Qusage: Speeding up in RcppArmadillo

Timothy J. Triche, Jr, Anthony R. Colombo

18 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 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.

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+20.4, a simple training set from the QuSAGE vignette.

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

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

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

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

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

```r
library(qusage)
```

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

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

```r
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)
ISG.geneSet<-ISG.geneSet[grepl("DER_IFN_GAMMA_RESPONSE_UP",names(ISG.geneSet))]
Baseline<-eset
PostTreatment<-eset+20.4
ncol(Baseline) #not splitting up eset
```

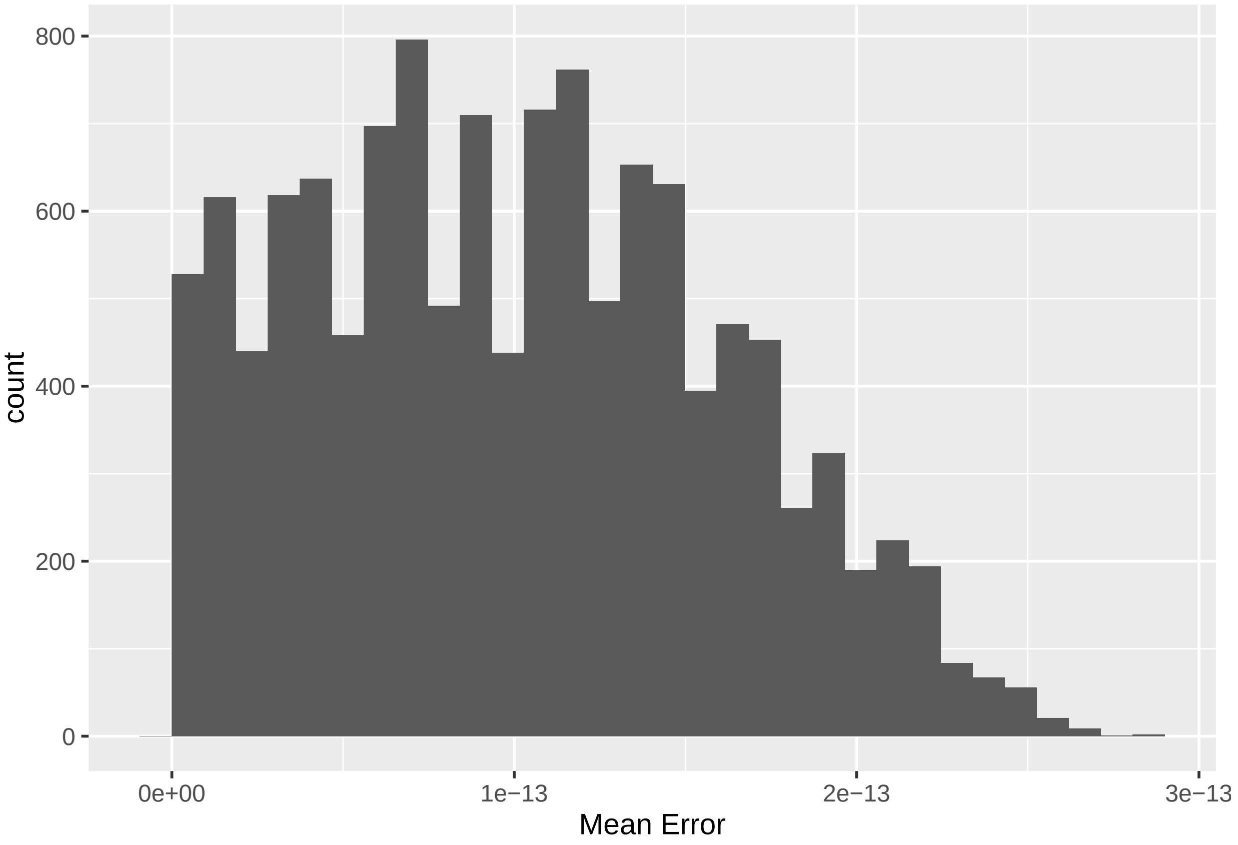
```
## [1] 268
```

```r
#paired
sourceCpp(file="/home/anthonycolombo/Documents/qusage/qusage_repos/qusage_speed/R/sigmaArm.cpp")
sourceCpp(file="/home/anthonycolombo/Documents/qusage/qusage_repos/qusage_speed/R/sigmasCpp.cpp")
test1<-calcIndividualExpressionsArm(Baseline,PostTreatment,paired=TRUE,min.variance.factor=10^-6)
```

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

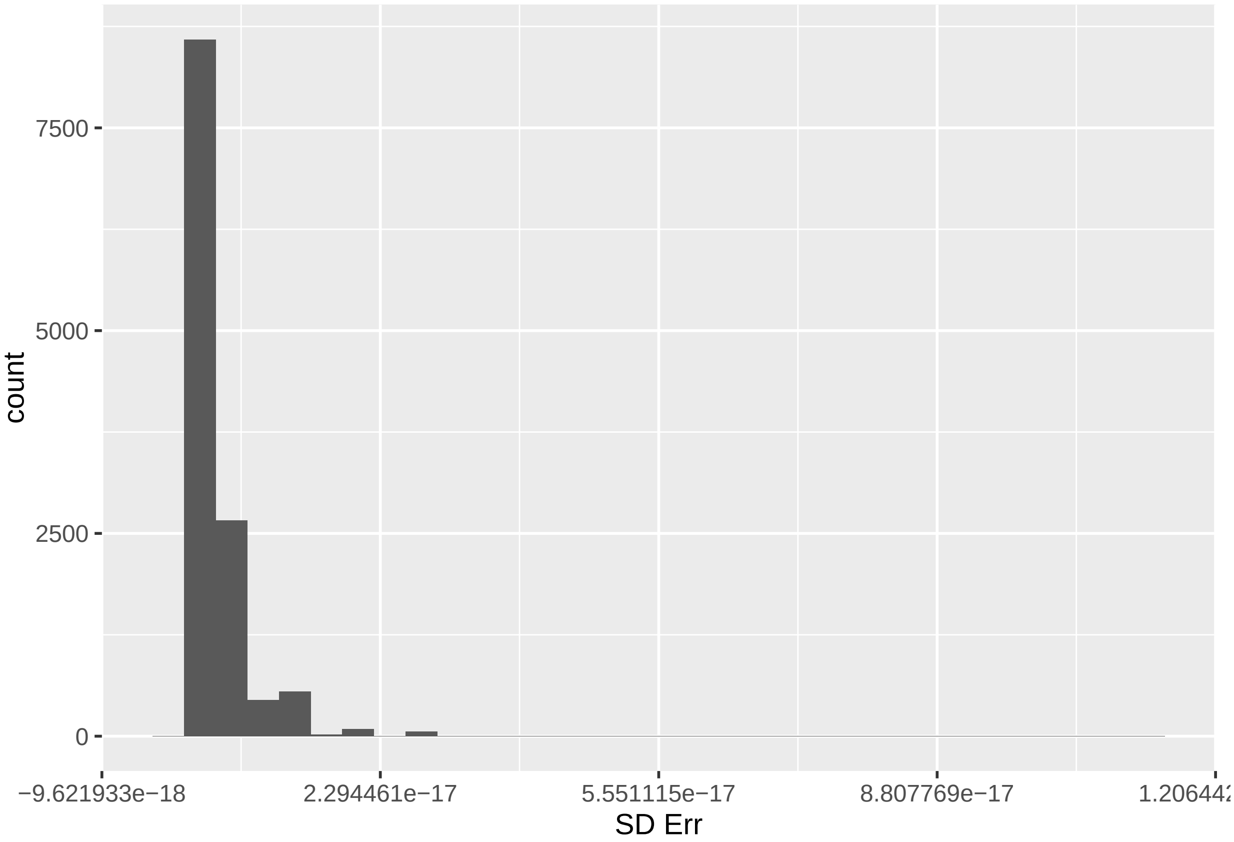
```r
test2<-calcIndividualExpressionsC(Baseline,PostTreatment,paired=TRUE,min.variance.factor=10^-6)
test3<-calcIndividualExpressions(Baseline,PostTreatment,paired=TRUE,min.variance.factor=10^-6,na.rm=TRU
qplot(abs(test1[[1]]-test3[[1]]), xlab="Mean Error")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



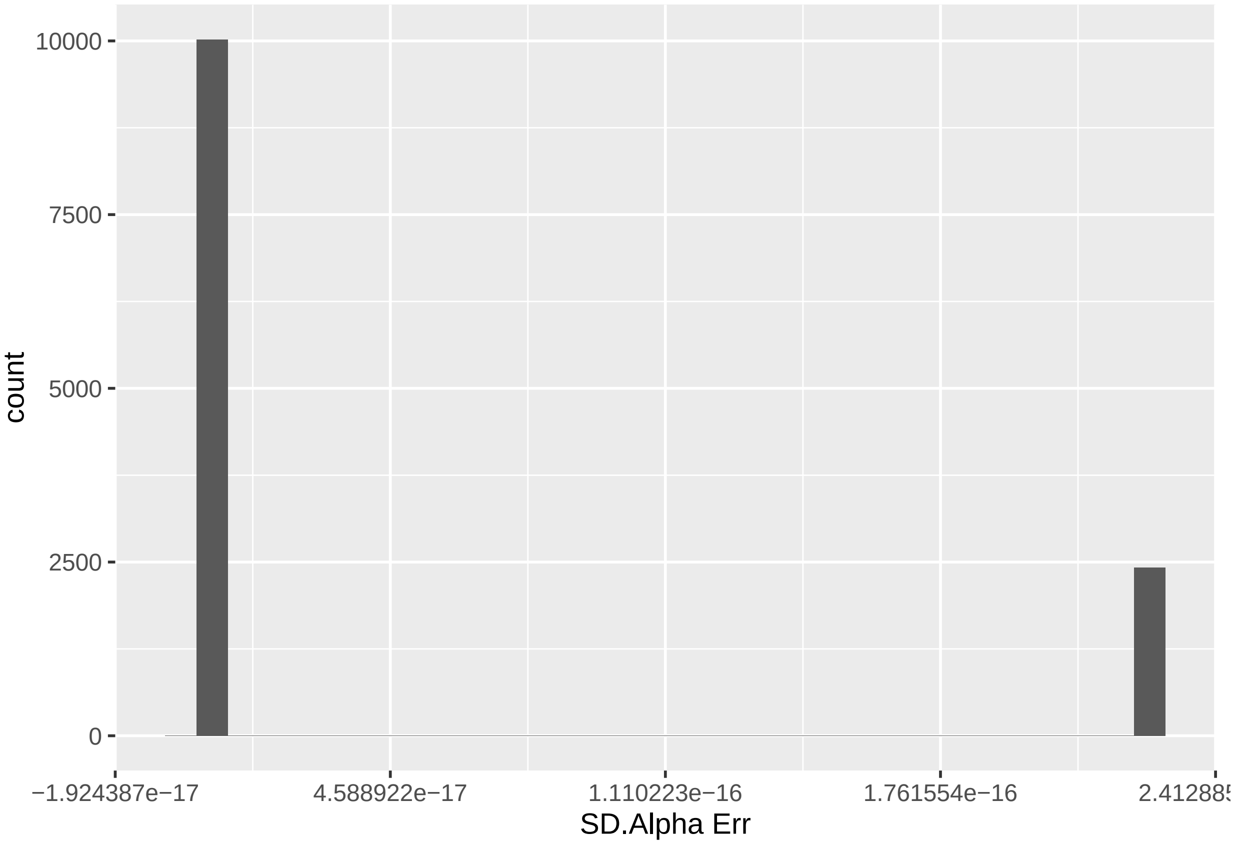
```
qplot(abs(test1[[2]]-test3[[2]]), xlab="SD Err")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



