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.

```
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)
identical(test2,test1)
```

```
## [1] FALSE
```

```
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))
#on average 1.49X faster
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
##  169.6035 172.4646 187.9819 176.0636 222.4185 231.5290   100   b
##  167.3527 170.0432 180.7431 173.3693 176.5717 230.4256   100  a
```

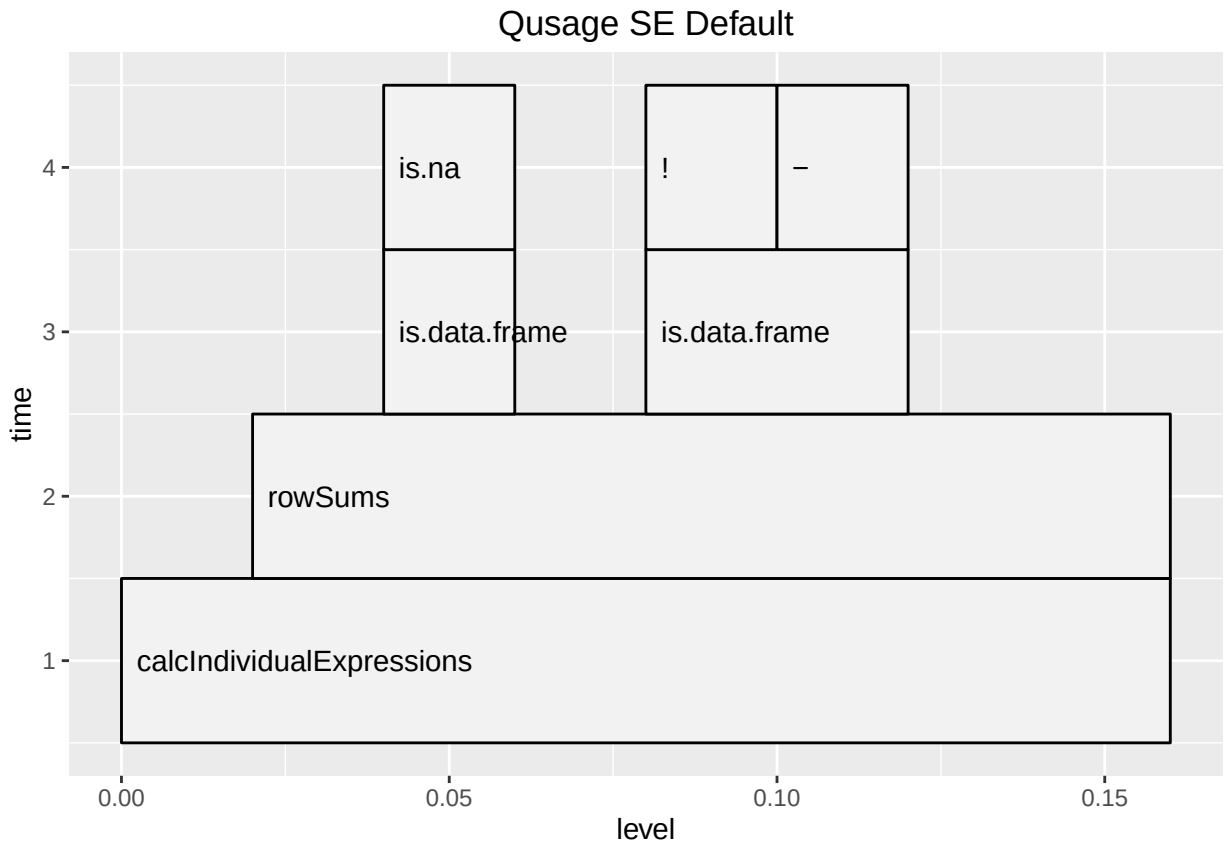
```
require(profr)
```

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

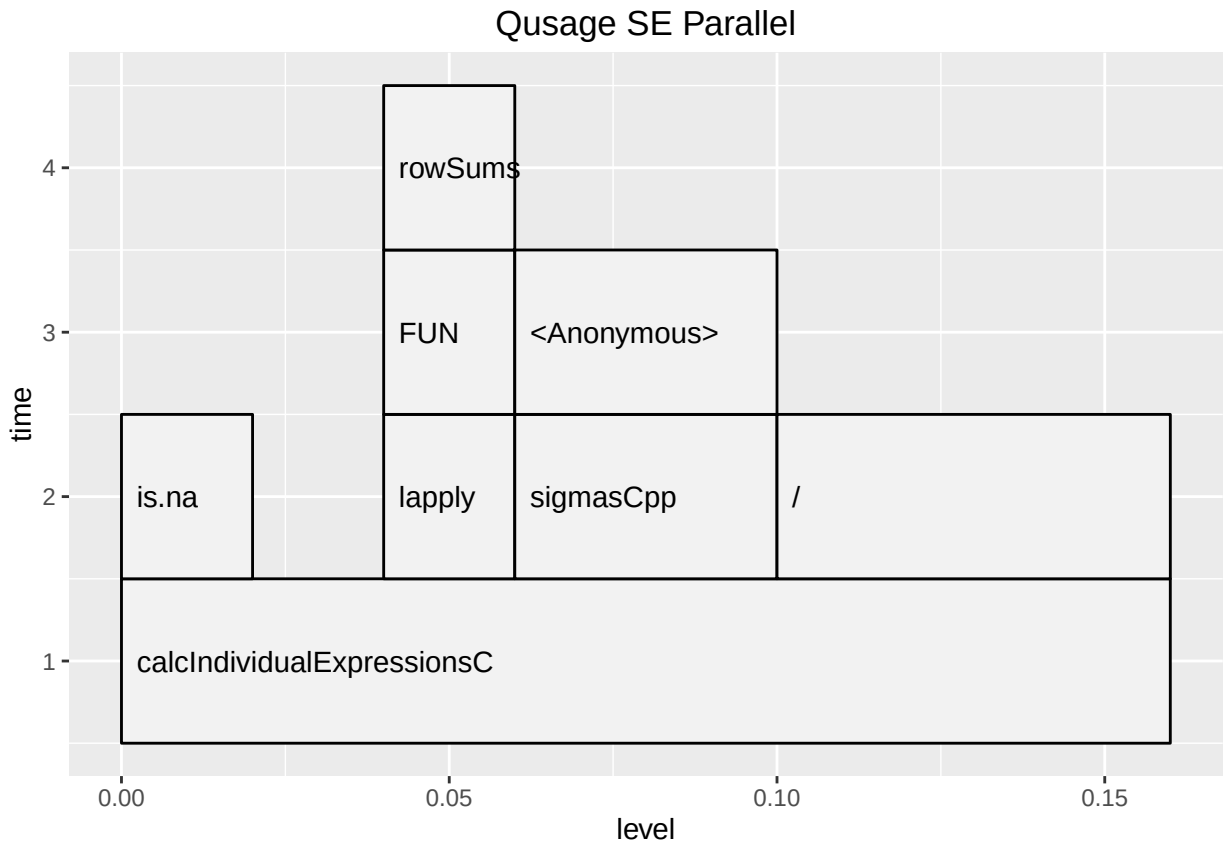
```
require(ggplot2)
```

```
## 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")
```



