runs, but having the user input no NAs enforcing good input, this speeds up calcIndividualExpressionsC 2X

2 Individual Expression Function

This test the local version which enforces no NA in Baseline or PostTreatment object, this reduces the flexibility.

```
library(speedSage)
```

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

```
library(qusage)
```

```
##
```

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

```
## The following object is masked from 'package:speedSage':
```

```
##
```

```
## makeComparison
```

```

eset<-system.file("extdata","eset.RData",package="speedSage")
load(eset)
labels<-c(rep("t0",134),rep("t1",134))
contrast<-"t1-t0"
fileISG<-system.file("extdata","c2.cgp.v5.1.symbols.gmt",package="speedSage")
ISG.geneSet<-read.gmt(fileISG)
ISG.geneSet<-ISG.geneSet[grepl("DER_IFN_GAMMA_RESPONSE_UP",names(ISG.geneSet))]
Baseline<-eset
PostTreatment<-eset+20.4
#non-paired
test1<-calcIndividualExpressions(Baseline,PostTreatment,paired=FALSE,min.variance.factor=10^-6,na.rm=TRUE)

```

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

```

test2<-calcIndividualExpressionsC(Baseline,PostTreatment,paired=FALSE,min.variance.factor=10^-6,no.cores)
identical(test2,test1)

```

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

```

library(microbenchmark)
mb<-microbenchmark(
test1<-calcIndividualExpressions(Baseline,PostTreatment,paired=FALSE,min.variance.factor=10^-6,na.rm=TRUE)
test2<-calcIndividualExpressionsC(Baseline,PostTreatment,paired=FALSE,min.variance.factor=10^-6,no.cores)
#on average 1.49X faster
mb

```

```
## Unit: milliseconds
```

```

##
## test1 <- calcIndividualExpressions(Baseline, PostTreatment, paired = FALSE, min.variance.factor = 10^-6, na.rm = TRUE)
## test2 <- calcIndividualExpressionsC(Baseline, PostTreatment, paired = FALSE, min.variance.factor = 10^-6, no.cores)
##      min      lq      mean     median      uq      max neval cld
## 169.8350 173.8207 192.2252 178.4514 217.7935 277.1283   100   b
## 128.4388 137.3831 165.2271 150.8124 191.8840 300.1632   100   a