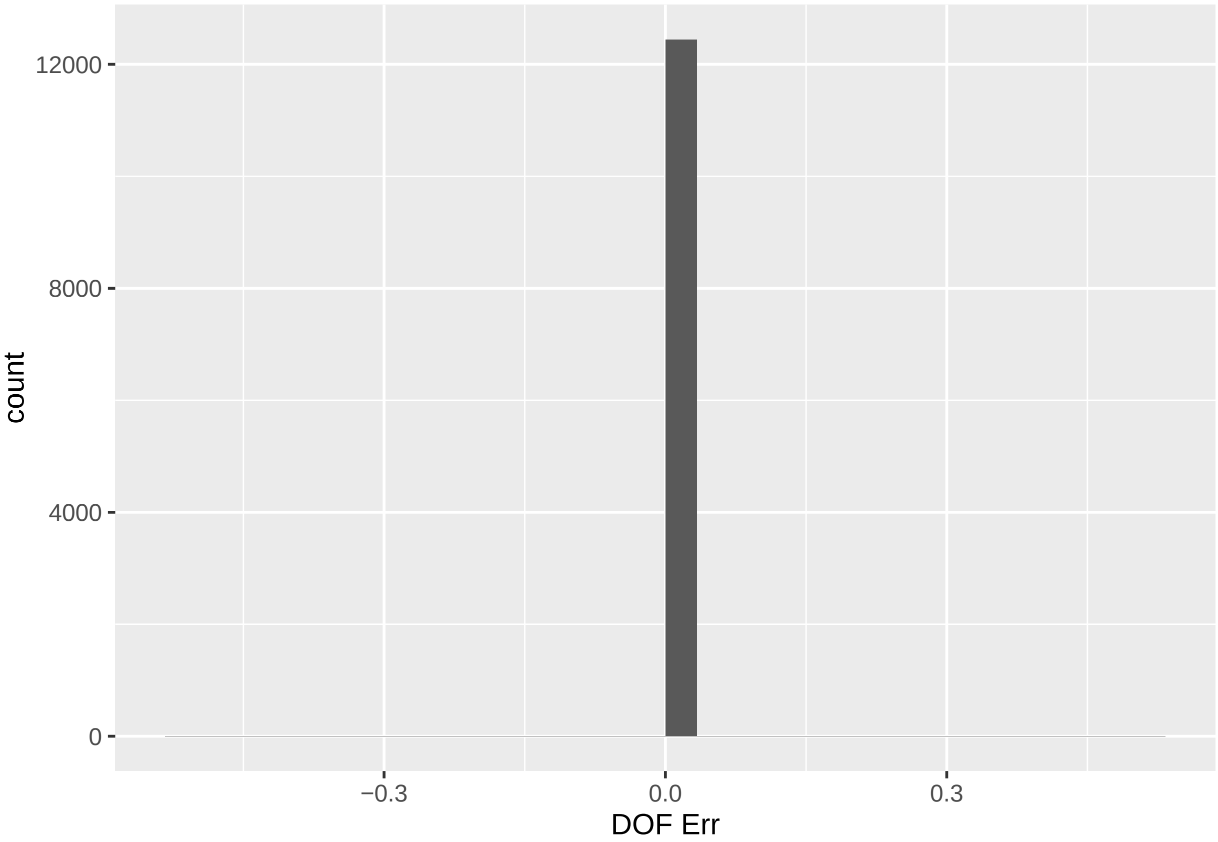
```r
qplot(abs(test1[[3]]-test3[[3]]), xlab="SD.Alpha Err")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
qplot(abs(test1[[4]]-test3[[4]]), xlab="DOF Err")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



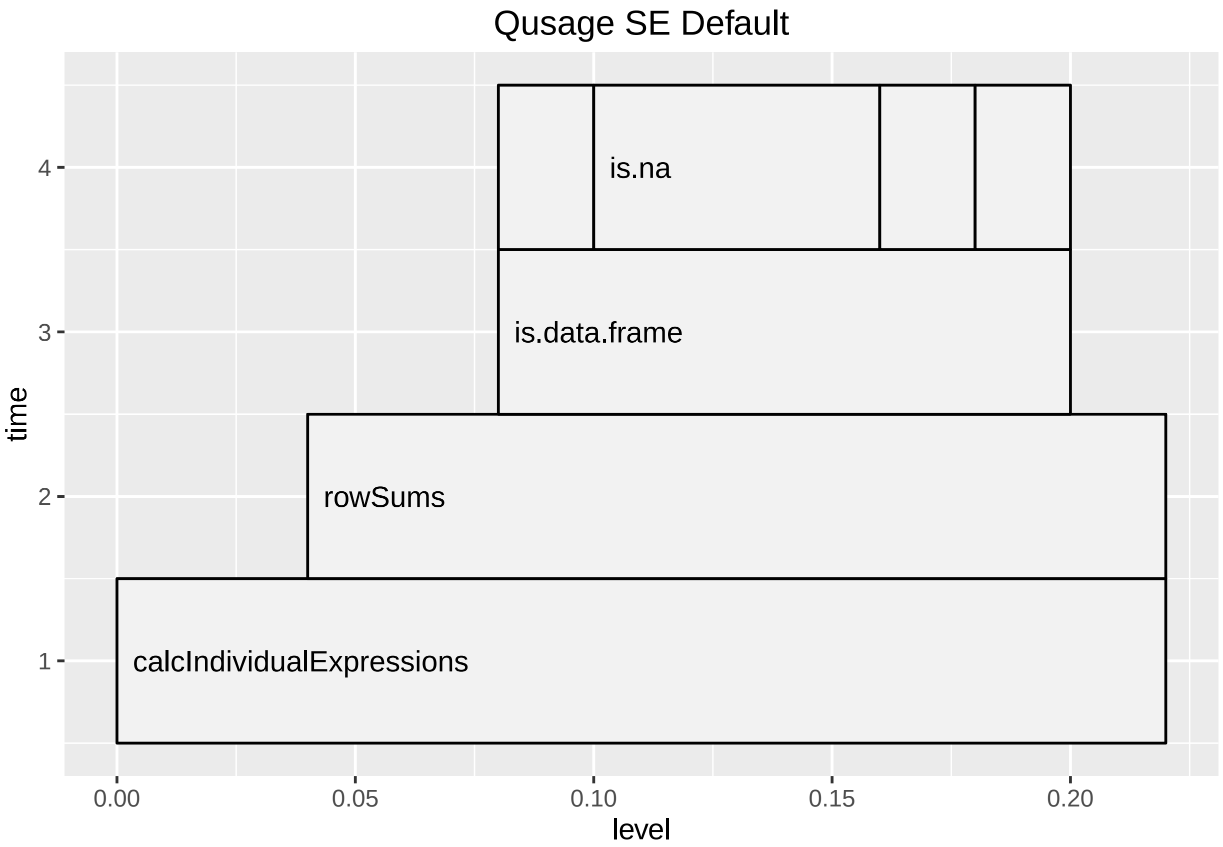
```
mb<-microbenchmark(
test1<-calcIndividualExpressionsArm(Baseline,PostTreatment,paired=TRUE,min.variance.factor=10^-6),
test2<-calcIndividualExpressionsC(Baseline,PostTreatment,paired=TRUE,min.variance.factor=10^-6),
test3<-calcIndividualExpressions(Baseline,PostTreatment,paired=TRUE,min.variance.factor=10^-6,na.rm=TRU
mb
```

```
## Unit: milliseconds
##
##             test1 <- calcIndividualExpressionsArm(Baseline, PostTreatment,      paired = TRUE, min.va
##               test2 <- calcIndividualExpressionsC(Baseline, PostTreatment,      paired = TRUE, min.va
##  test3 <- calcIndividualExpressions(Baseline, PostTreatment, paired = TRUE,      min.variance.factor
##       min        lq     mean    median        uq      max neval cld
##   86.7703  89.21739 100.3838  92.30324  96.18027 156.7902   100 a
##  124.3601 127.67895 141.9754 130.35234 136.05756 191.6604   100  b
##  140.0840 145.85332 166.0265 148.11940 202.16970 247.1822   100   c
```

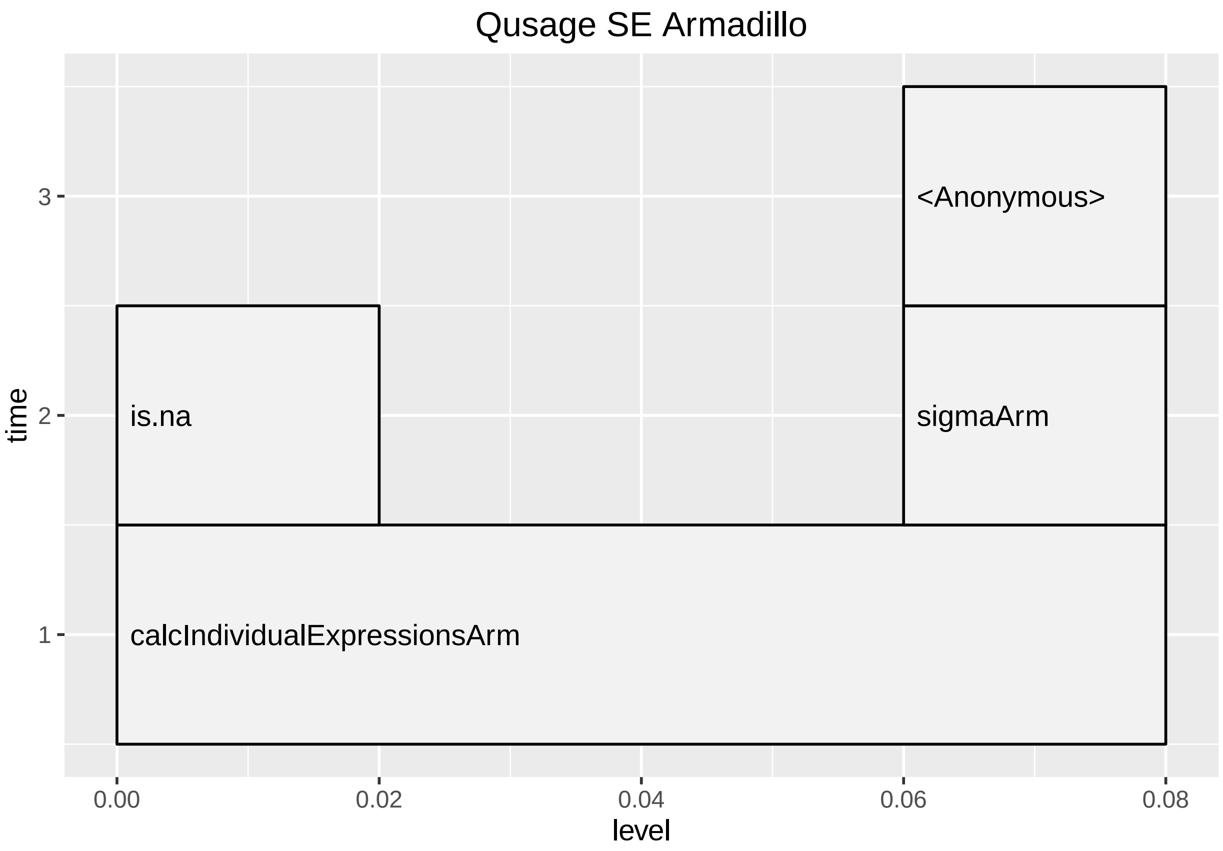
```
require(profr)
```

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

