Qusage: Speeding in Parallel

Timothy J. Triche, Jr, Anthony R. Colombo

10 February, 2016

Contents

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. this test data is from the vignette where postTreatment was modified to be Baseline+40, a simple training set.

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

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

```
library(qusage)
```

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

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

```
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)
ISG.geneSet<-ISG.geneSet[grepl("DER_IFN_GAMMA_RESPONSE_UP",names(ISG.geneSet))]
Baseline<-eset
PostTreatment<-eset+20.4
#non-paired
sourceCpp(file="/home/anthonycolombo/Documents/qusage/qusage_repos/qusage_speed/R/sigmasCpp.cpp")
test1<-calcIndividualExpressions(Baseline,PostTreatment,paired=FALSE,min.variance.factor=10^-6,na.rm=TRU
```

```
## 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)
summary(abs(test2$mean-test1$mean)) #machine error precision
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0       0       0       0
```

```
library(microbenchmark)
mb<-microbenchmark(
test1<-calcIndividualExpressions(Baseline,PostTreatment,paired=FALSE,min.variance.factor=10^-6,na.rm=TRU
test2<-calcIndividualExpressionsC(Baseline,PostTreatment,paired=FALSE,min.variance.factor=10^-6))
mb
```

```
## Unit: milliseconds
##
##  test1 <- calcIndividualExpressions(Baseline, PostTreatment, paired = FALSE,      min.variance.factor
##               test2 <- calcIndividualExpressionsC(Baseline, PostTreatment,      paired = FALSE, min.v
##       min       lq     mean   median       uq      max neval cld
##  172.9217 176.8298 191.9751 179.9153 194.5312 249.8955   100   b
##  120.4069 122.9060 128.9261 124.8503 127.9143 182.1017   100  a
```