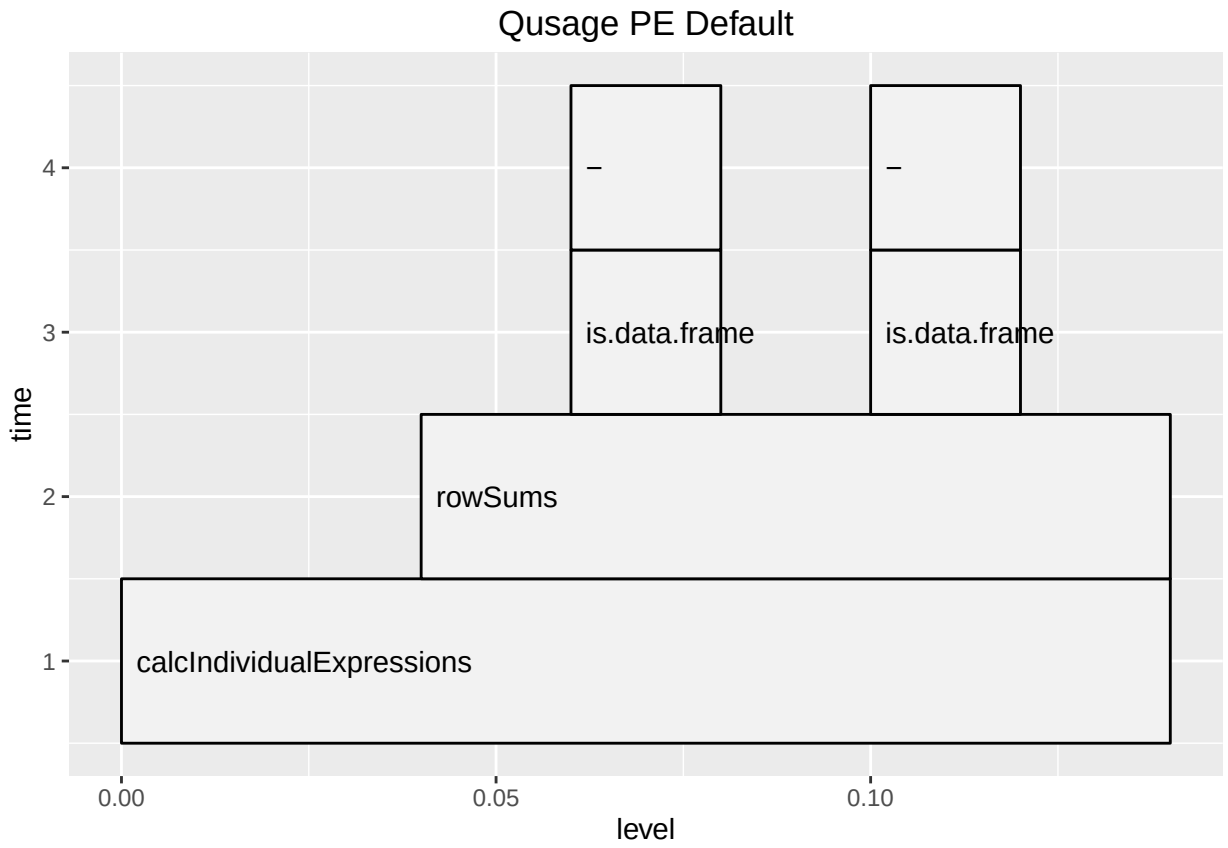

```
x2<-profr(calcIndividualExpressionsC(Baseline,PostTreatment,paired=FALSE,min.variance.factor=10^-6))
ggplot(x2)+labs(title="Qusage SE Parallel")
```



