

Qusage: Speeding up in RcppArmadillo

Timothy J. Triche, Jr, Anthony R. Colombo

26 February, 2016

Contents

1	SpeedSage Intro	1
1.1	changes Armadillo C++	1
2	Individual Expression Function	1
3	Alternate training sets	15
4	Paired end revised demo set , not split by label	15
5	Non-paired end the eset.1, eset.2 split by label	21

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
```

```
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
```

```
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
```

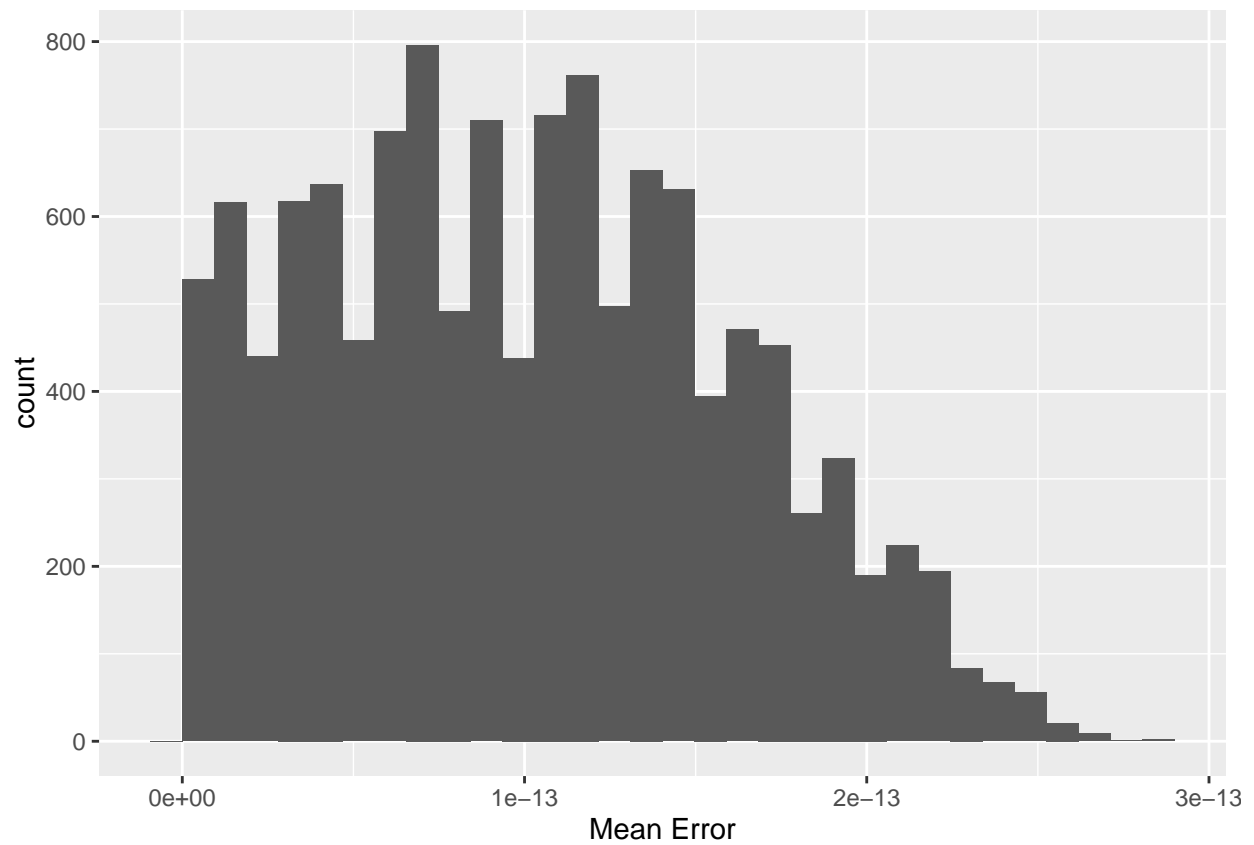
```
#paired
```