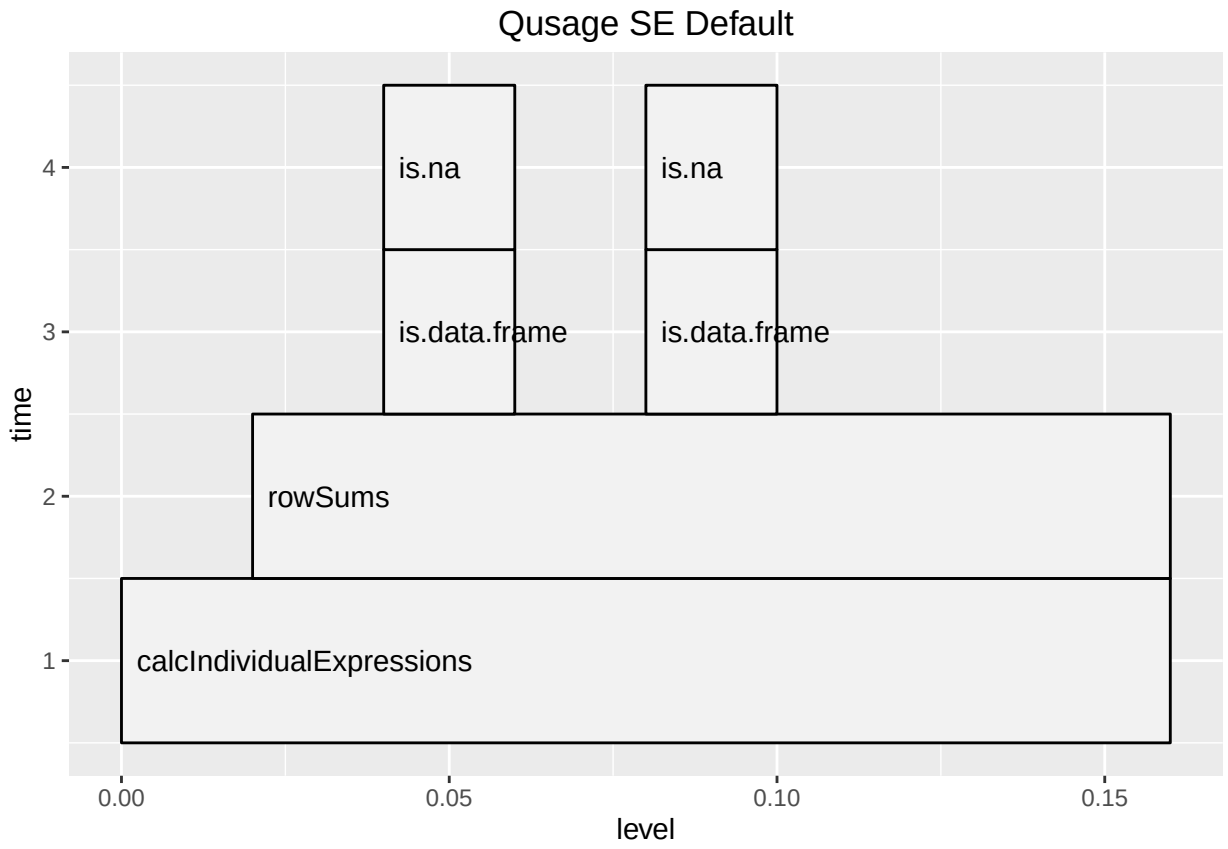
```
require(profr)
```

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

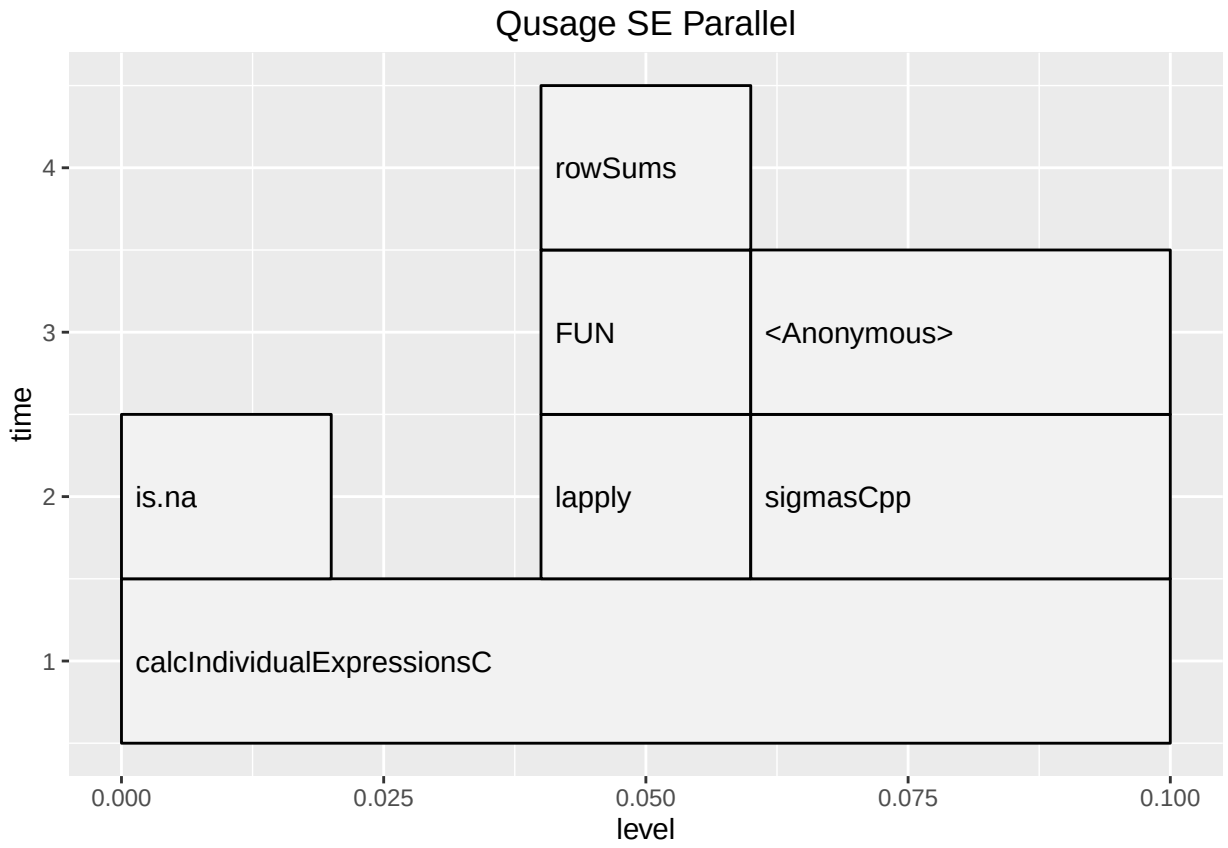
```
require(ggplot2)
```

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

```
x1<-profr(calcIndividualExpressions(Baseline,PostTreatment,paired=FALSE,min.variance.factor=10^-6,na.rm=
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")
```