```
require(ggplot2)
x1<-profr(calcIndividualExpressions(Baseline,PostTreatment,paired=TRUE,min.variance.factor=10^-6,na.rm=
ggplot(x1)+labs(title="Qusage SE Default")
```

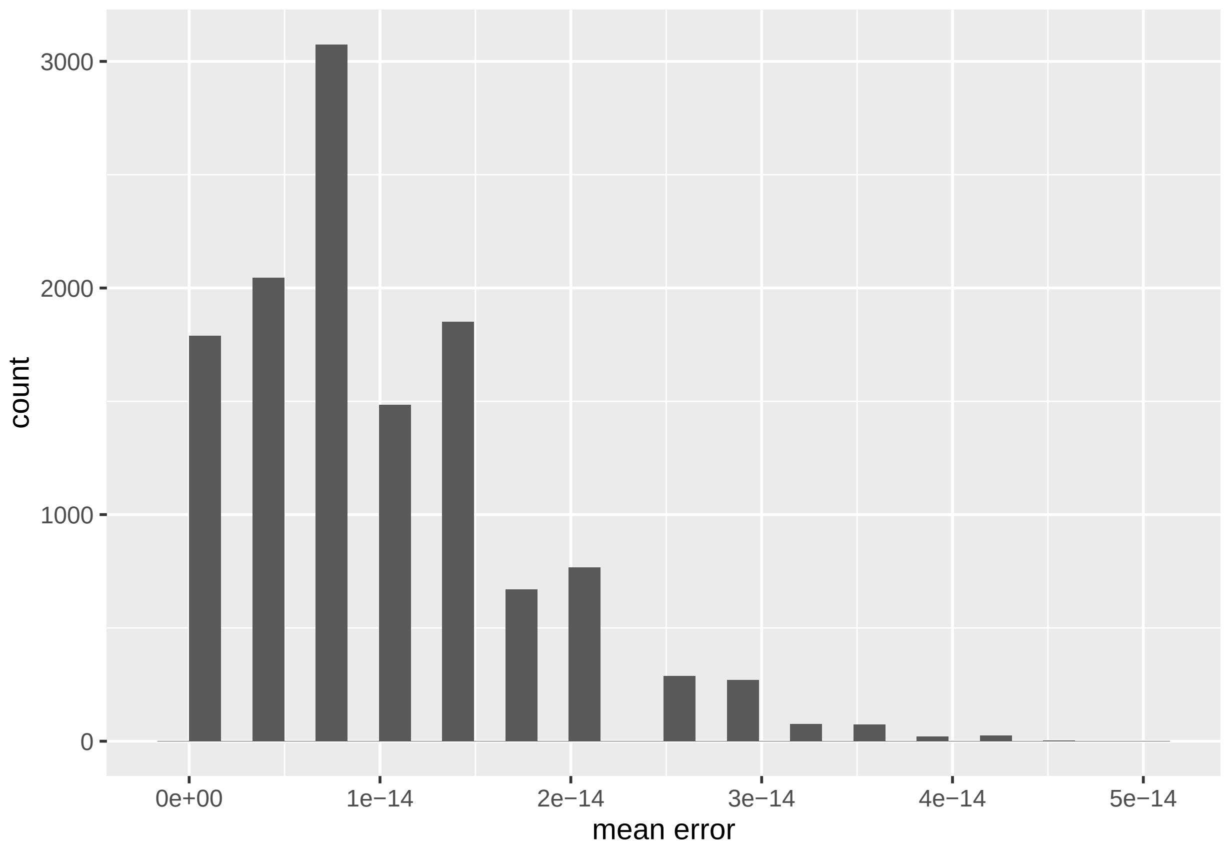


```
x2<-profr(calcIndividualExpressionsArm(Baseline,PostTreatment,paired=TRUE,min.variance.factor=10^-6))
ggplot(x2)+labs(title="Qusage SE Armadillo")
```



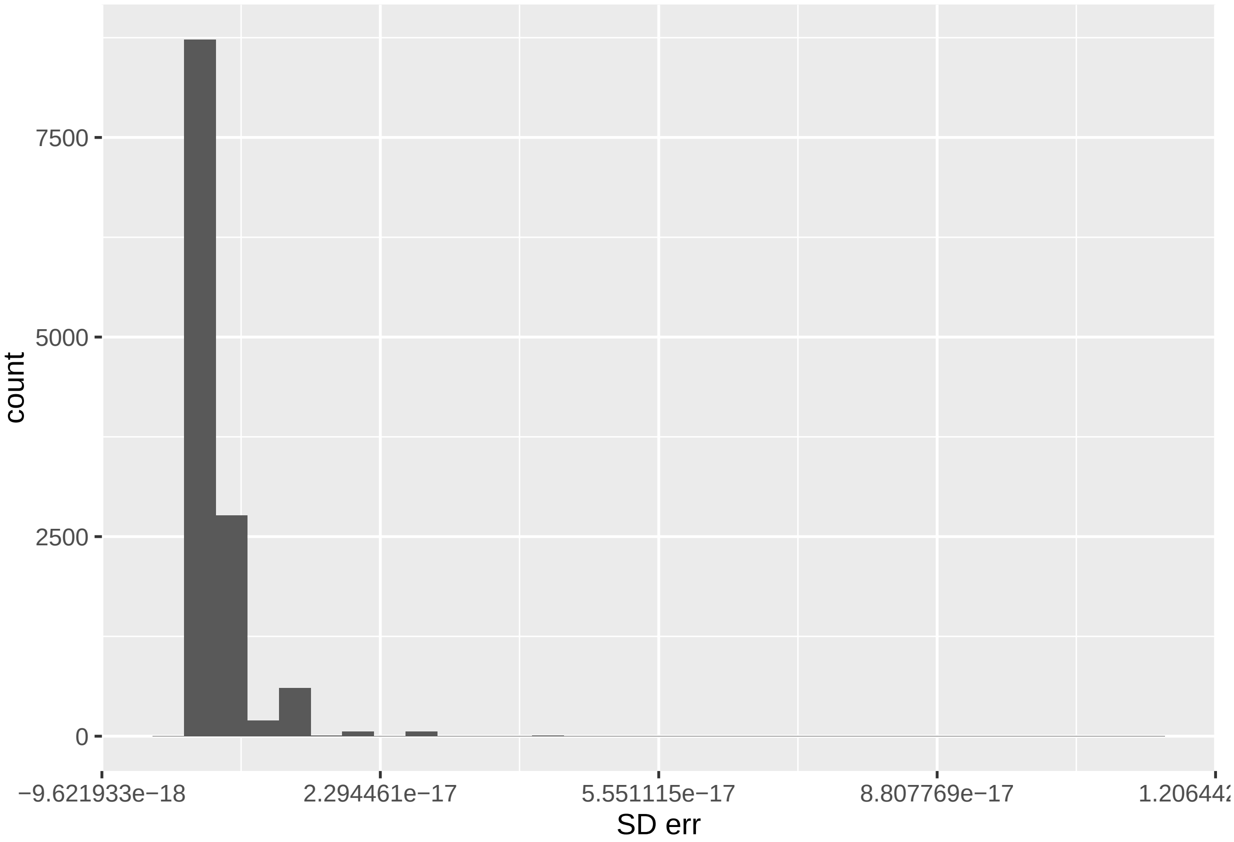
```
#single end testing
sourceCpp("/home/anthonycolombo/Documents/qusage/qusage_repos/qusage_speed/R/sigmaSingle.cpp")
testSE1<-calcIndividualExpressions(Baseline,PostTreatment,paired=FALSE,min.variance.factor=10^-6,na.rm=
testSE2<-calcIndividualExpressionsArm(Baseline,PostTreatment,paired=FALSE,min.variance.factor=10^-6)
testSE3<-calcIndividualExpressionsC(Baseline,PostTreatment,paired=FALSE,min.variance.factor=10^-6)


e1<-(abs(testSE1[[1]]-testSE2[[1]]))
e2<-(abs(testSE1[[2]]-testSE2[[2]]))
e3<-(abs(testSE1[[3]]-testSE2[[3]]))
e4<-(abs(testSE1[[4]]-testSE2[[4]]))
qplot(as.vector(e1), xlab="mean error")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