```
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'
```

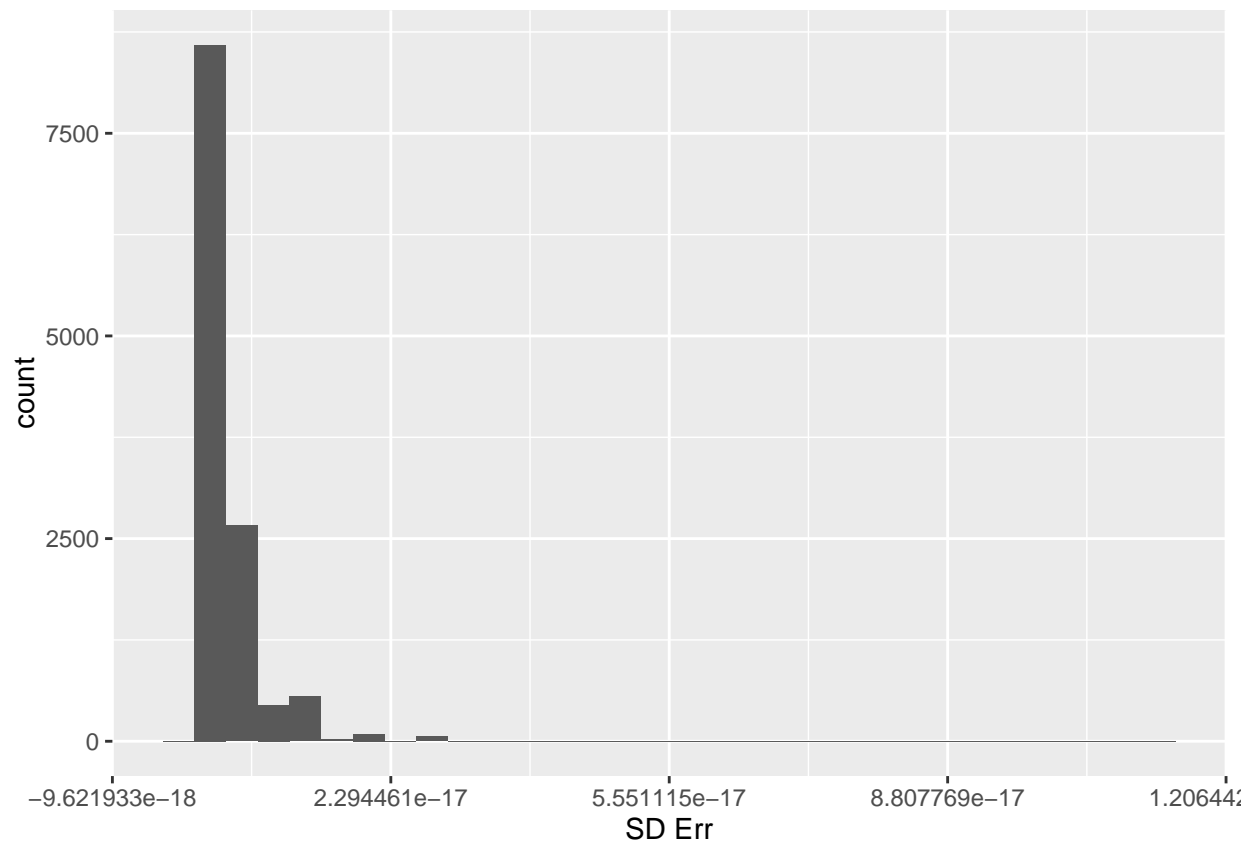
```
test2<-calcIndividualExpressionsC(Baseline, PostTreatment, paired=TRUE, min.variance.factor=10^-6)
test3<-calcIndividualExpressions(Baseline, PostTreatment, paired=TRUE, min.variance.factor=10^-6, na.rm=TRUE)
qplot(abs(test1[[1]]-test3[[1]]), xlab="Mean Error")
```

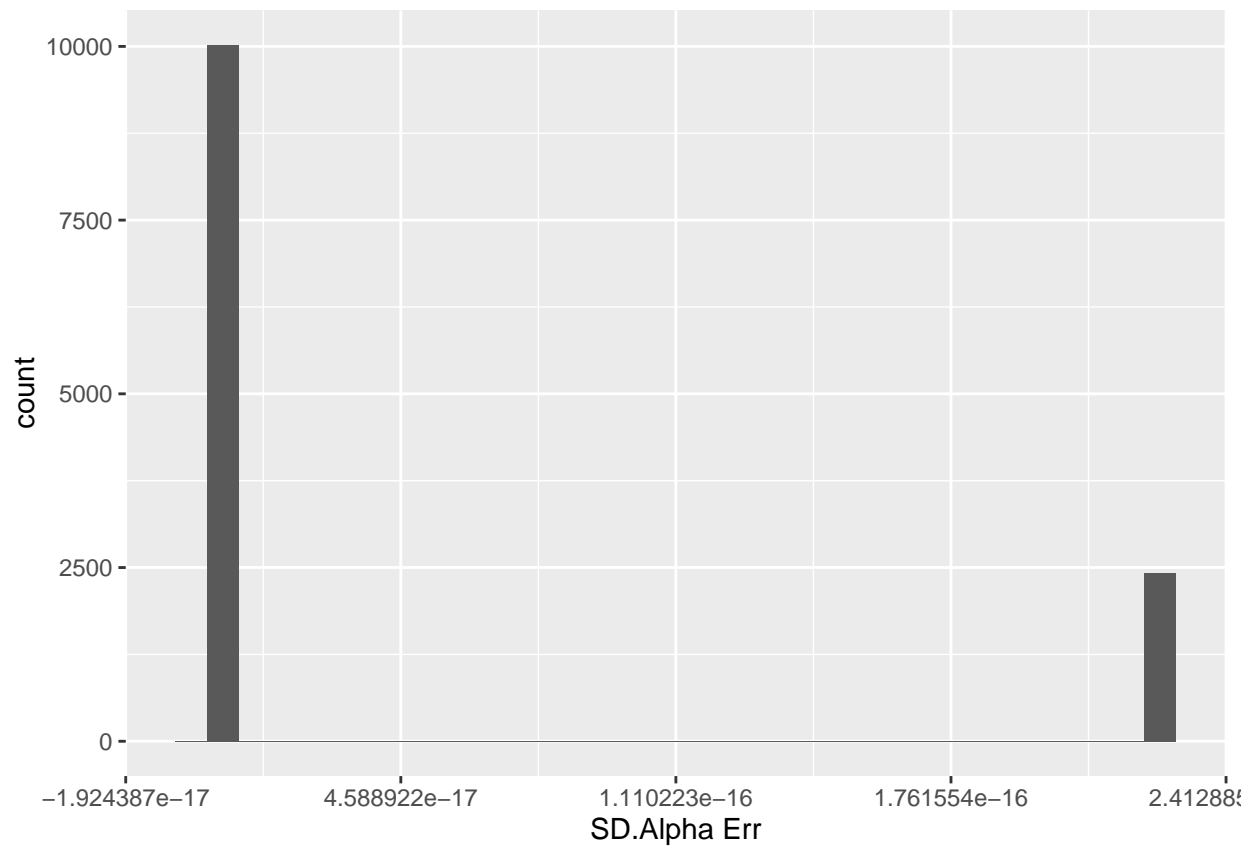
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



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

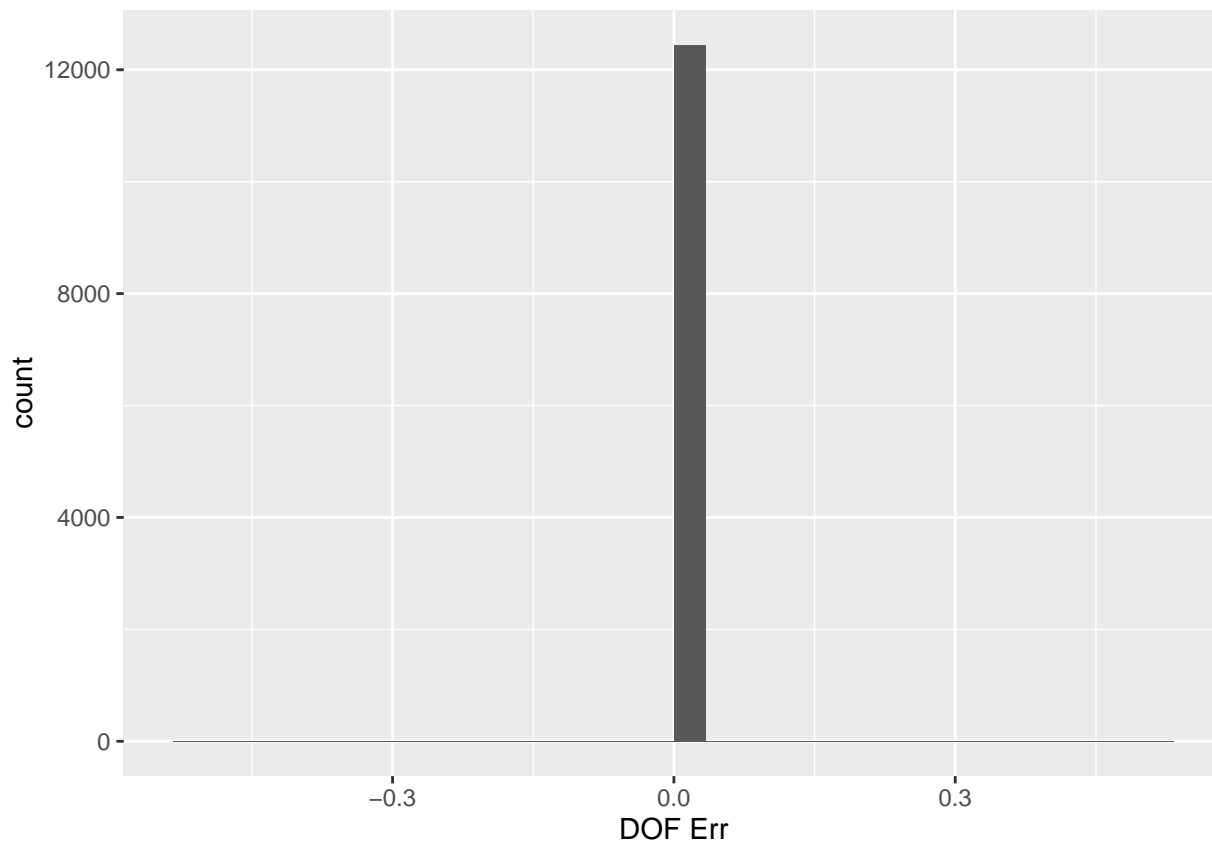
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```