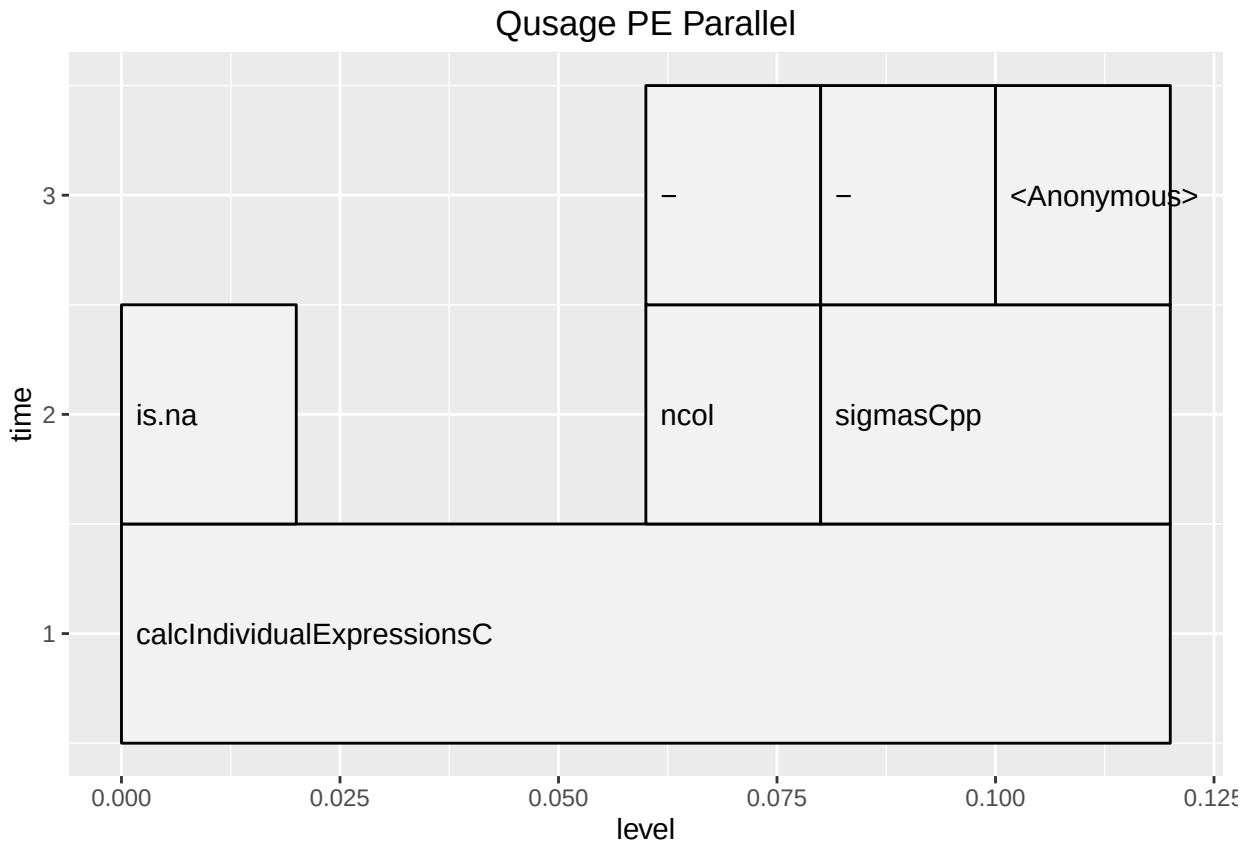

```
#paired end testing
testPE1<-calcIndividualExpressions(Baseline,PostTreatment,paired=TRUE,min.variance.factor=10^-6,na.rm=T
testPE2<-calcIndividualExpressionsC(Baseline,PostTreatment,paired=TRUE,min.variance.factor=10^-6)
for(i in 1:length(test1)){
message(paste0(identical(testPE1[[i]],testPE2[[i]]),"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

```
## TRUE 1

## FALSE 2

## FALSE 3

## FALSE 4

## TRUE 5
```

```
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
##  139.8126 141.9204 153.3843 143.6768 146.3989 205.5734   100   b
##  122.8728 125.2726 131.7081 127.0918 129.2121 189.2620   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