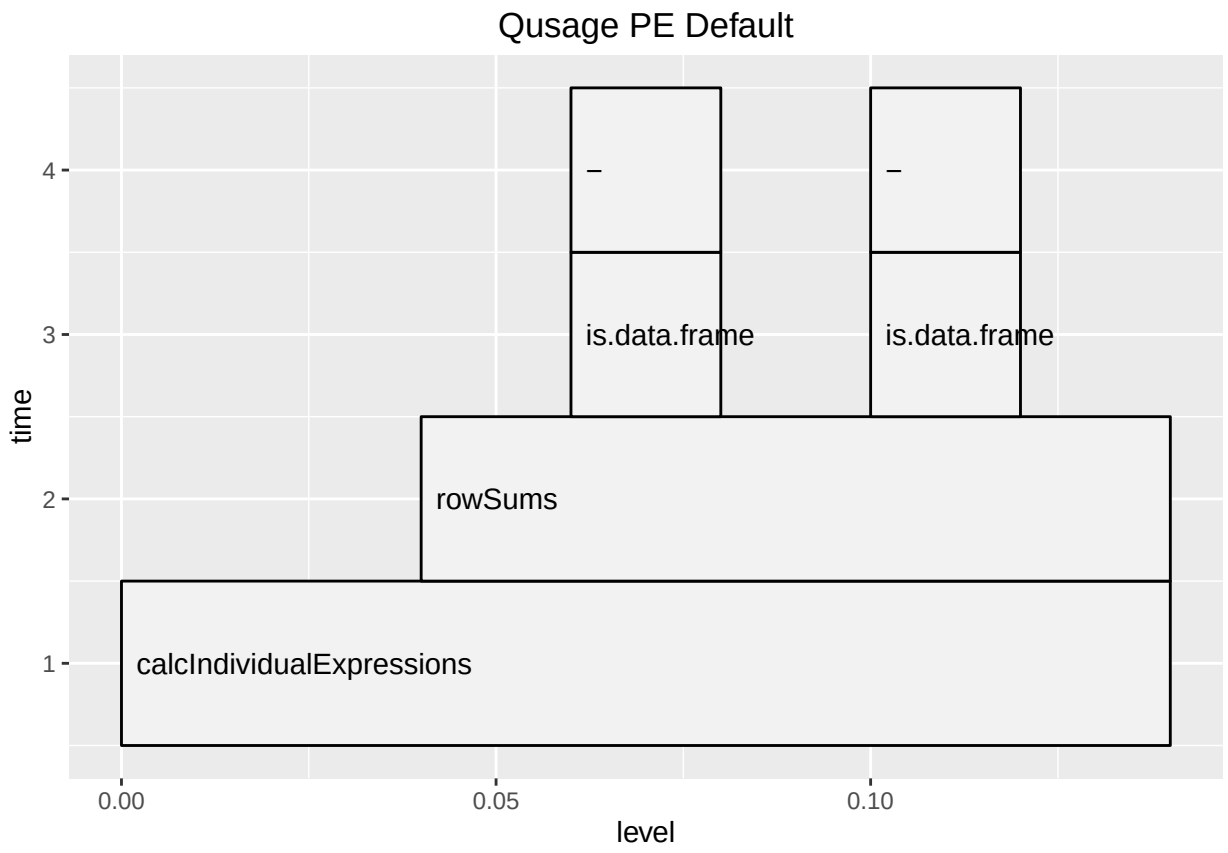

```
#paired end testing
testPE1<-calcIndividualExpressions(Baseline,PostTreatment,paired=TRUE,min.variance.factor=10^-6,na.rm=T
testPE2<-calcIndividualExpressionsC(Baseline,PostTreatment,paired=TRUE,min.variance.factor=10^-6)
for(i in 1:length(test1)){
message(paste0(identical(testPE1[[i]],testPE2[[i]]),"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