```
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=TRUE),
  mb
```

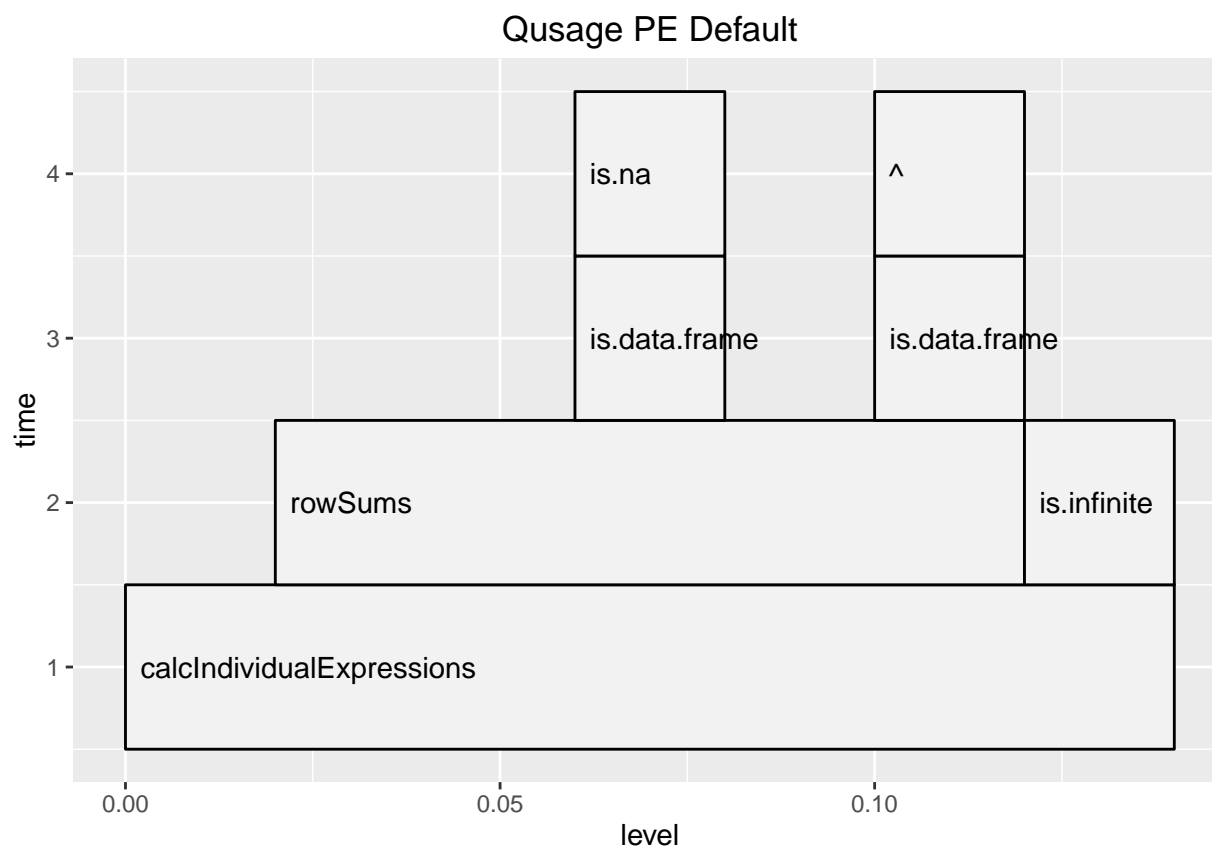
```
## Unit: milliseconds
```

```
##
##           test1 <- calcIndividualExpressionsArm(Baseline, PostTreatment,      paired = TRUE, min.v
##           test2 <- calcIndividualExpressionsC(Baseline, PostTreatment,      paired = TRUE, min.v
## test3 <- calcIndividualExpressions(Baseline, PostTreatment, paired = TRUE,    min.variance.factor
##      min      lq      mean      median      uq      max neval cld
## 87.03518 89.0748 97.58231 90.78744 94.63863 151.2286 100 a
## 123.35911 126.0537 140.24208 129.01461 134.06502 199.4518 100 b
## 141.59669 144.4918 169.49232 148.41973 202.73389 213.7289 100 c
```

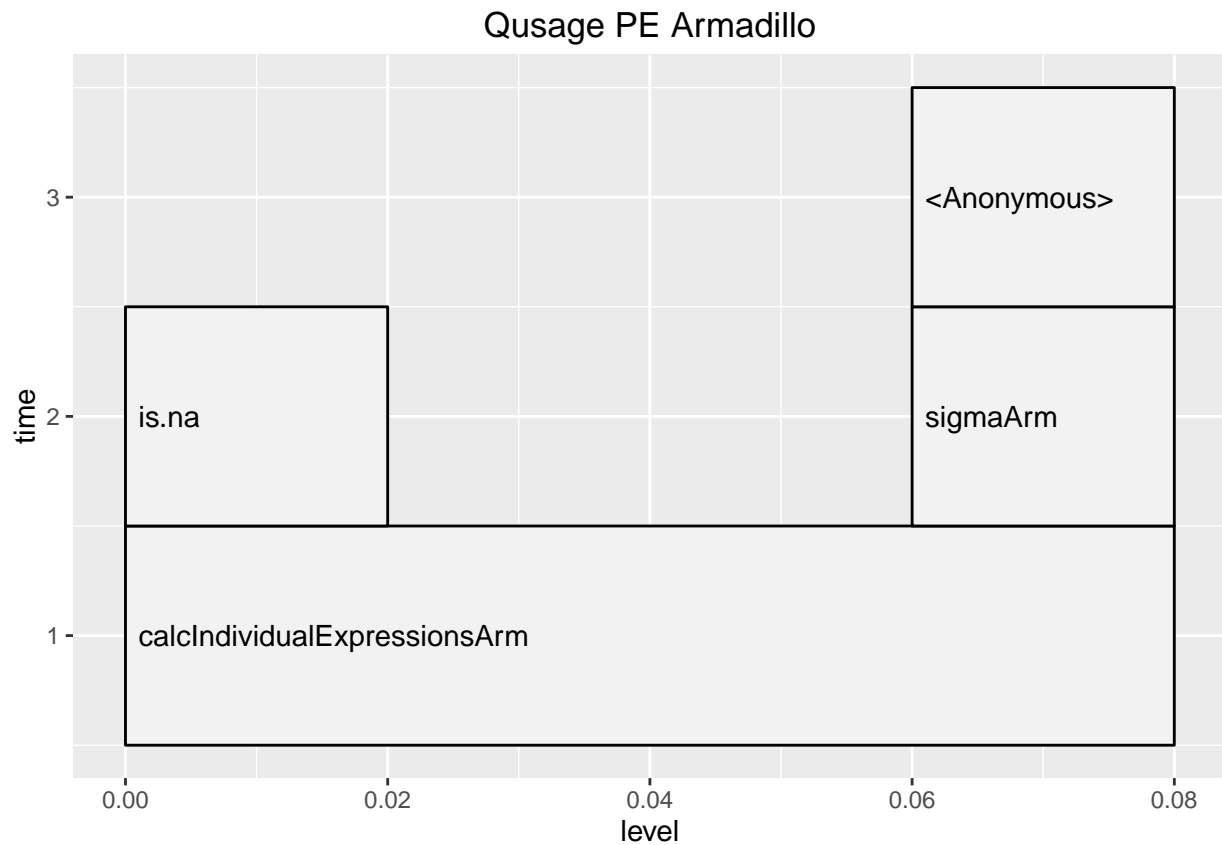
```
require(profr)
```