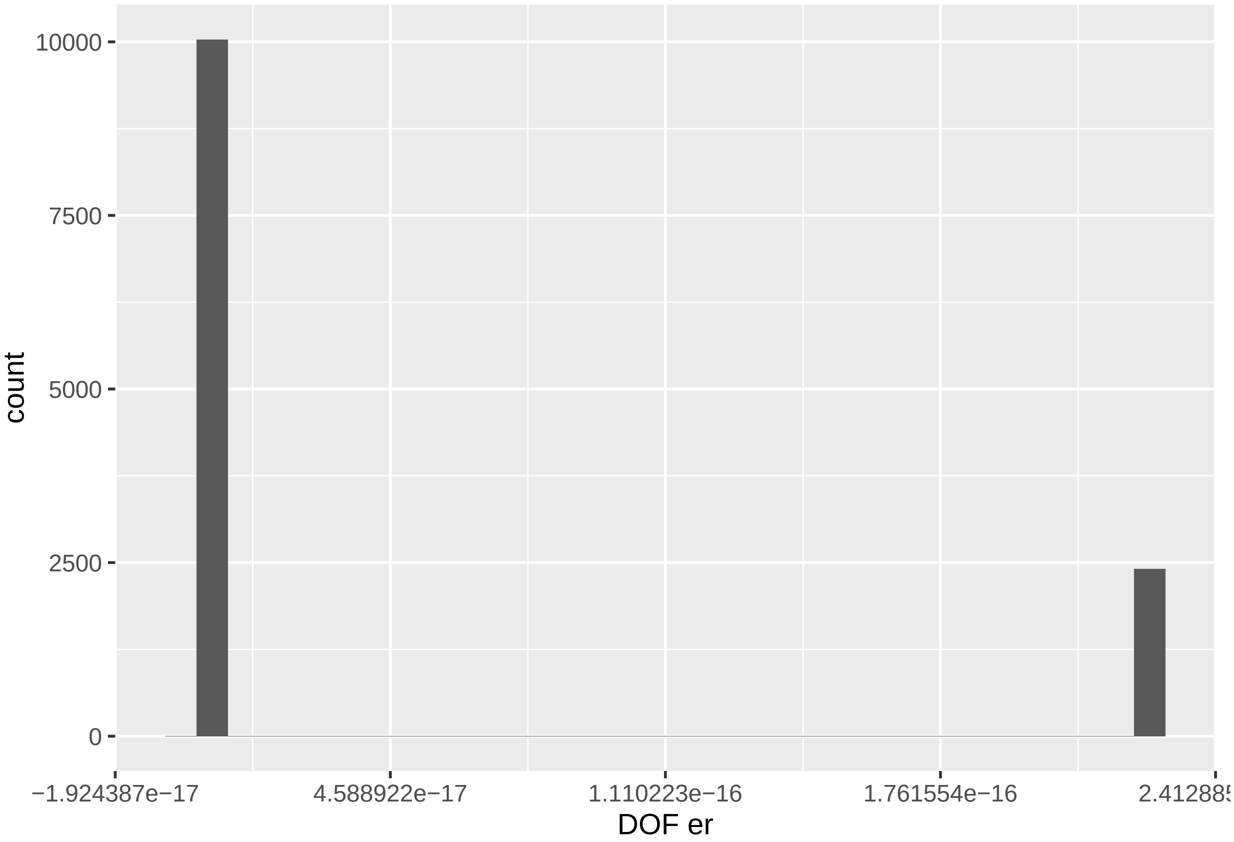
```
qplot(as.vector(e2), xlab="SD err")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



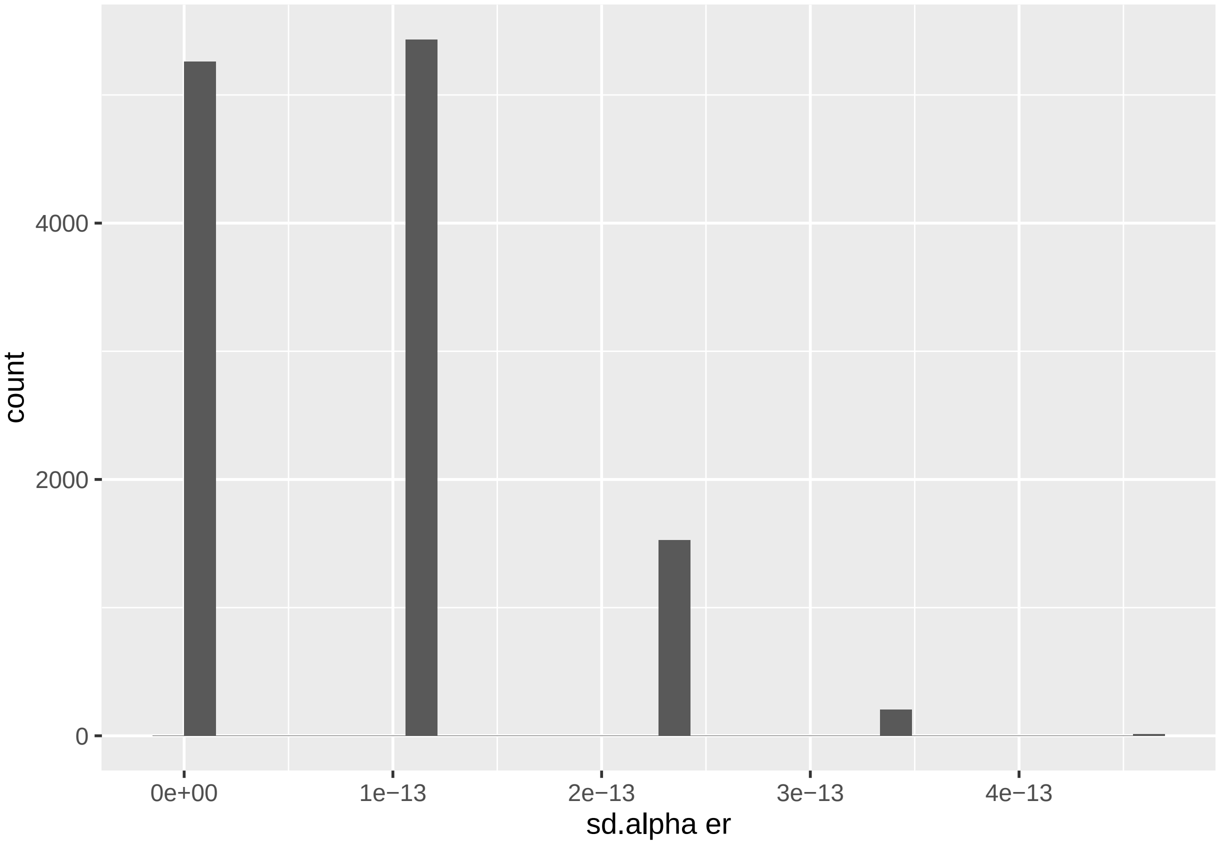
```
qplot(as.vector(e3), xlab= "DOF er")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

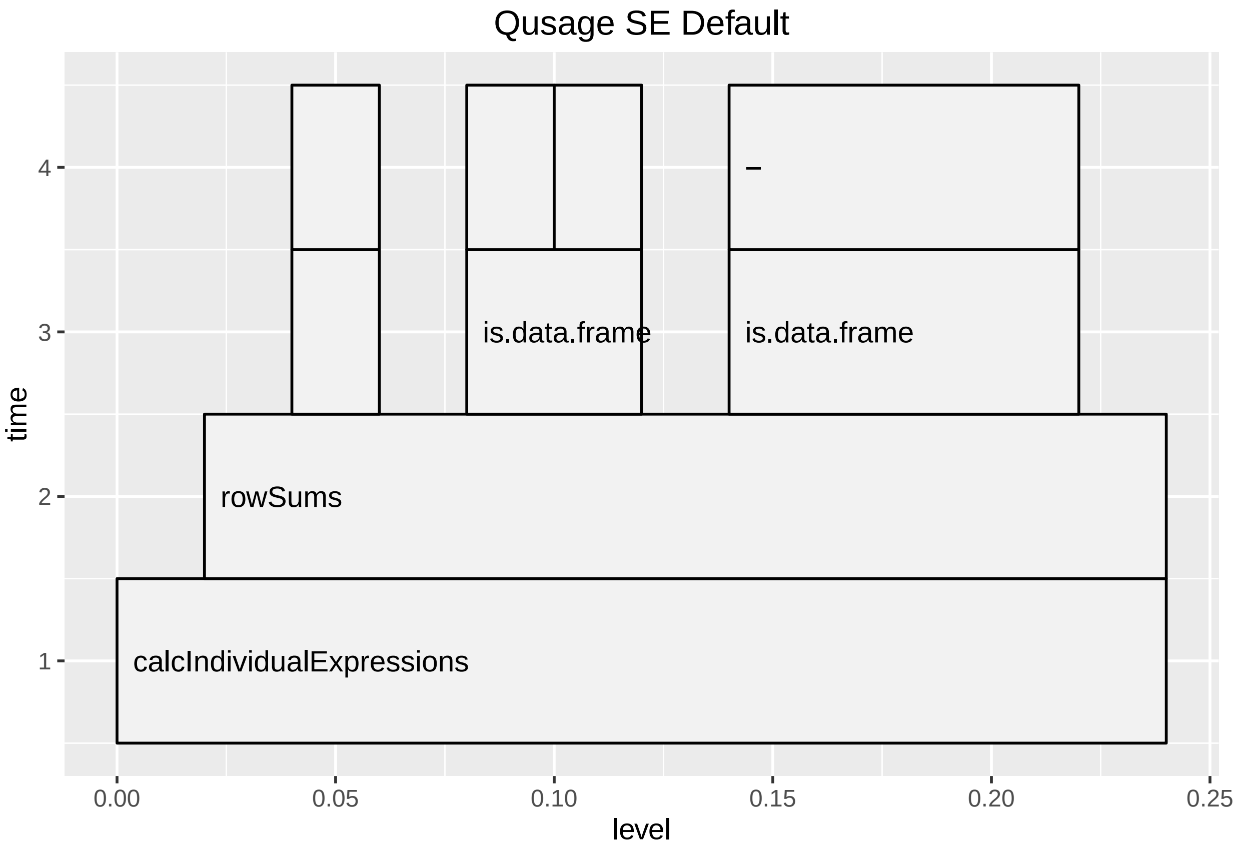


```
qplot(as.vector(e4), xlab="sd.alpha er")
```

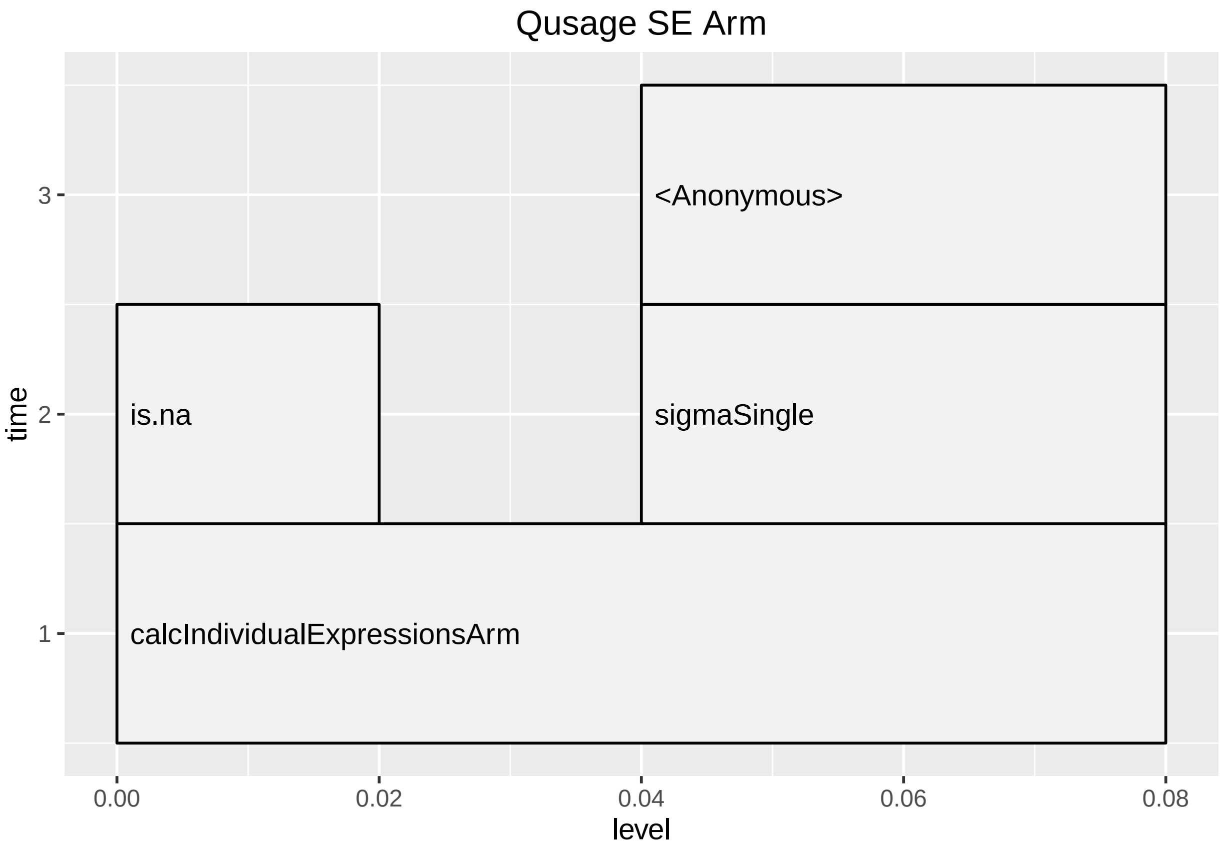
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