```
## TRUE 1

## FALSE 2

## FALSE 3

## FALSE 4

## TRUE 5
```

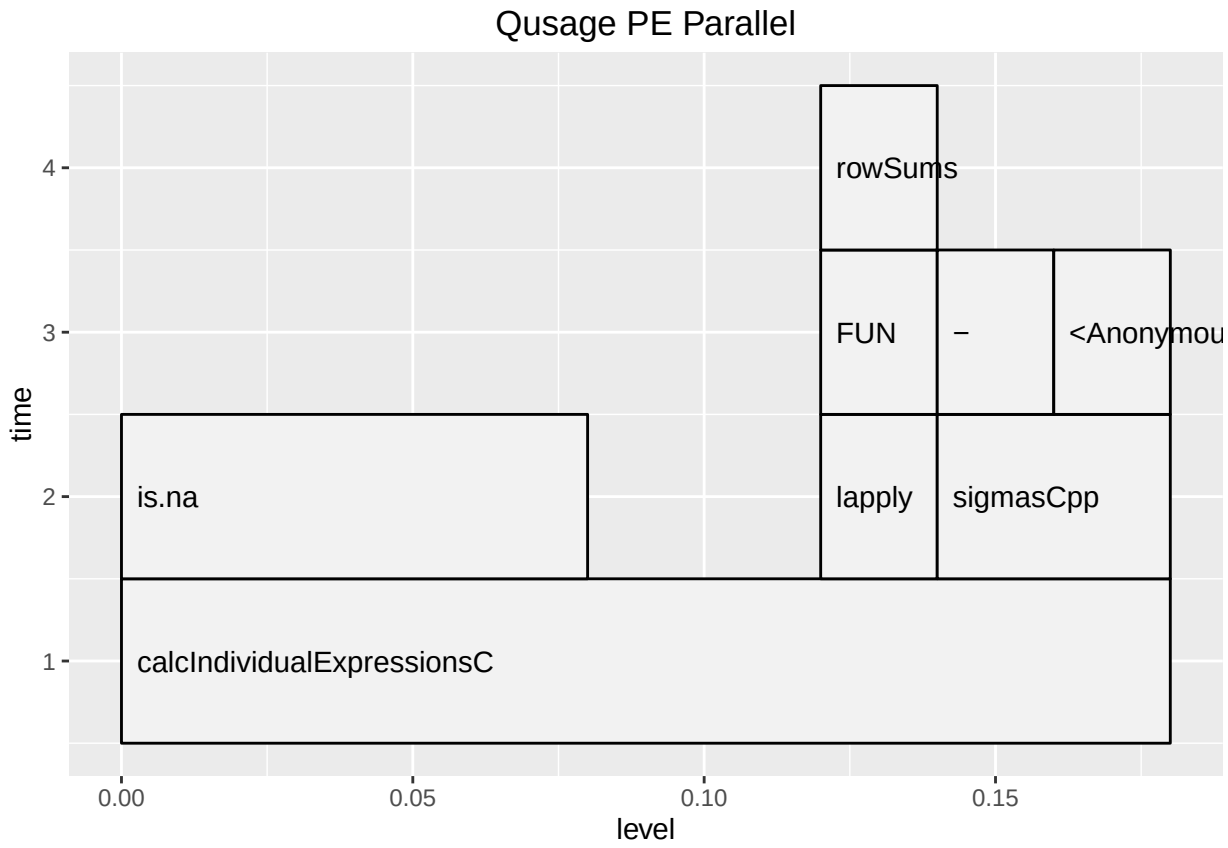
```
summary(abs(testPE1$mean-testPE2$mean))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0       0       0       0
```