```
## 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 PE Default")
```



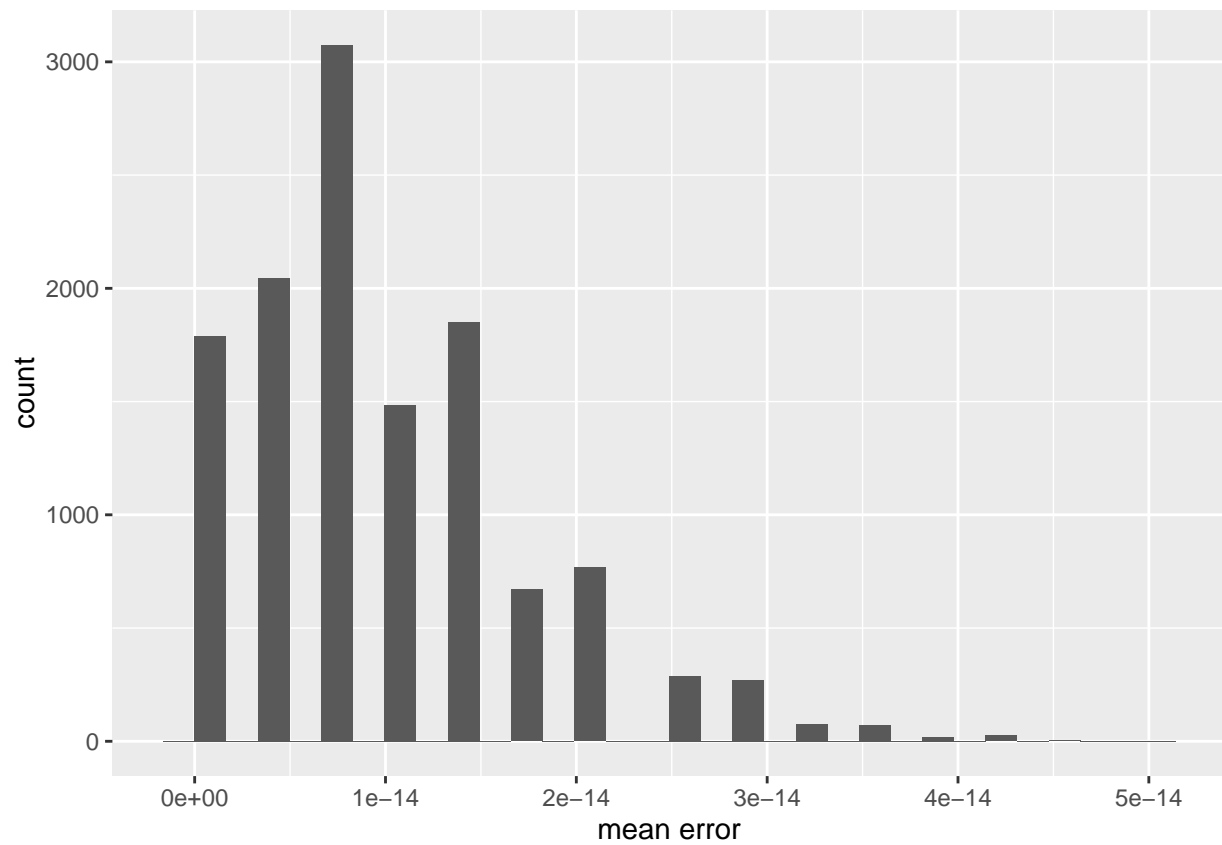
```
x2<-profr(calcIndividualExpressionsArm(Baseline,PostTreatment,paired=TRUE,min.variance.factor=10^-6))
ggplot(x2)+labs(title="Qusage PE Armadillo")
```



```
#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)