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



```r
ggplot(y2)+labs(title="Qusage SE Arm")
```



```
#this shows that the only difference is the vector of Non-NA columns per each row; which is the same as
seMB<-microbenchmark(
testSE1<-calcIndividualExpressions(Baseline,PostTreatment,paired=FALSE,min.variance.factor=10^-6,na.rm='
testSE2<-calcIndividualExpressionsArm(Baseline,PostTreatment,paired=FALSE,min.variance.factor=10^-6)
)
seMB
```

```
## Unit: milliseconds
##
##  testSE1 <- calcIndividualExpressions(Baseline, PostTreatment,      paired = FALSE, min.variance.fact
##             testSE2 <- calcIndividualExpressionsArm(Baseline, PostTreatment,      paired = FALSE, mi
##        min        lq      mean   median        uq      max neval cld
##  170.43174 175.11130 201.57736 180.6738 239.29210 263.3010   100   b
##   81.91386  84.22734  91.19849  86.3656  87.97492 151.9621   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 Alternate training 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 makeComparison funciton which compares two different labels after splitting the eset by column names label type.

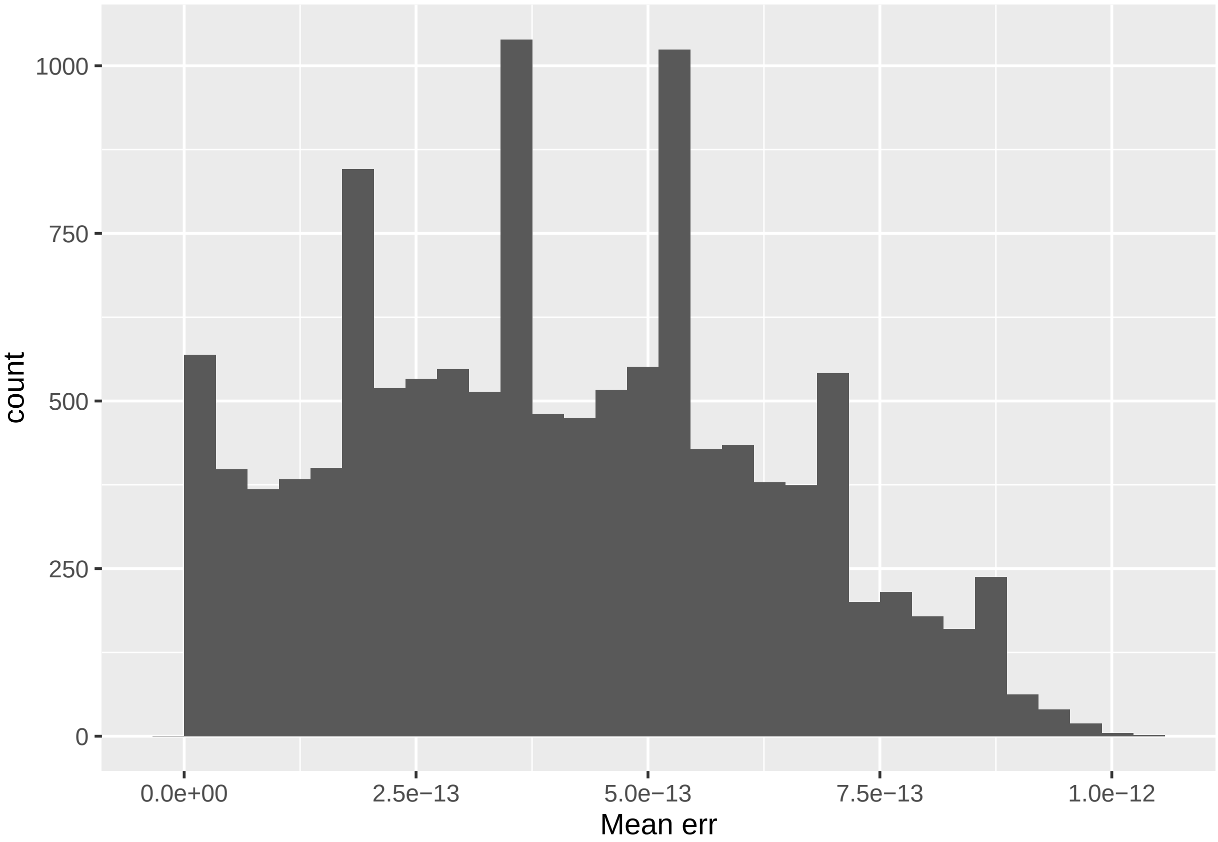
4 Paired end revised demo set , not split by label

```r
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")
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")

eset.1<-eset-40.3
eset.2<-eset+100.5
ncol(eset.1)
```

```
## [1] 268
```

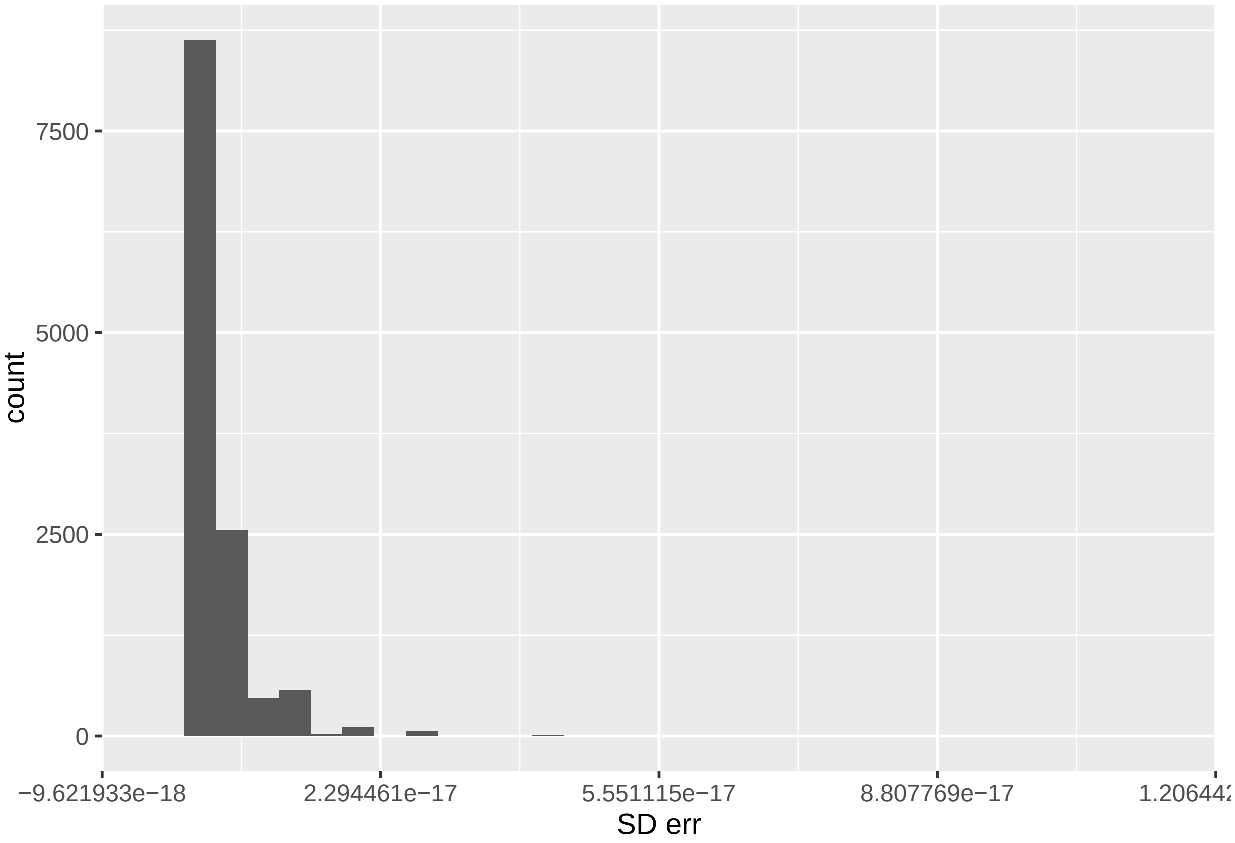
```r
original<-calcIndividualExpressions(eset.1,eset.2,paired=TRUE)
cpp<-calcIndividualExpressionsC(eset.1,eset.2,paired=TRUE)
arm<-calcIndividualExpressionsArm(eset.1,eset.2,paired=TRUE)

e1<-(abs(original[[1]]-arm[[1]]))
e2<-(abs(original[[2]]-arm[[2]]))
e3<-(abs(original[[3]]-arm[[3]]))
e4<-(abs(original[[4]]-arm[[4]]))
qplot(as.vector(e1),xlab="Mean err")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