```
require(profr)
require(ggplot2)
y1<-profr(calcIndividualExpressions(Baseline,PostTreatment,paired=TRUE,min.variance.factor=10^-6,na.rm=
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=T
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.facto
##               testPE2 <- calcIndividualExpressionsC(Baseline, PostTreatment,      paired = TRUE, min
##       min       lq     mean   median       uq      max neval cld
##  142.0597 145.6209 174.7606 151.7962 205.7013 262.8261   100   b
##  123.1765 126.8468 141.7542 130.3627 135.4670 199.1003   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
```

3 Issue with smaller sets

there is an issue when calling makeComparisons on eset.1 and eset.2 test object, the mclapply is dispatching twice which causes slowness, also I wish to compile R computations for certain functions to speed up before run-time. This eset was then created from makeCompairson funciton which compares two different labels after splitting the eset by column names label type.

4 Paired end revised demo set , not split by label

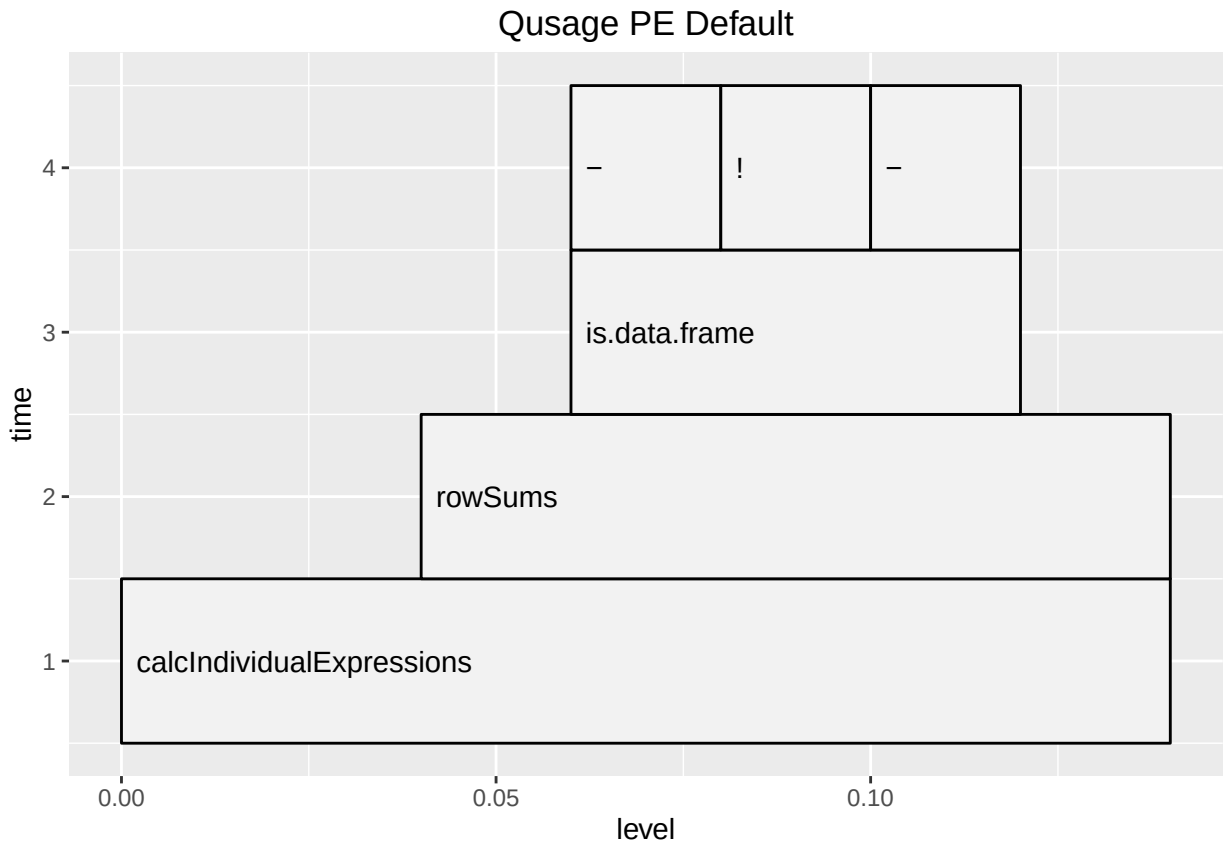
```
library(Rcpp)
library(parallel)
library(speedSage)
library(qusage)
eset<-system.file("extdata","eset.RData",package="speedSage")
load(eset)
labels<-c(rep("t0",134),rep("t1",134))
contrast<-"t1-t0"
colnames(eset)<-c(rep("t0",134),rep("t1",134))
fileISG<-system.file("extdata","c2.cgp.v5.1.symbols.gmt",package="speedSage")
ISG.geneSet<-read.gmt(fileISG)
ISG.geneSet<-ISG.geneSet[grepl("DER_IFN_GAMMA_RESPONSE_UP",names(ISG.geneSet))]
sourceCpp(file="/home/anthonycolombo/Documents/qusage/qusage_repos/qusage_speed/R/sigmasCpp.cpp")
eset.1<-eset-40.3
eset.2<-eset+100.5
original<-calcIndividualExpressions(eset.1,eset.2,paired=TRUE)
cpp<-calcIndividualExpressionsC(eset.1,eset.2,paired=TRUE)
summary(abs(original$mean-cpp$mean)) #identical results
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0       0       0       0
```

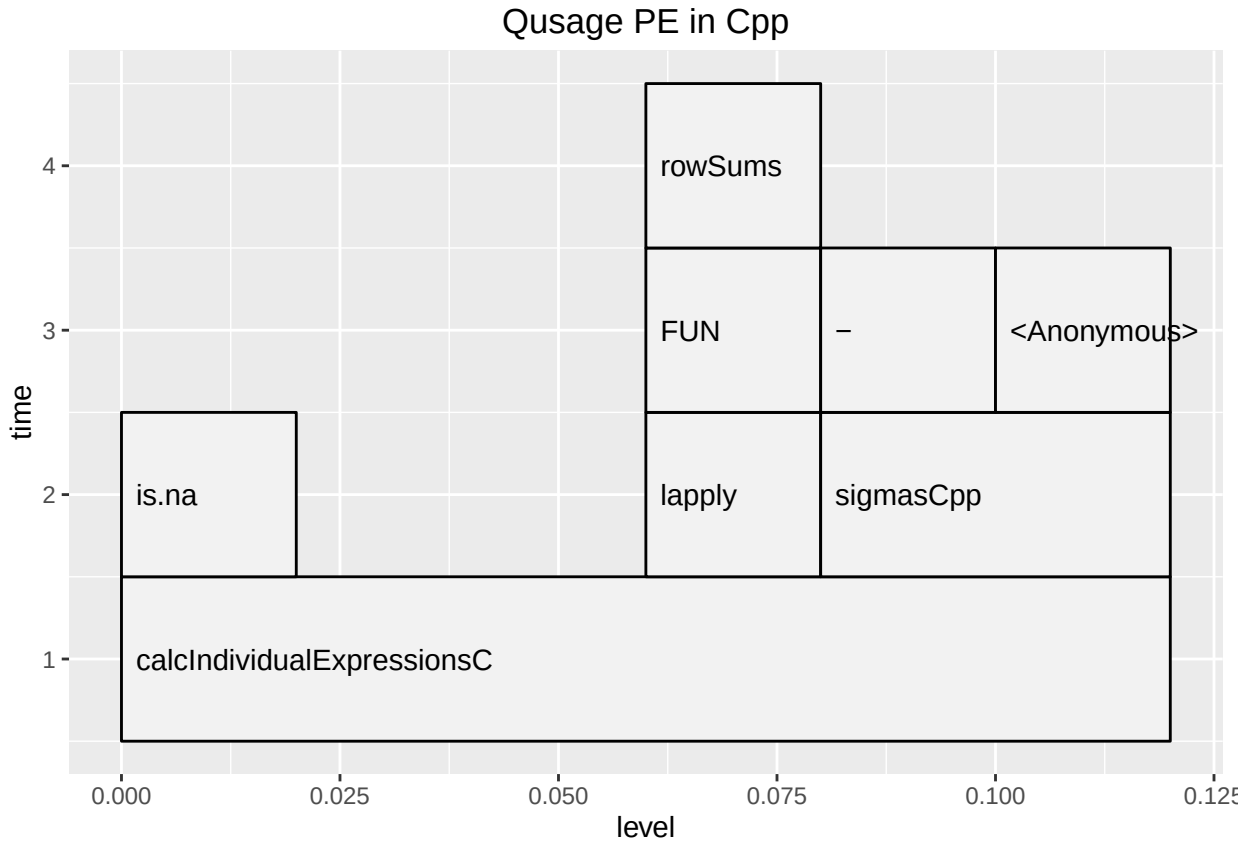
```
microbenchmark(
 original<-calcIndividualExpressions(eset.1,eset.2,paired=TRUE),
 cpp<-calcIndividualExpressionsC(eset.1,eset.2,paired=TRUE))
```