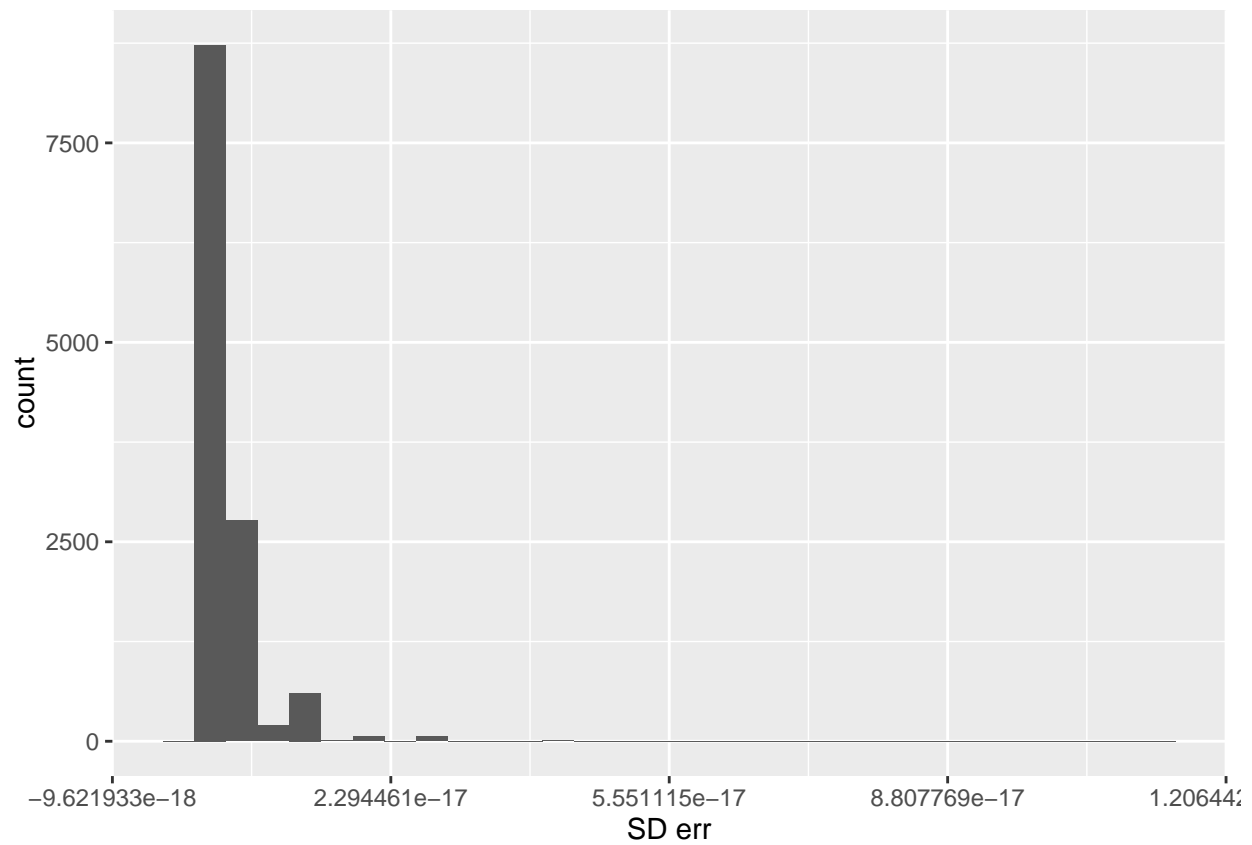
e1<-(abs(testSE1[[1]]-testSE2[[1]]))
e2<-(abs(testSE1[[2]]-testSE2[[2]]))
e3<-(abs(testSE1[[3]]-testSE2[[3]]))
e4<-(abs(testSE1[[4]]-testSE2[[4]]))
qplot(as.vector(e1), xlab="mean error")
```

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

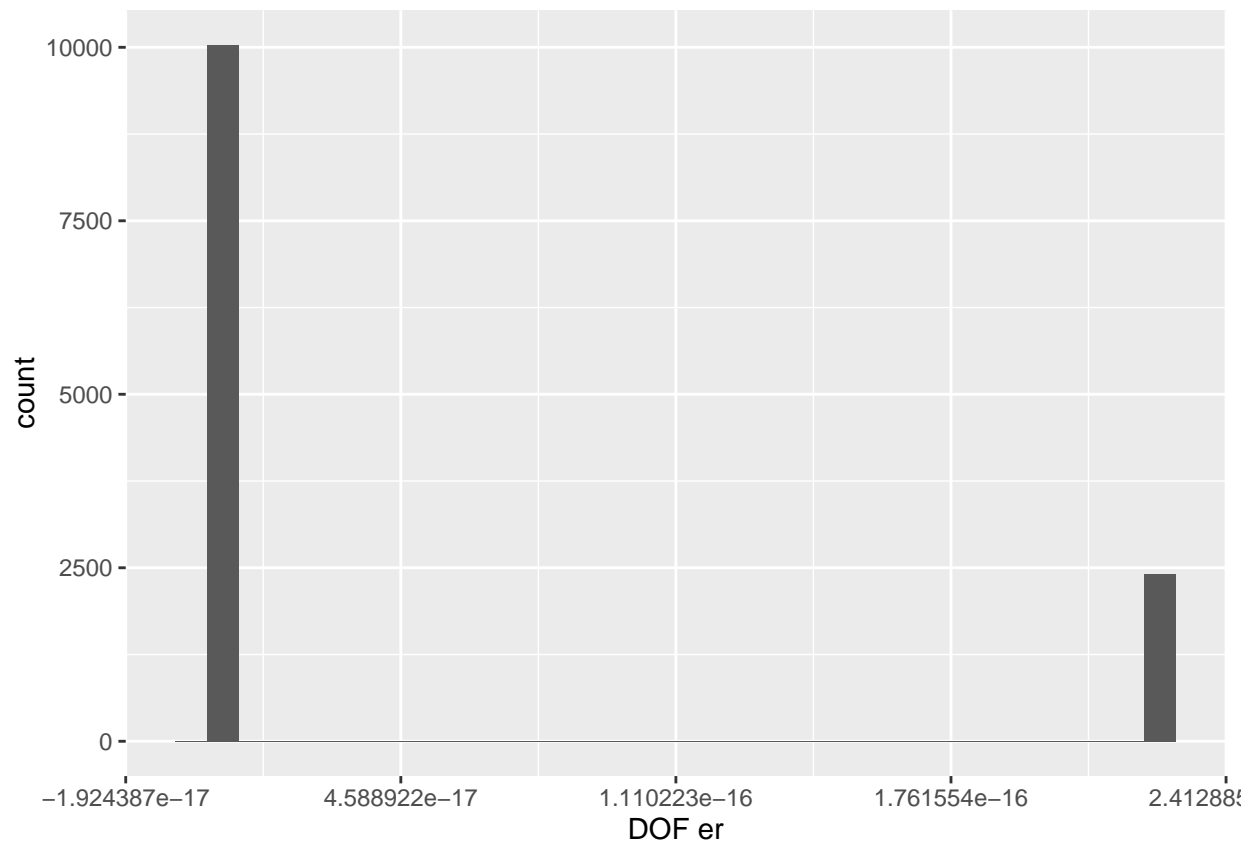
```
qplot(as.vector(e2), xlab="SD err")
```

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



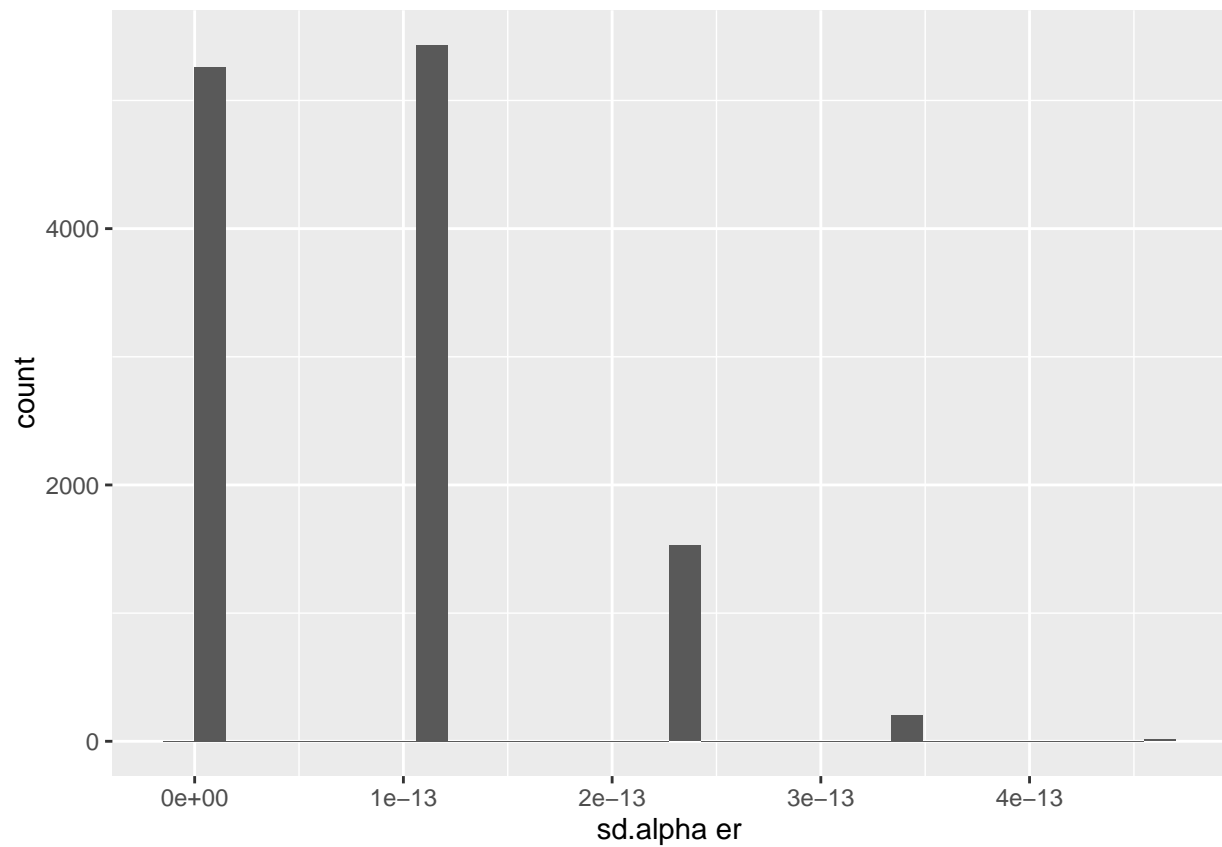
```
qplot(as.vector(e3), xlab= "DOF er")
```

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

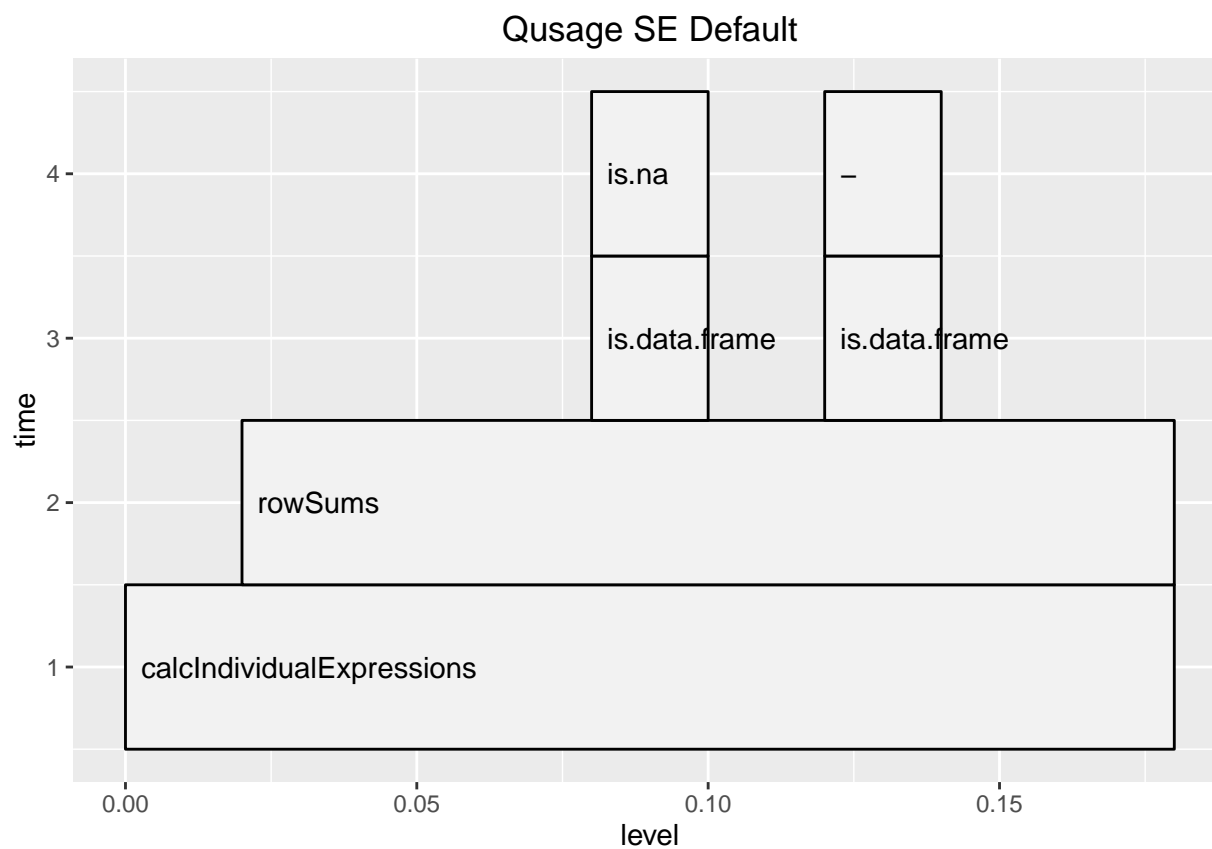