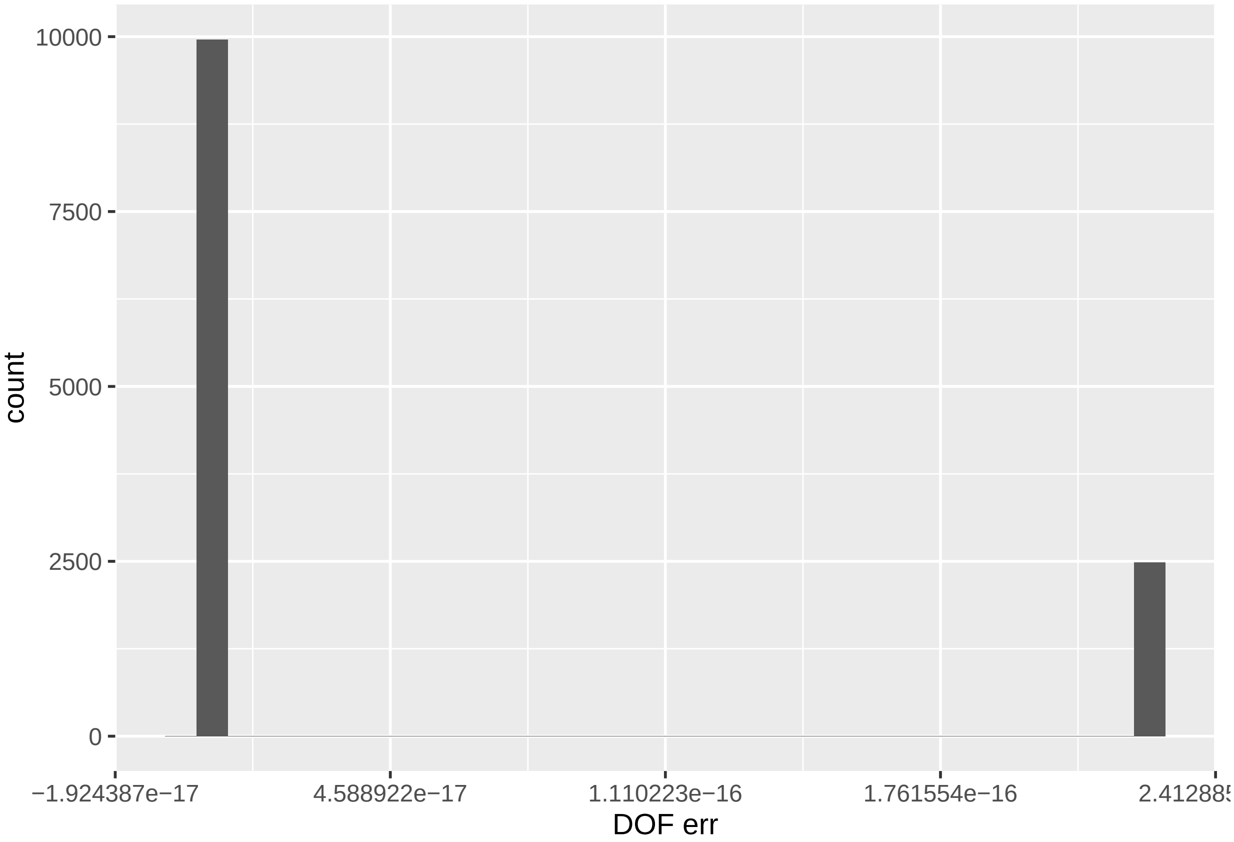
```
qplot(as.vector(e2), xlab="SD err")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



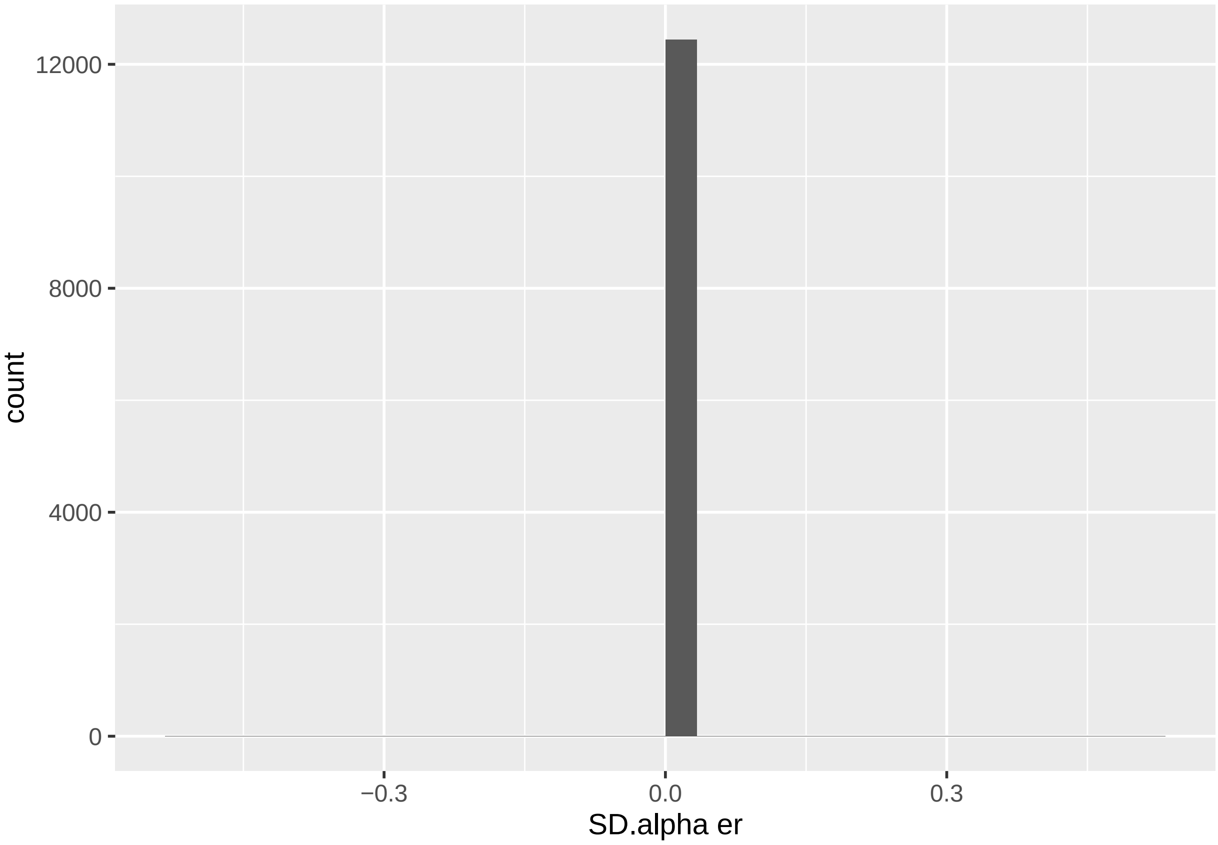
```
qplot(as.vector(e3), xlab="DOF err")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
qplot(as.vector(e4), xlab="SD.alpha er")
```

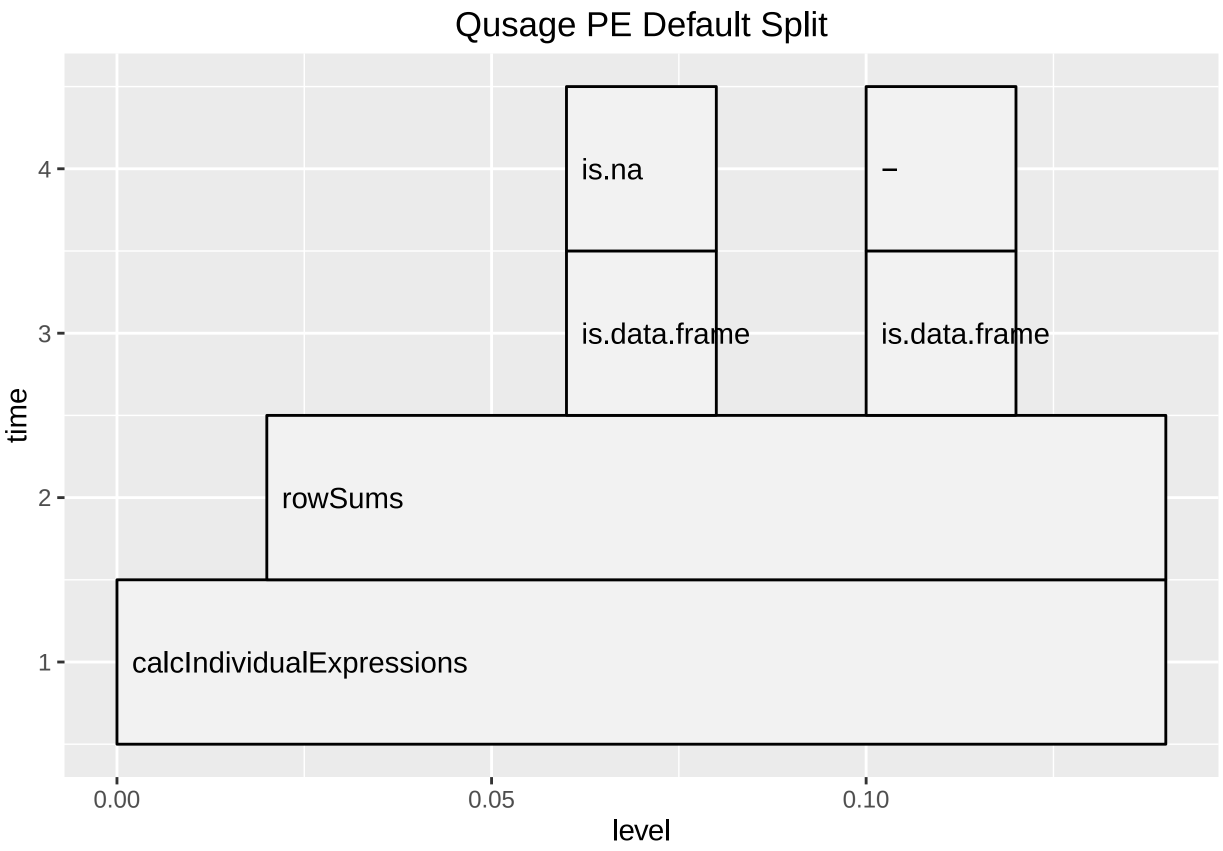
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



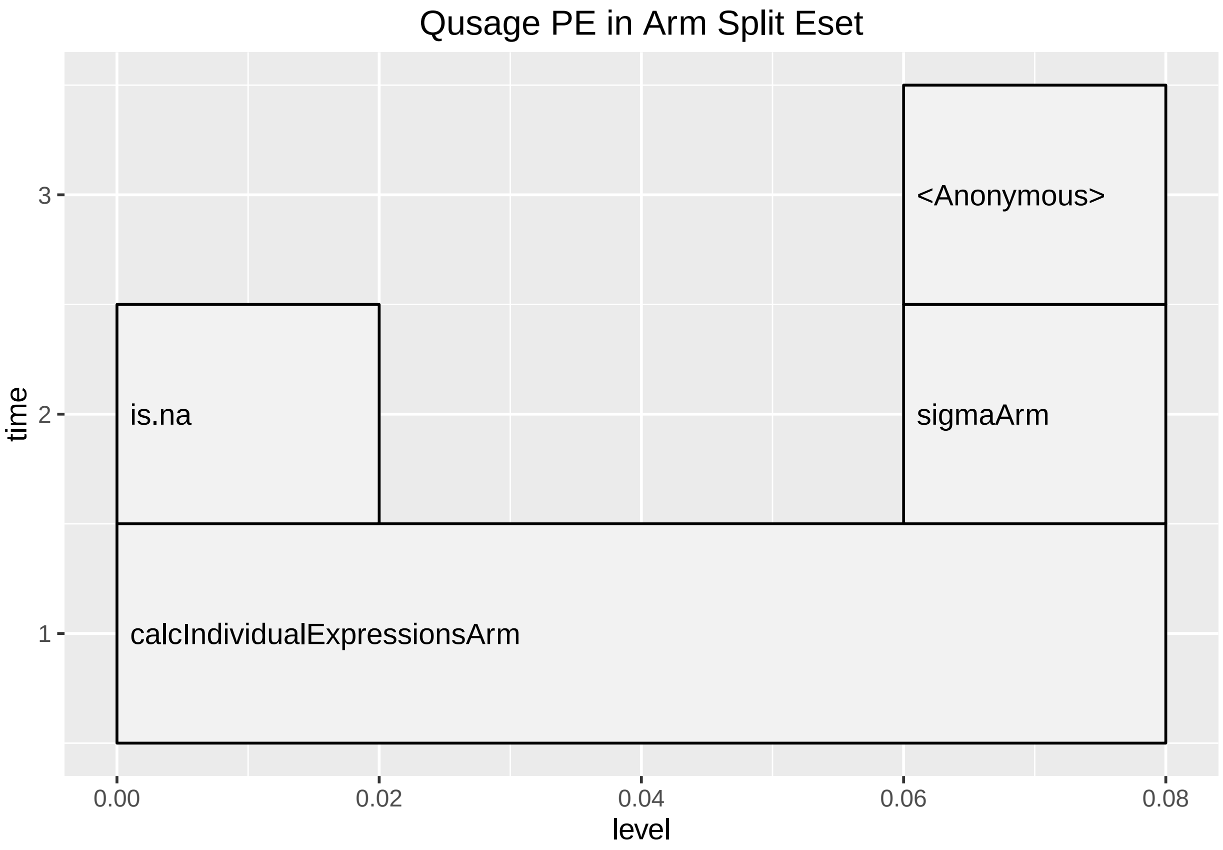
```
microbenchmark(
 original<-calcIndividualExpressions(eset.1,eset.2,paired=TRUE),
 cpp<-calcIndividualExpressionsC(eset.1,eset.2,paired=TRUE),
 arm<-calcIndividualExpressionsArm(eset.1,eset.2,paired=TRUE))
```

```
## Unit: milliseconds
##                                                                expr
##  original <- calcIndividualExpressions(eset.1, eset.2, paired = TRUE)
##      cpp <- calcIndividualExpressionsC(eset.1, eset.2, paired = TRUE)
##    arm <- calcIndividualExpressionsArm(eset.1, eset.2, paired = TRUE)
##        min        lq      mean    median        uq      max neval cld
##  140.54593 146.48165 163.71777 149.61885 184.50489 231.2002   100   c
##  123.66894 127.80480 140.82032 131.19882 138.93064 245.2807   100  b
##   87.66164  90.62898  99.45325  92.79854  98.87934 196.5865   100 a
```

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

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