4 Paired revised

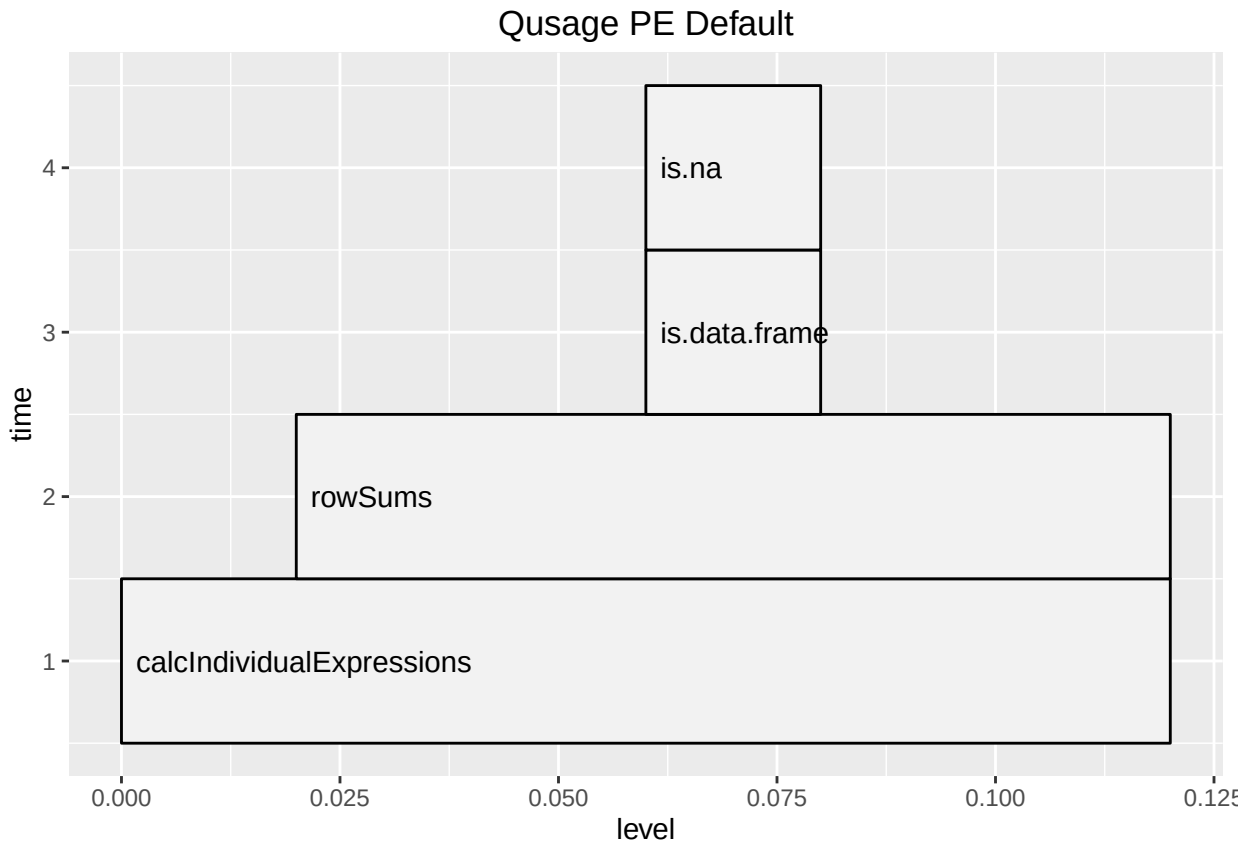
```
library(Rcpp)
library(parallel)
library(speedSage)
library(qusage)
eset<-system.file("extdata","eset.RData",package="speedSage")
load(eset)
labels<-c(rep("t0",134),rep("t1",134))
contrast<-"t1-t0"
colnames(eset)<-c(rep("t0",134),rep("t1",134))
fileISG<-system.file("extdata","c2.cgp.v5.1.symbols.gmt",package="speedSage")
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
```