```
qplot(as.vector(e4), xlab="sd.alpha er")
```

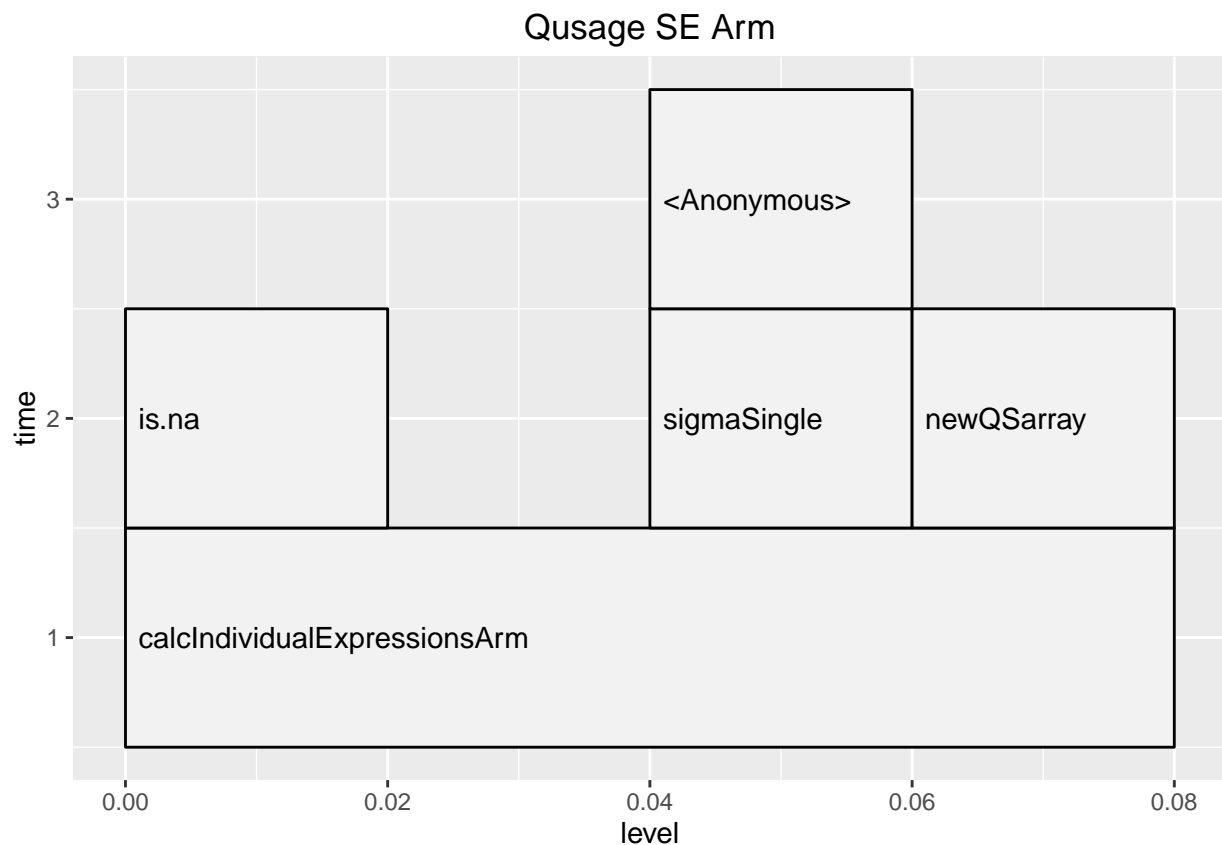
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



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



```
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.factor=10^-6, na.rm=TRUE)
## testSE2 <- calcIndividualExpressionsArm(Baseline, PostTreatment,    paired = FALSE, min.variance.factor=10^-6, na.rm=TRUE)
##      min      lq      mean      median      uq      max neval cld
## 172.33190 176.22536 199.72326 181.30663 238.60225 250.1520  100  b
##   81.88369  85.34136  92.87221  87.20874  89.93394 153.2955  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 function 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")
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")