```r
tt<-profr(calcIndividualExpressionsArm(eset.1,eset.2,paired=TRUE))
ggplot(tt)+ labs(title="Qusage PE in Arm Split Eset")
```



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

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

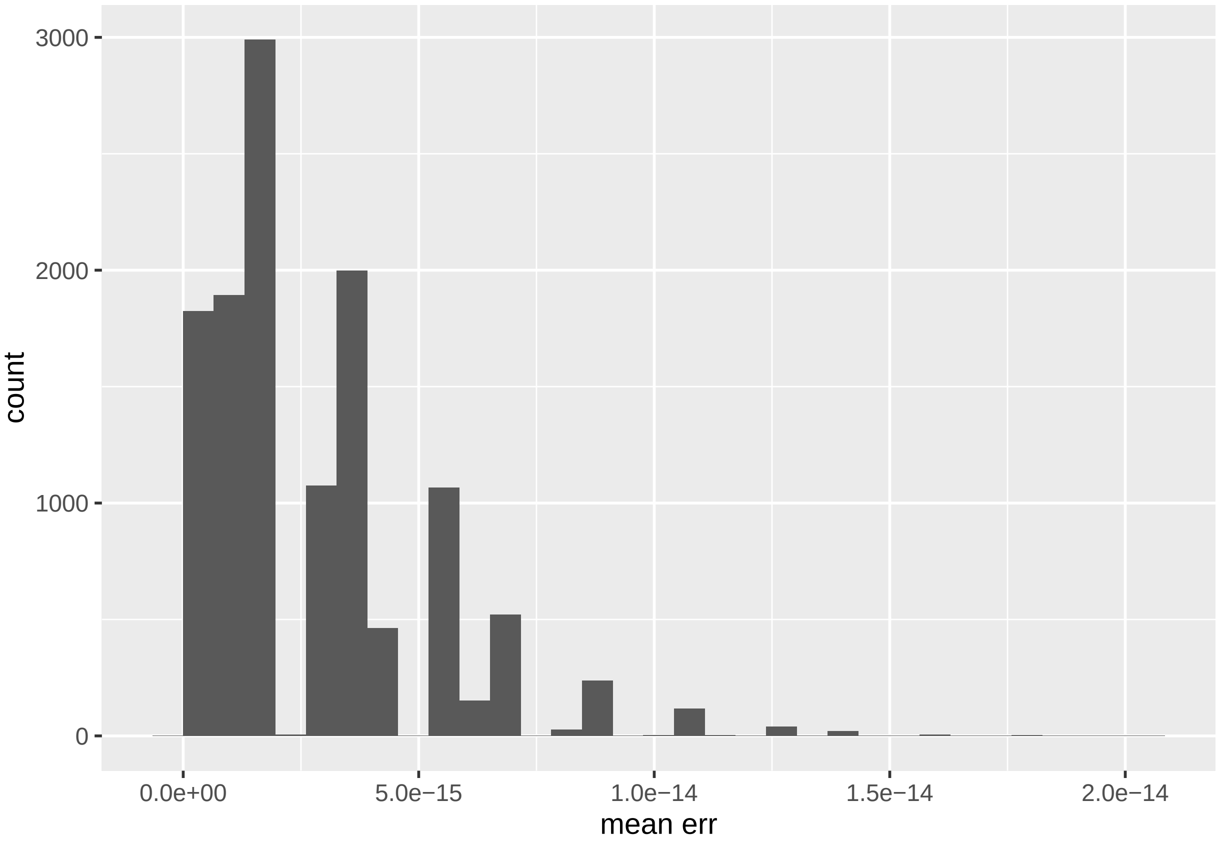
```
library(microbenchmark)
library(profr)
library(ggplot2)
library(Rcpp)
eset.1<-system.file("extdata","eset.1.RData",package="speedSage")
eset.2<-system.file("extdata","eset.2.RData",package="speedSage")
load(eset.1)
load(eset.2)
ncol(eset.1) #split by label
```

```
## [1] 134
```

```
sourceCpp(file="/home/anthonycolombo/Documents/qusage/qusage_repos/qusage_speed/R/sigmasCpp.cpp")
sourceCpp(file="/home/anthonycolombo/Documents/qusage/qusage_repos/qusage_speed/R/sigmaArm.cpp")
original<-calcIndividualExpressions(eset.1,eset.2,paired=FALSE)
cpp<-calcIndividualExpressionsC(eset.1,eset.2,paired=FALSE)
arm<-calcIndividualExpressionsArm(eset.1,eset.2,paired=FALSE)
e1<-(abs(original[[1]]-arm[[1]]))
e2<-(abs(original[[2]]-arm[[2]]))
e3<-(abs(original[[3]]-arm[[3]]))
```

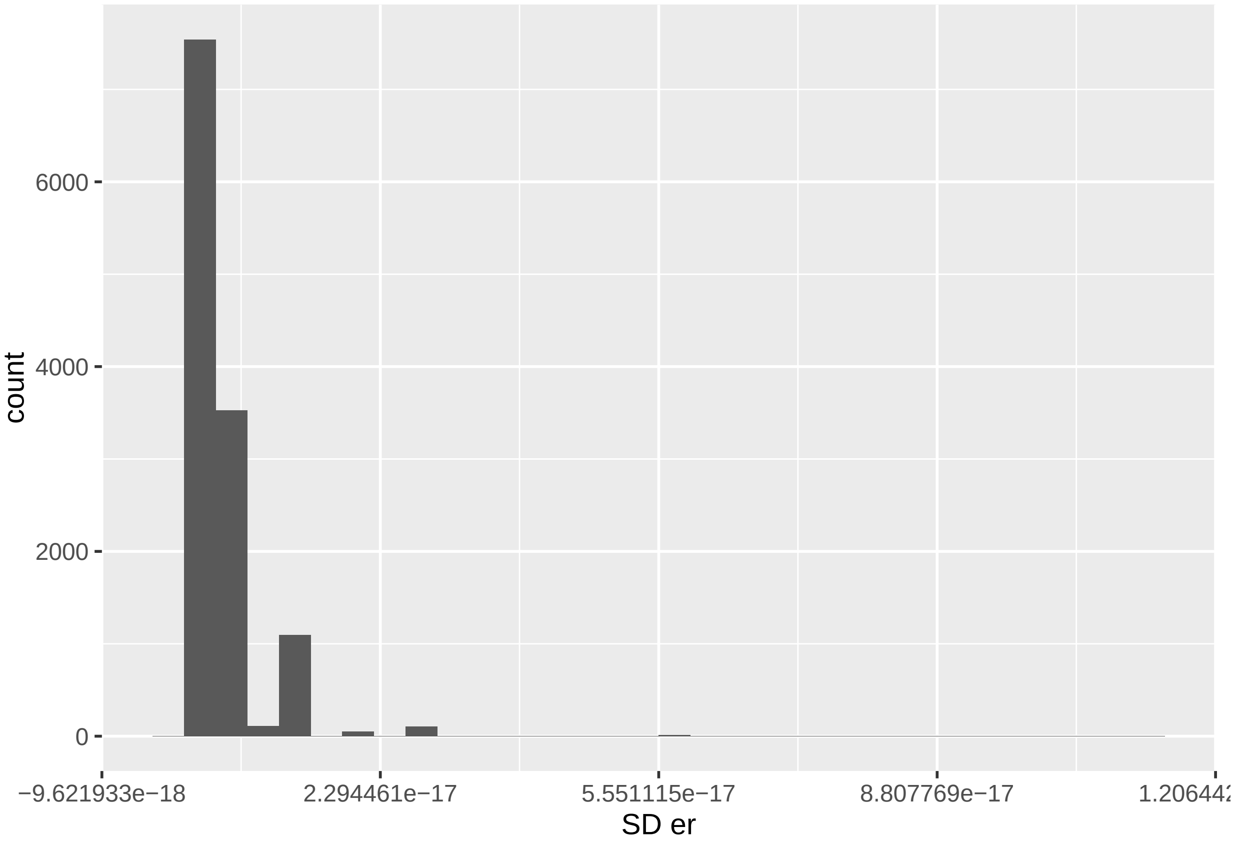
```
e4<-(abs(original[[4]]-arm[[4]]))
qplot(as.vector(e1), xlab="mean err")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