```

```
require(profr)
```

```
## Loading required package: profr
```

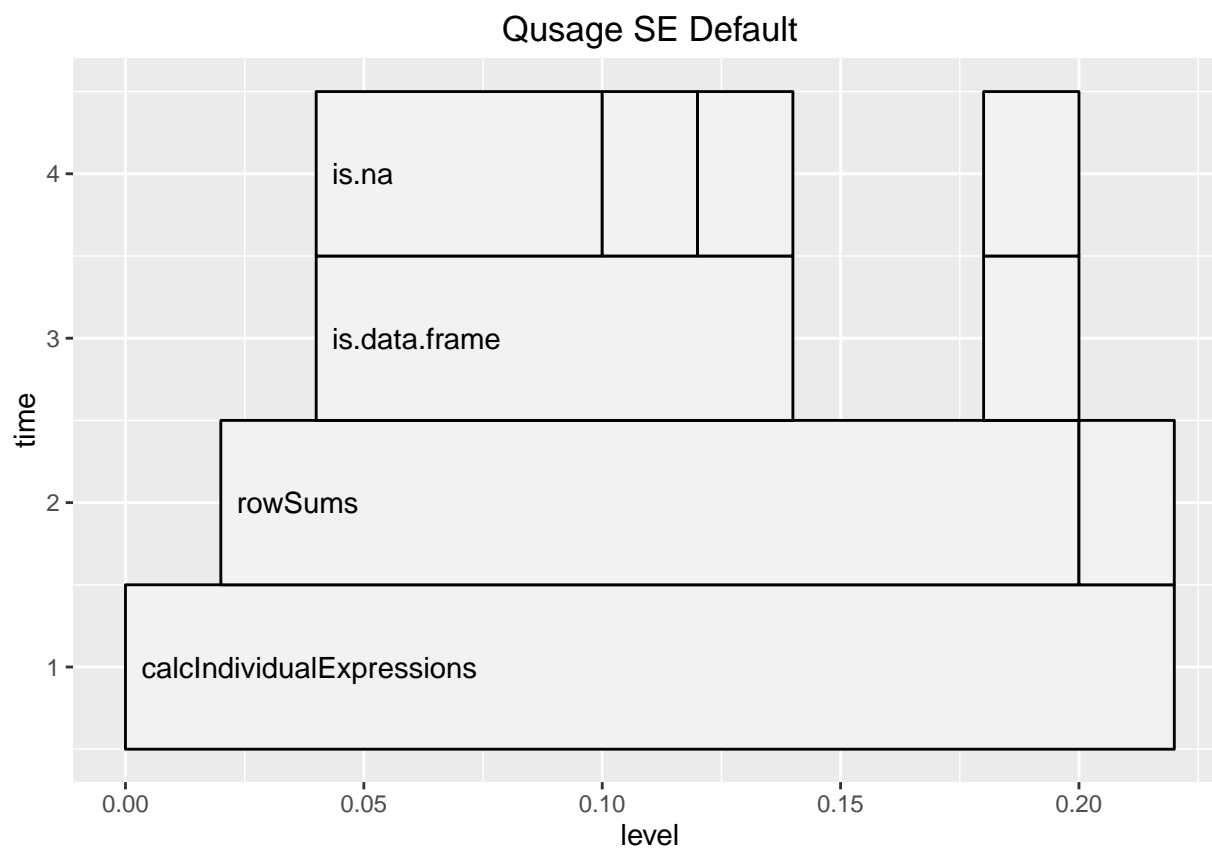
```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

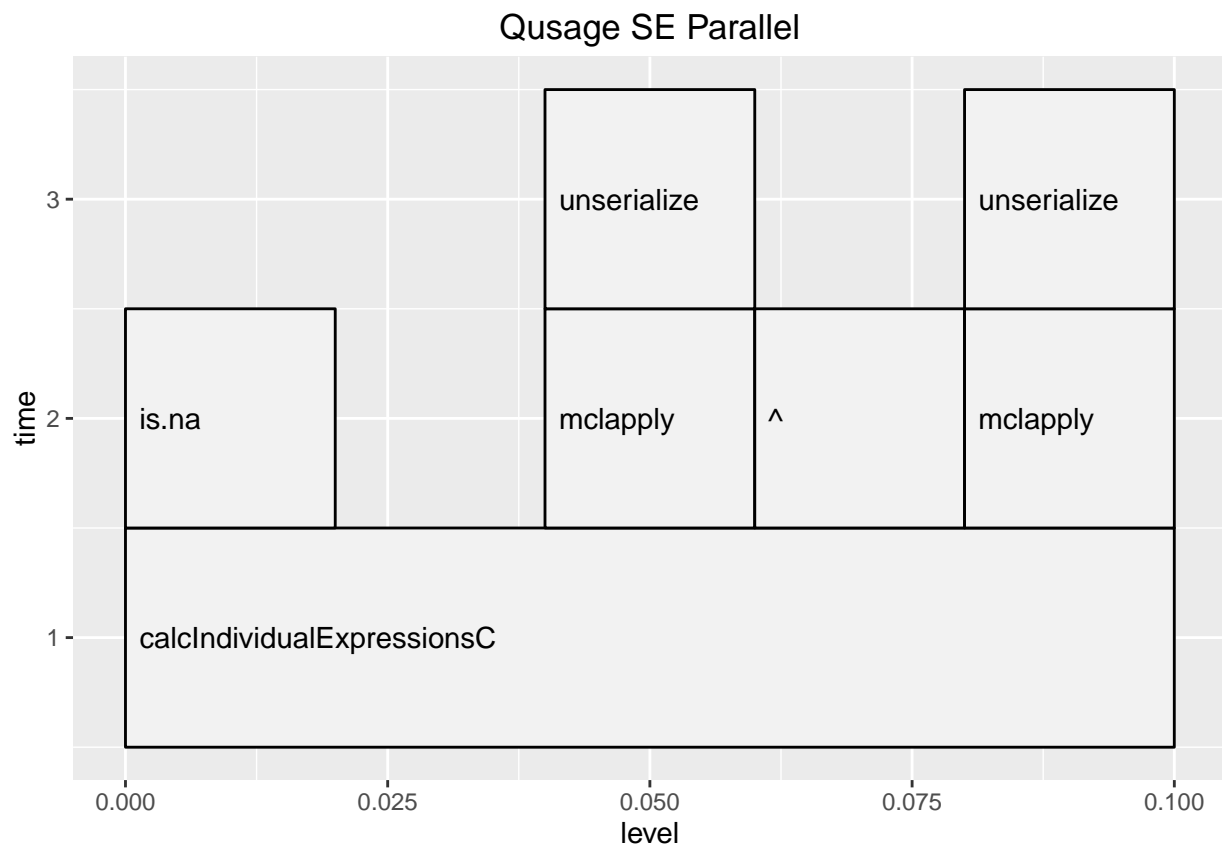
```

x1<-profr(calcIndividualExpressions(Baseline,PostTreatment,paired=FALSE,min.variance.factor=10^-6,na.rm=TRUE))
ggplot(x1)+labs(title="Qusage SE Default")

```