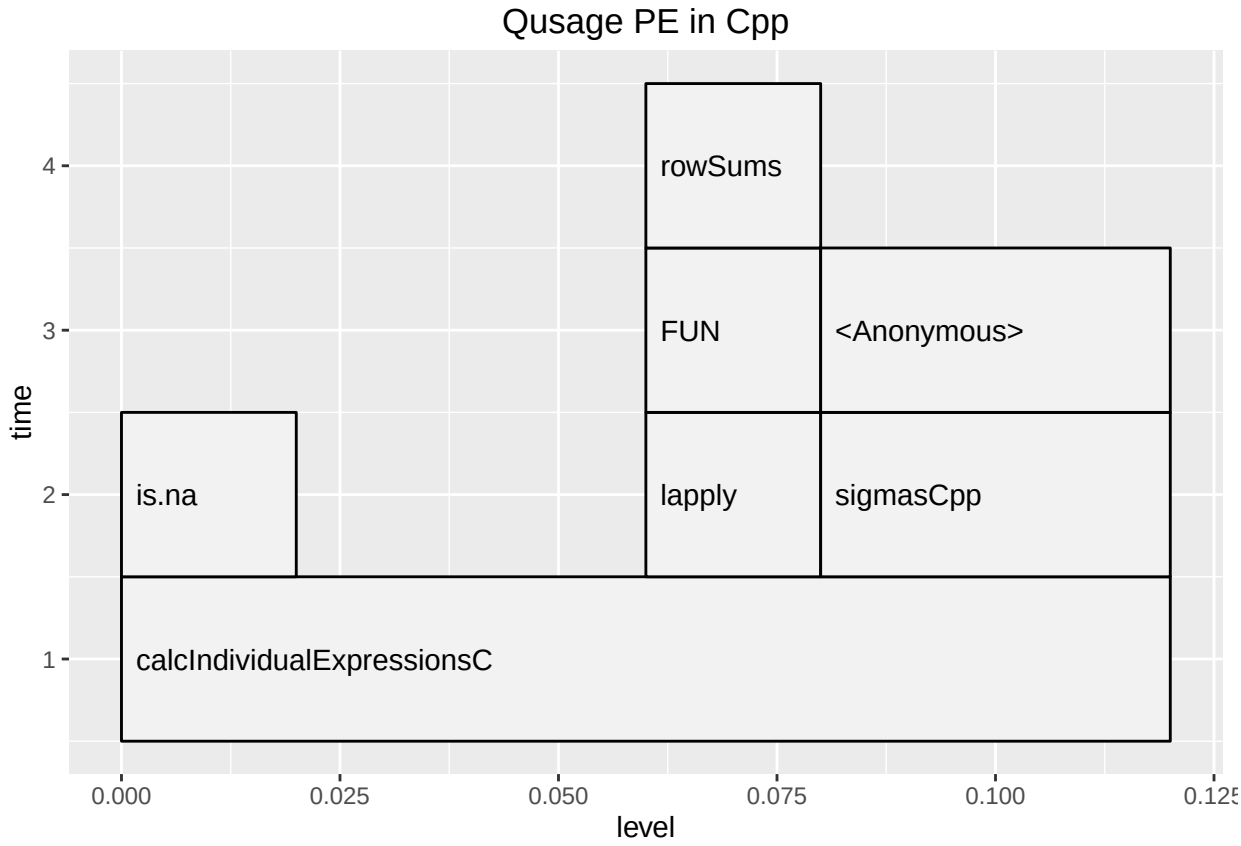
```
microbenchmark(
 original<-calcIndividualExpressions(eset.1,eset.2,paired=TRUE),
 cpp<-calcIndividualExpressionsC(eset.1,eset.2,paired=TRUE))
```

```
## 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
##  139.5709 142.1453 151.2096 144.1039 146.1386 204.9111   100   b
##  122.7501 124.8442 137.9403 127.0056 132.2607 188.3307   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 for non-paired end the eset.1, eset.2 is split

```
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)
cpp<-calcIndividualExpressionsC(eset.1,eset.2)
```