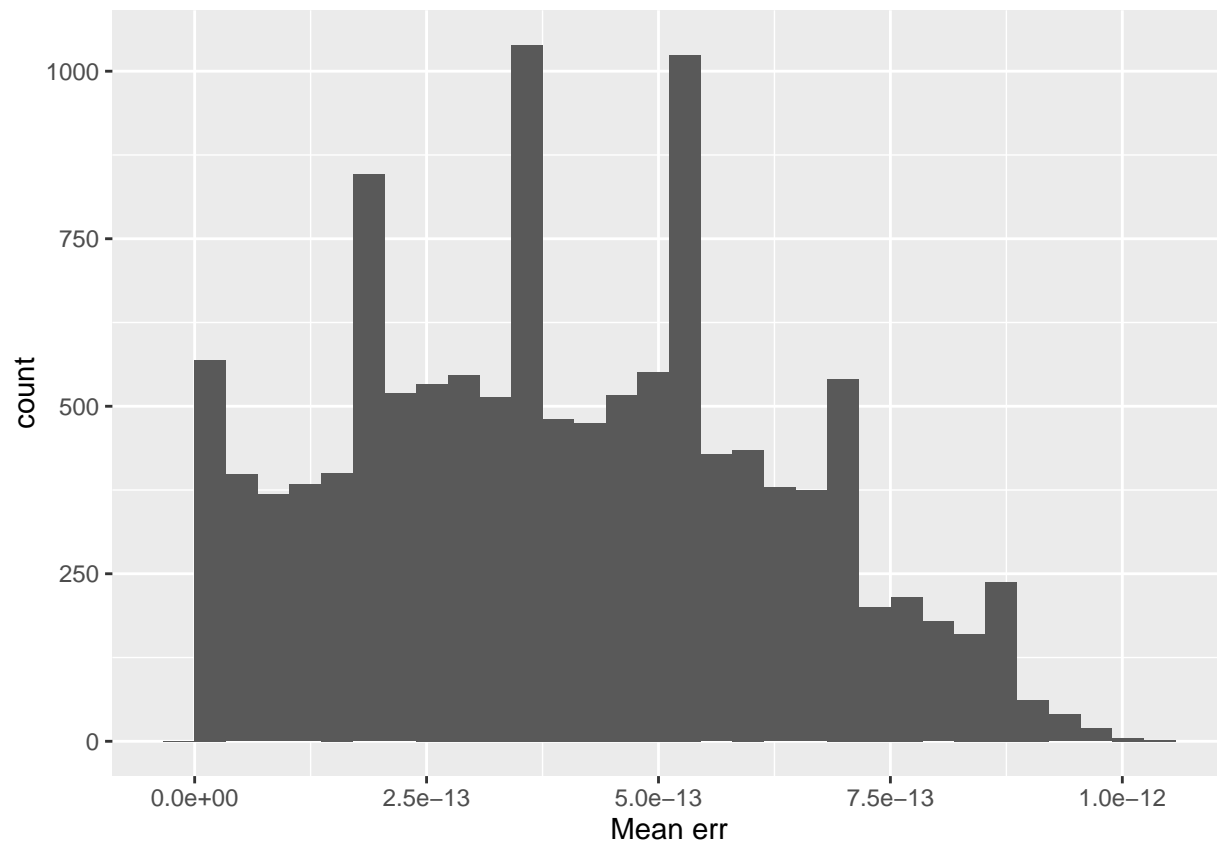
eset.1<-eset-40.3
eset.2<-eset+100.5
ncol(eset.1)
```

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

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

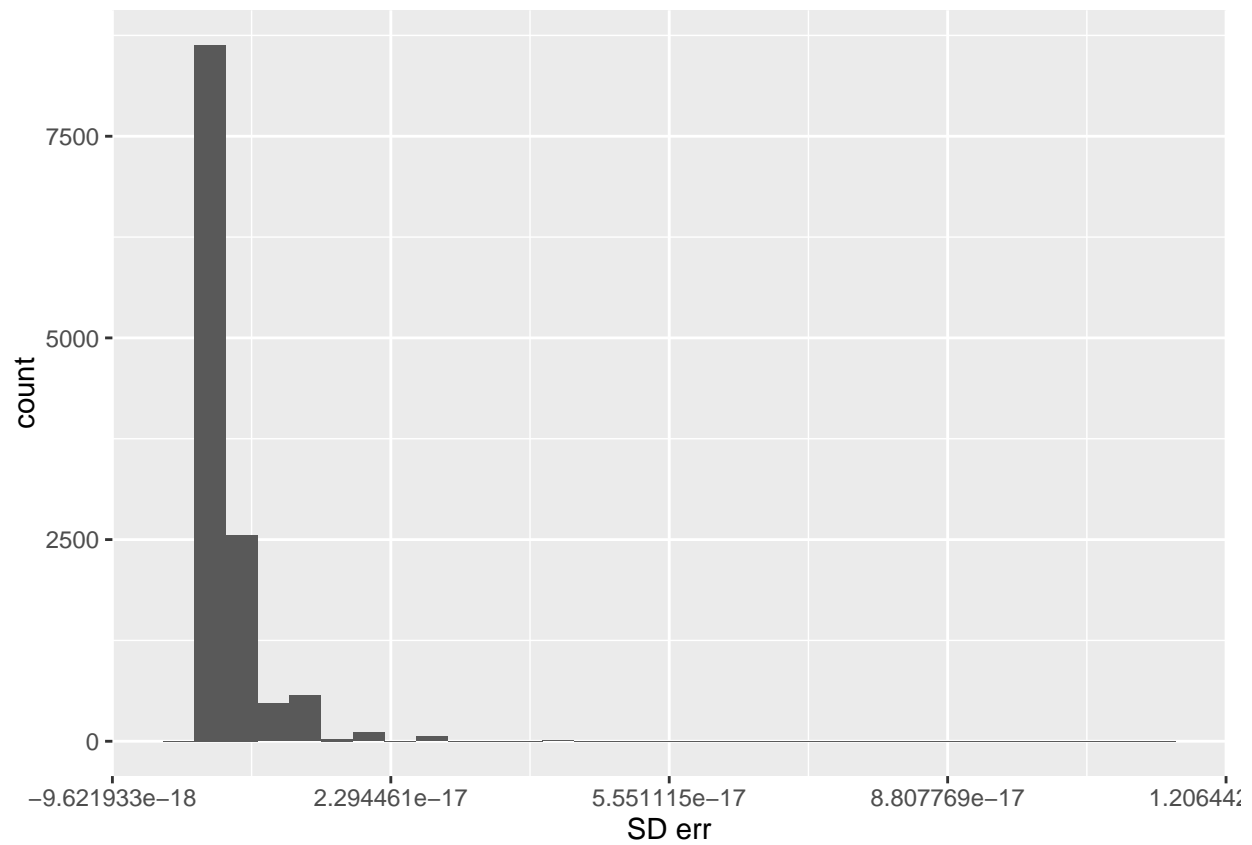
e1<-(abs(original[[1]]-arm[[1]]))
e2<-(abs(original[[2]]-arm[[2]]))
e3<-(abs(original[[3]]-arm[[3]]))
e4<-(abs(original[[4]]-arm[[4]]))
qplot(as.vector(e1),xlab="Mean err")
```

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



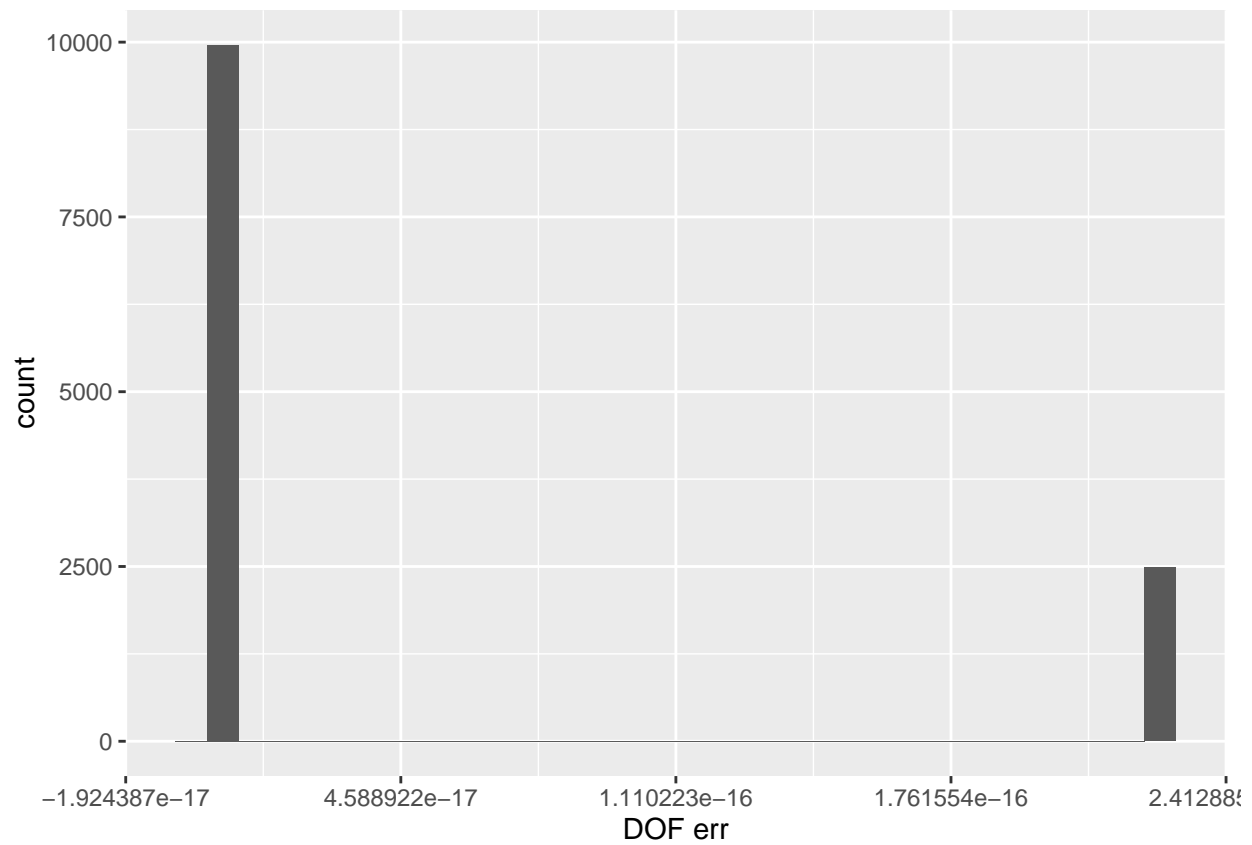
```
qplot(as.vector(e2), xlab="SD err")
```