```
x2<-profr(calcIndividualExpressionsC(Baseline,PostTreatment,paired=FALSE,min.variance.factor=10^-6))
ggplot(x2)+labs(title="Qusage SE Parallel")
```



```
#paired end testing
testPE1<-calcIndividualExpressions(Baseline,PostTreatment,paired=TRUE,min.variance.factor=10^-6,na.rm=TRUE)
testPE2<-calcIndividualExpressionsC(Baseline,PostTreatment,paired=TRUE,min.variance.factor=10^-6)
for(i in 1:length(test1)){
  message(paste0(identical(testPE1[[i]],testPE2[[i]])," ",i))
}
```

```
## TRUE 1
```

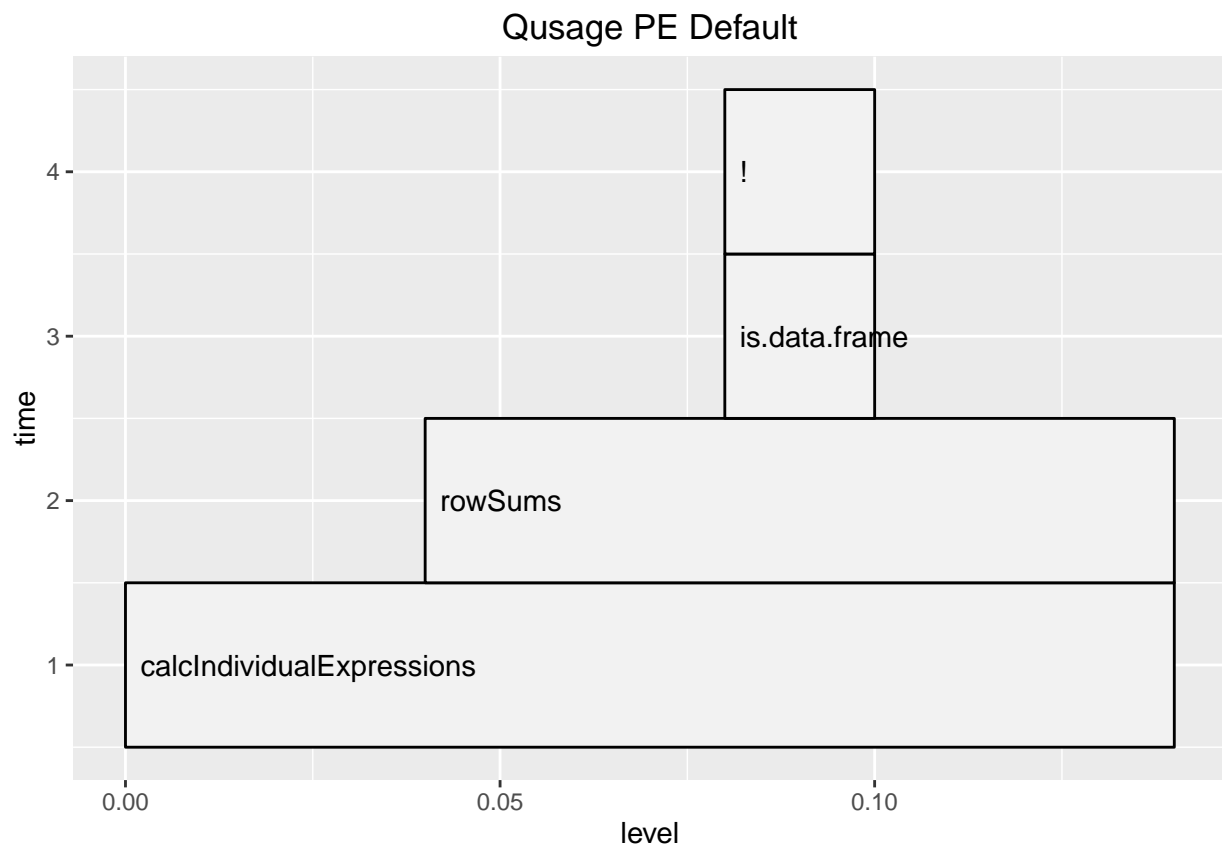
```
## TRUE 2
```

```
## TRUE 3
```

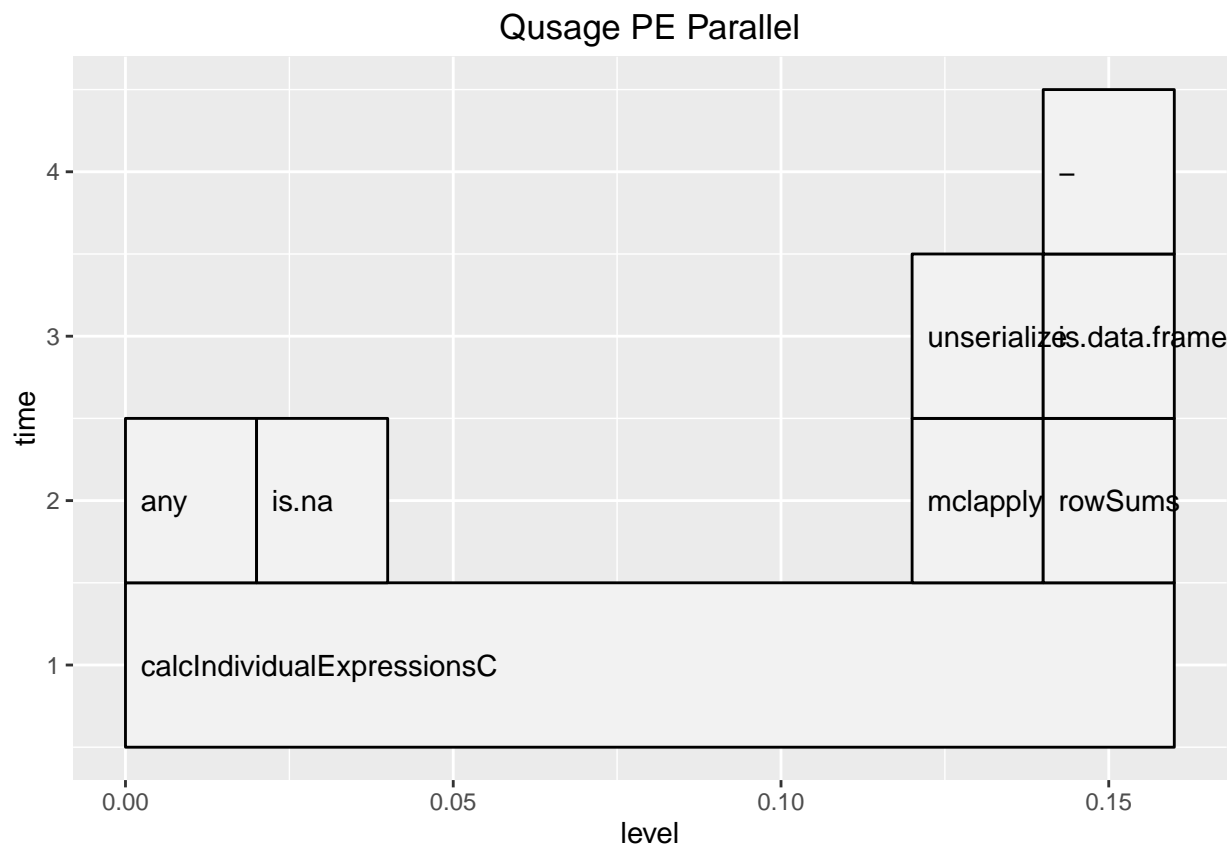
```
## FALSE 4
```

```
## TRUE 5
```

```
require(profr)
require(ggplot2)
y1<-profr(calcIndividualExpressions(Baseline,PostTreatment,paired=TRUE,min.variance.factor=10^-6,na.rm=TRUE))
y2<-profr(calcIndividualExpressionsC(Baseline,PostTreatment,paired=TRUE,min.variance.factor=10^-6))
ggplot(y1)+labs(title="Qusage PE Default")
```



```
ggplot(y2)+labs(title="Qusage PE Parallel")
```



```
#this shows that the only difference is the vector of Non-NA columns per each row; which is the same as