```
## Unit: milliseconds
##                                                                  expr
##  original <- calcIndividualExpressions(eset.1, eset.2, paired = TRUE)
##      cpp <- calcIndividualExpressionsC(eset.1, eset.2, paired = TRUE)
##       min       lq     mean   median       uq      max neval cld
##  140.5466 146.5559 157.0399 149.8408 154.7405 216.2175   100   b
##  123.3946 131.0393 138.8488 133.7987 138.0690 248.0172   100  a
```

```
#showing profiles
library(profr)
library(ggplot2)

yy<-profr(calcIndividualExpressions(eset.1,eset.2,paired=TRUE))
ggplot(yy) + labs(title="Qusage PE Default")
```




```
tt<-profr(calcIndividualExpressionsC(eset.1,eset.2,paired=TRUE))
ggplot(tt)+ labs(title="Qusage PE in Cpp")
```



5 Non-paired end the eset.1, eset.2 split by label

This simulates how makeComparison will compare a split eset with label split

```
library(microbenchmark)
library(profr)
library(ggplot2)
library(Rcpp)
eset.1<-system.file("extdata","eset.1.RData",package="speedSage")
eset.2<-system.file("extdata","eset.2.RData",package="speedSage")
load(eset.1)
load(eset.2)
sourceCpp(file="/home/anthonycolombo/Documents/qusage/qusage_repos/qusage_speed/R/sigmasCpp.cpp")

original<-calcIndividualExpressions(eset.1,eset.2,paired=FALSE)
cpp<-calcIndividualExpressionsC(eset.1,eset.2,paired=FALSE)
summary(abs(original$mean-cpp$mean))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0       0       0       0
```