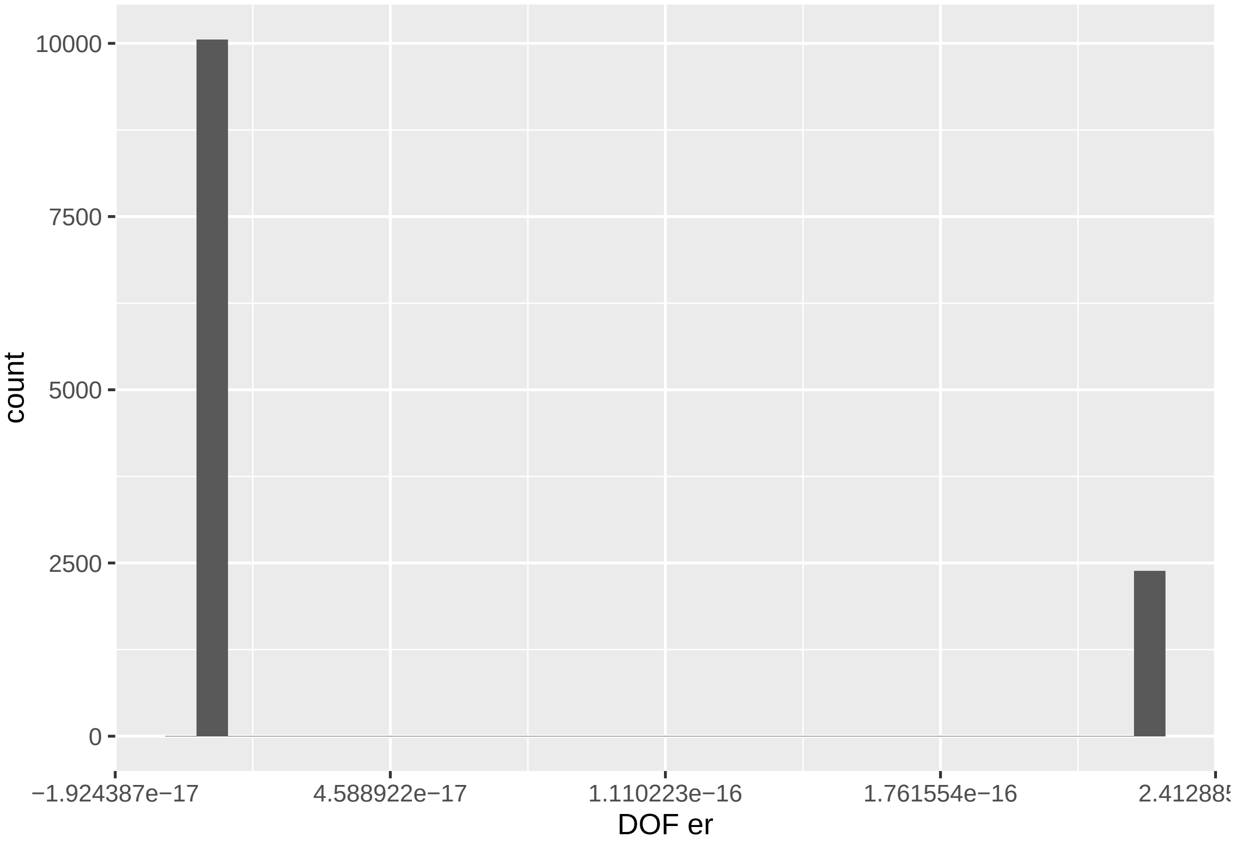
```
qplot(as.vector(e2), xlab="SD er")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



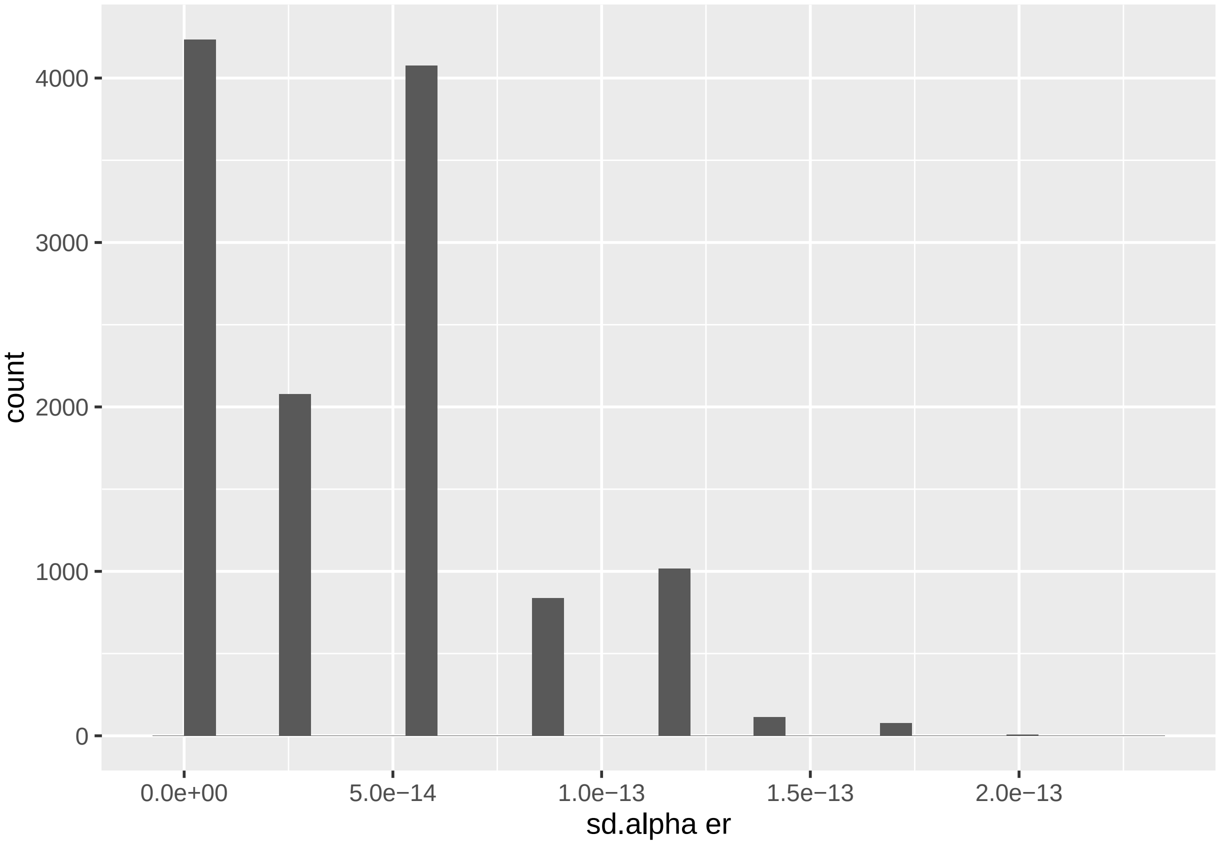
```
qplot(as.vector(e3), xlab="DOF er")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
qplot(as.vector(e4), xlab="sd.alpha er")
```

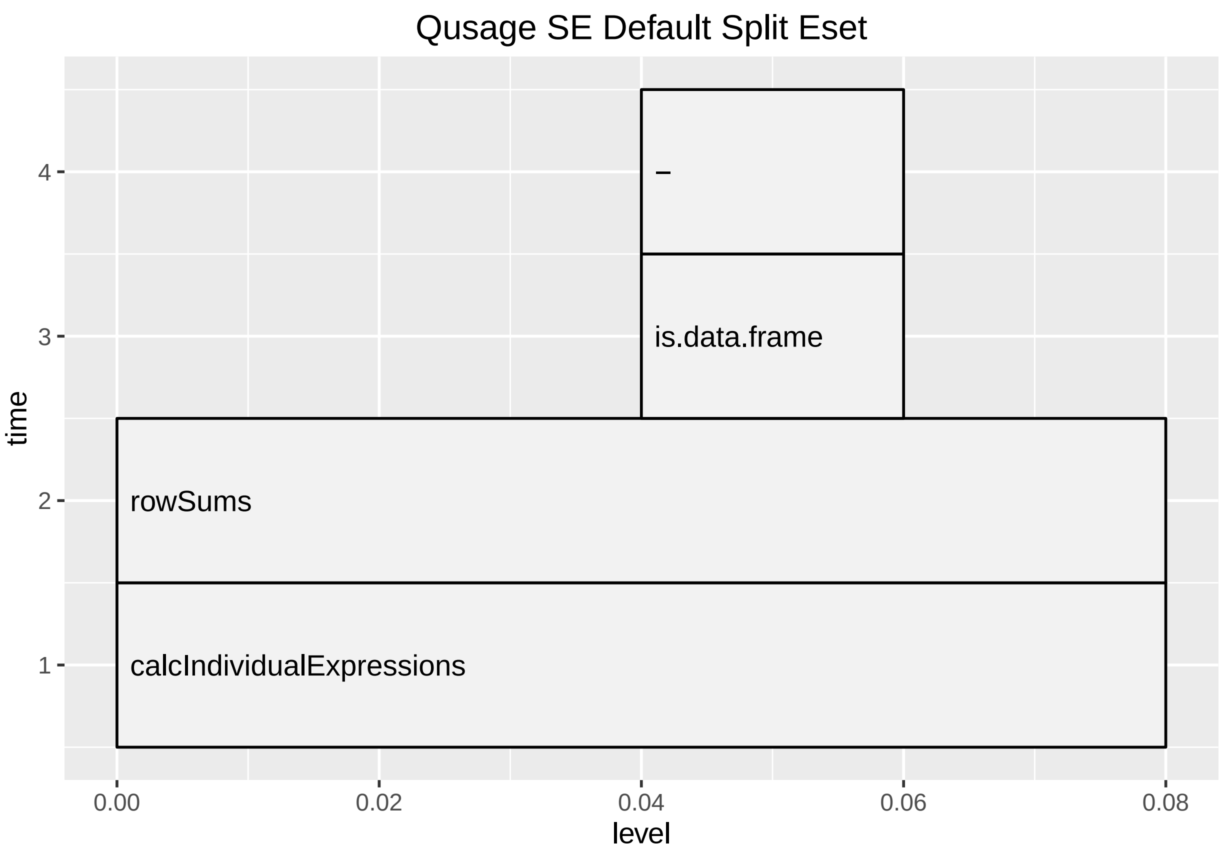
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



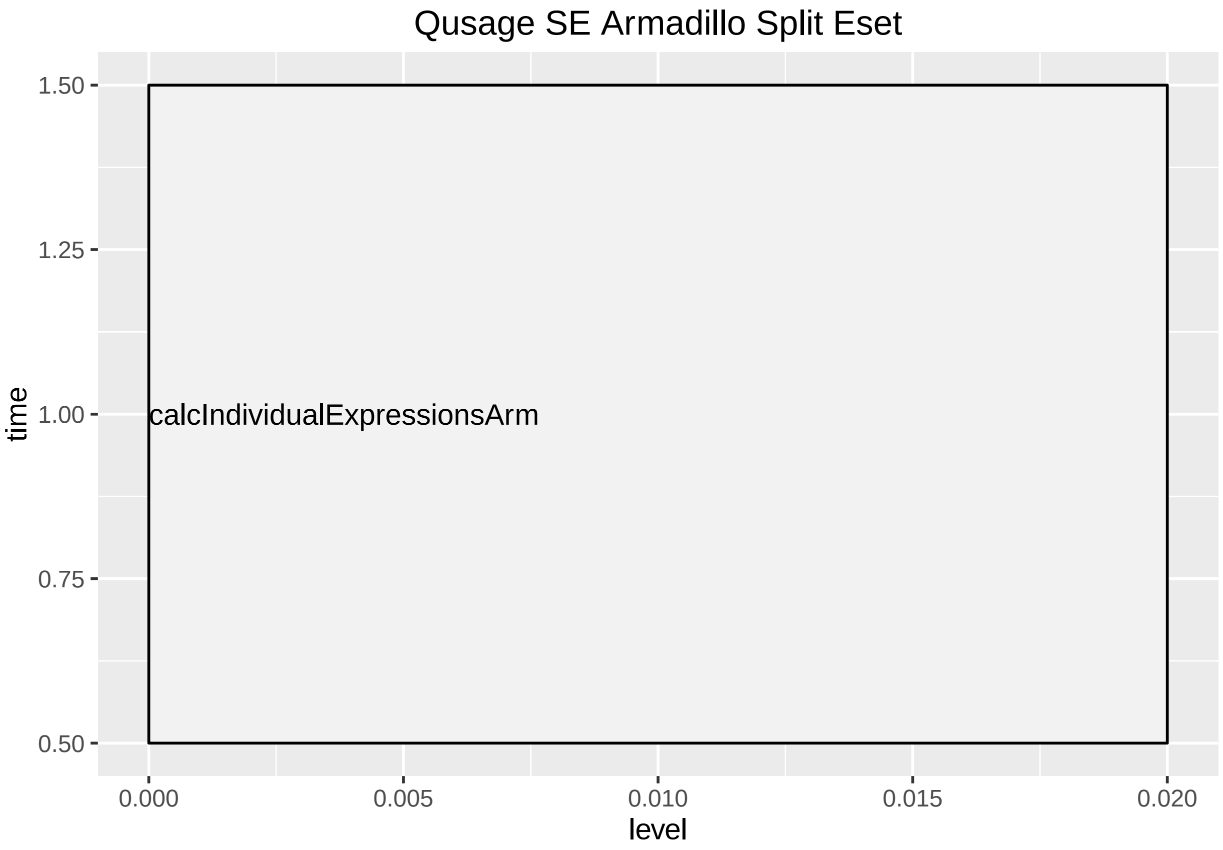
```
microbenchmark(
 original<-calcIndividualExpressions(eset.1,eset.2,paired=FALSE),
 cpp<-calcIndividualExpressionsC(eset.1,eset.2,paired=FALSE),
 arm<-calcIndividualExpressionsArm(eset.1,eset.2,paired=FALSE))
```

```
## Unit: milliseconds
##                                                                  expr
##  original <- calcIndividualExpressions(eset.1, eset.2, paired = FALSE)
##      cpp <- calcIndividualExpressionsC(eset.1, eset.2, paired = FALSE)
##    arm <- calcIndividualExpressionsArm(eset.1, eset.2, paired = FALSE)
##       min       lq     mean   median       uq       max neval cld
##  87.80482 91.61654 95.11149 93.87329 96.64628 153.43794   100   c
##  61.81630 63.40558 66.05675 65.21917 67.20239  86.72837   100  b
##  42.15936 43.60420 45.67067 44.63987 46.24137 111.67705   100 a
```

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



```
ggplot(y) + labs(title="Qusage SE Armadillo Split Eset")
```


Qusage SE Armadillo Split Eset
1.50
1.25
1.00
0.75
0.50
time
calcIndividualExpressionsArm
0.000
0.005
0.010
0.015
0.020
level