peMB<-microbenchmark(
testPE1<-calcIndividualExpressions(Baseline,PostTreatment,paired=TRUE,min.variance.factor=10^-6,na.rm=TRUE)
testPE2<-calcIndividualExpressionsC(Baseline,PostTreatment,paired=TRUE,min.variance.factor=10^-6)
) #for paired end 1.2X faster
peMB
```

```
## Unit: milliseconds
##
## testPE1 <- calcIndividualExpressions(Baseline, PostTreatment,      paired = TRUE, min.variance.factor=10^-6, na.rm=TRUE)
## testPE2 <- calcIndividualExpressionsC(Baseline, PostTreatment,      paired = TRUE, min.variance.factor=10^-6, na.rm=TRUE)
##      min      lq      mean      median      uq      max neval cld
## 151.7579 155.0447 186.2354 165.0879 220.9855 227.7695   100   b
## 137.9902 145.8967 173.7131 155.2386 208.8553 254.3968   100   a
```

```
#add NAs and test
testPT<-PostTreatment[1:20,]
testPT<-cbind(rbind(testPT,NaN),NA)
rownames(testPT)[nrow(testPT)]<-"NA"
testB<-Baseline[1:20,]
testB<-cbind(rbind(testB,NaN),NA)
rownames(testB)[nrow(testB)]<-"NA"
#calcIndividualExpressionsC(testB,testPT)) will produce error and stop if NA
```