```
summary(abs(original$SD-cpp$SD))
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.000e+00 0.000e+00 0.000e+00 3.844e-18 6.939e-18 5.551e-17
```

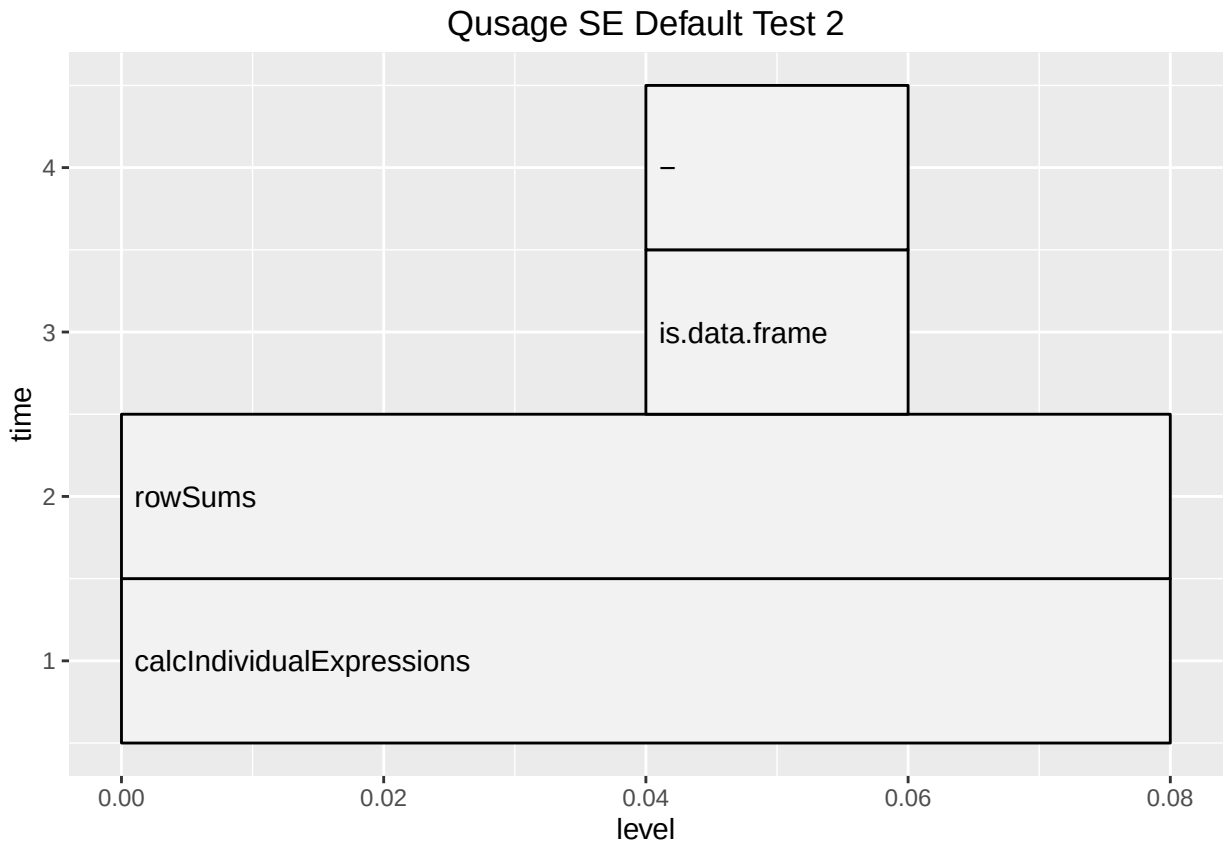
```
summary(abs(original$dof-cpp$dof))
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.000e+00 0.000e+00 2.842e-14 4.103e-14 5.684e-14 2.274e-13
```

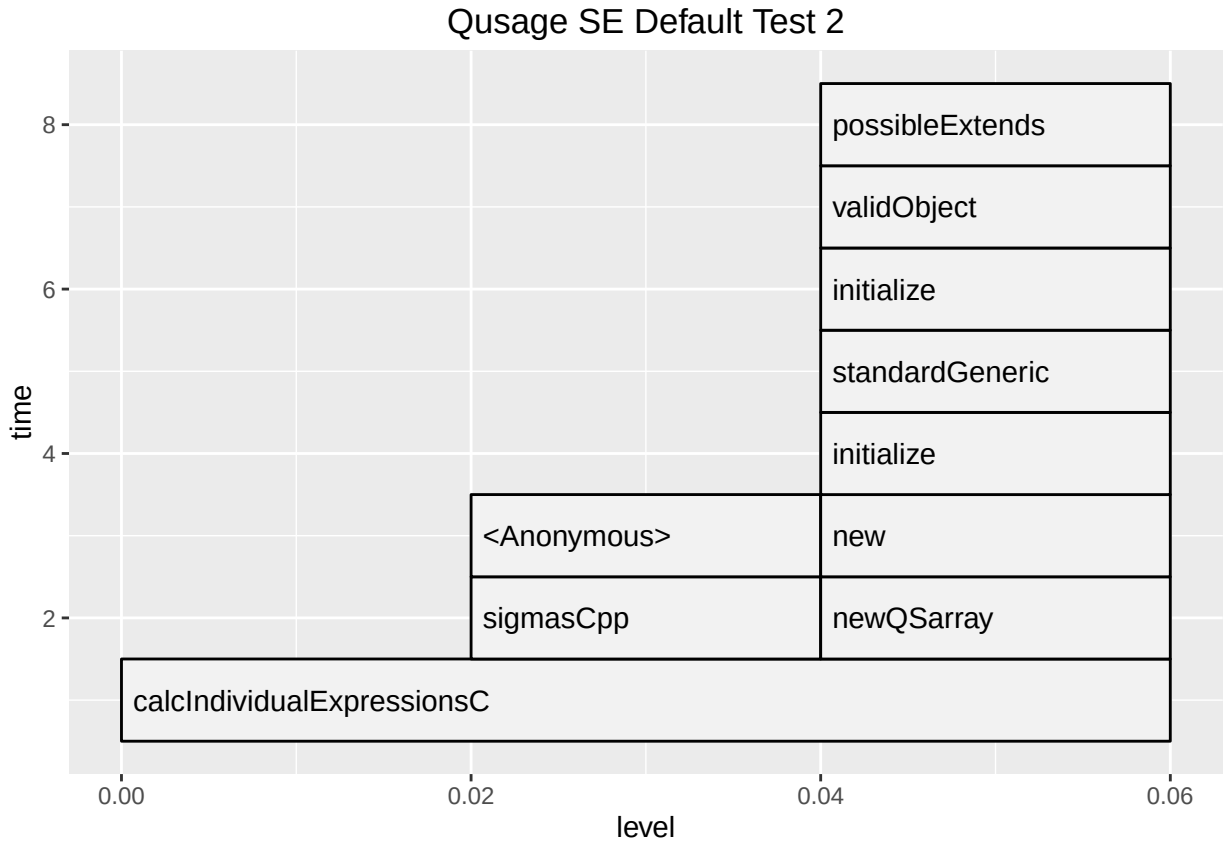
```
microbenchmark(
 original<-calcIndividualExpressions(eset.1,eset.2,paired=FALSE),
 cpp<-calcIndividualExpressionsC(eset.1,eset.2,paired=FALSE))
```

```
## Unit: milliseconds
##                                                                    expr
##  original <- calcIndividualExpressions(eset.1, eset.2, paired = FALSE)
##      cpp <- calcIndividualExpressionsC(eset.1, eset.2, paired = FALSE)
##       min       lq     mean   median       uq      max neval cld
##  87.36843 88.37343 90.77532 89.57440 90.51210 146.62017   100   b
##  61.92054 62.42676 63.59162 63.06735 64.45717  67.38336   100  a
```

```
x<-profr(calcIndividualExpressions(eset.1,eset.2,paired=FALSE))
y<-profr(calcIndividualExpressionsC(eset.1,eset.2,paired=FALSE))
ggplot(x) + labs(title="Qusage SE Default Test 2")
```




```
ggplot(y) + labs(title="Qusage SE Default Test 2")
```


Qusage SE Default Test 2
time
level
8
6
4
2
0.00
0.02
0.04
0.06
possibleExtends
validObject
initialize
standardGeneric
initialize
<Anonymous>
new
sigmasCpp
newQSarray
calcIndividualExpressionsC