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

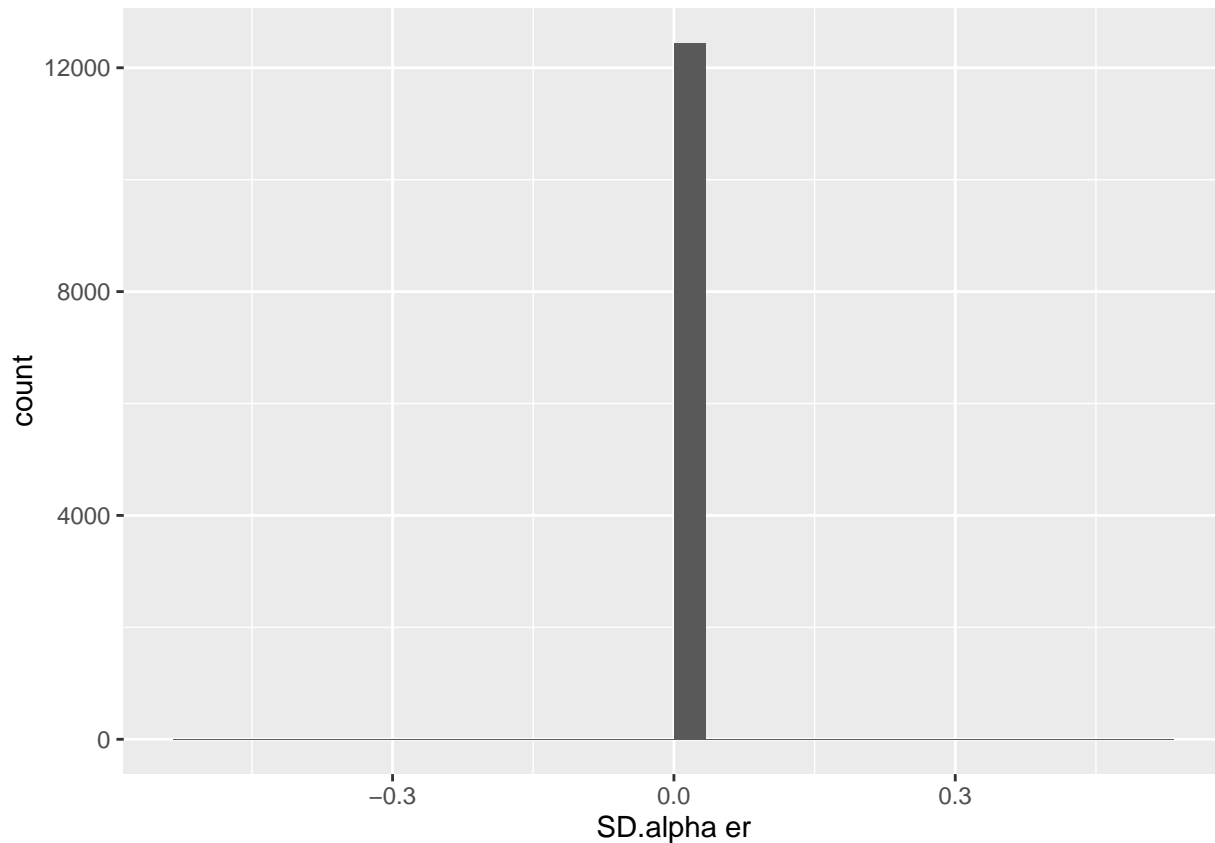
```
qplot(as.vector(e3), xlab="DOF err")
```

```
## `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`.
```

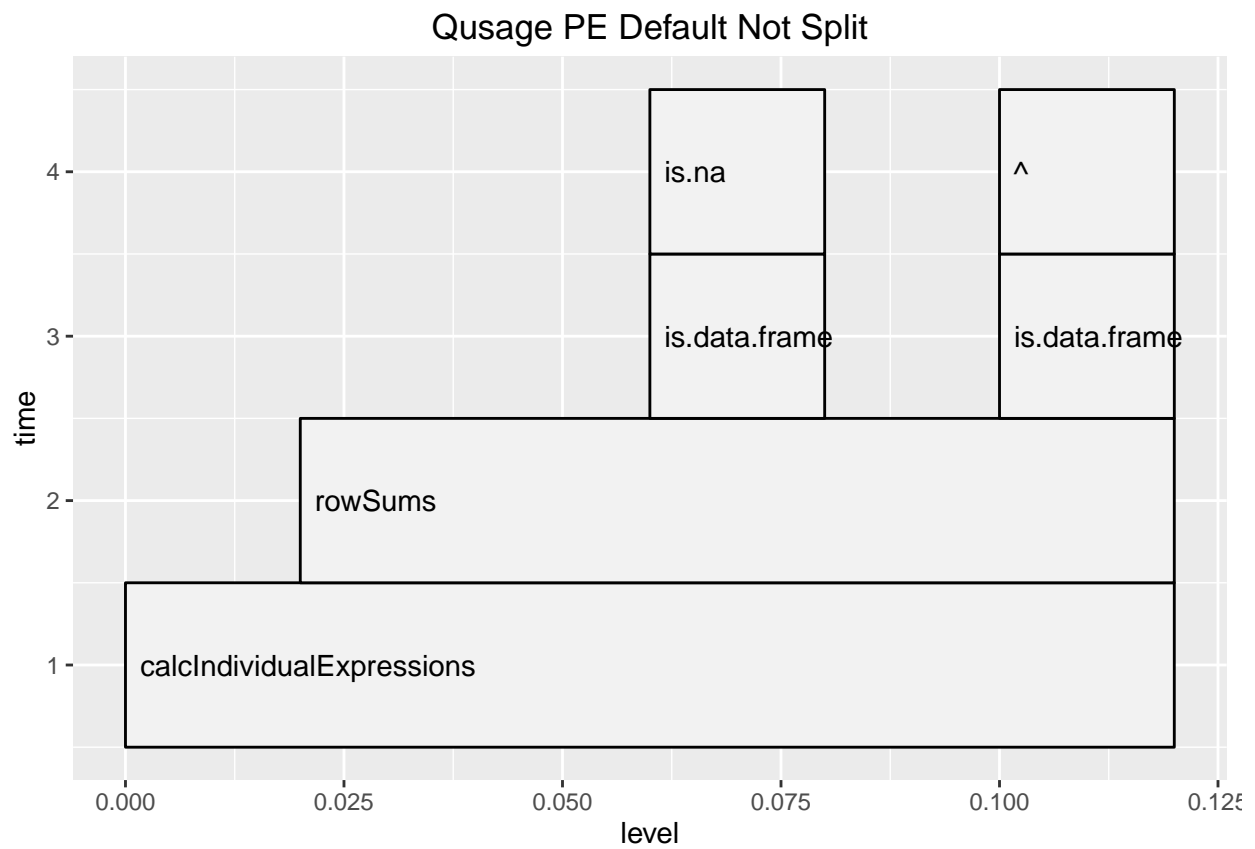


```
microbenchmark(
  original<-calcIndividualExpressions(eset.1,eset.2,paired=TRUE),
  cpp<-calcIndividualExpressionsC(eset.1,eset.2,paired=TRUE),
  arm<-calcIndividualExpressionsArm(eset.1,eset.2,paired=TRUE))

## 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
## 139.41153 143.34426 157.52478 145.1537 149.77613 210.3631   100   c
## 122.78354 125.85071 134.36786 128.7505 131.30281 195.2012   100   b
##   86.19445  89.12938  95.54723  91.0114  93.85477 157.2971   100   a
```

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

yy<-profr(calcIndividualExpressions(eset.1,eset.2,paired=TRUE))
ggplot(yy) + labs(title="Qusage PE Default Not Split")
```



```
tt<-profr(calcIndividualExpressionsArm(eset.1,eset.2,paired=TRUE))
ggplot(tt)+ labs(title="Qusage PE in Arm Not 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

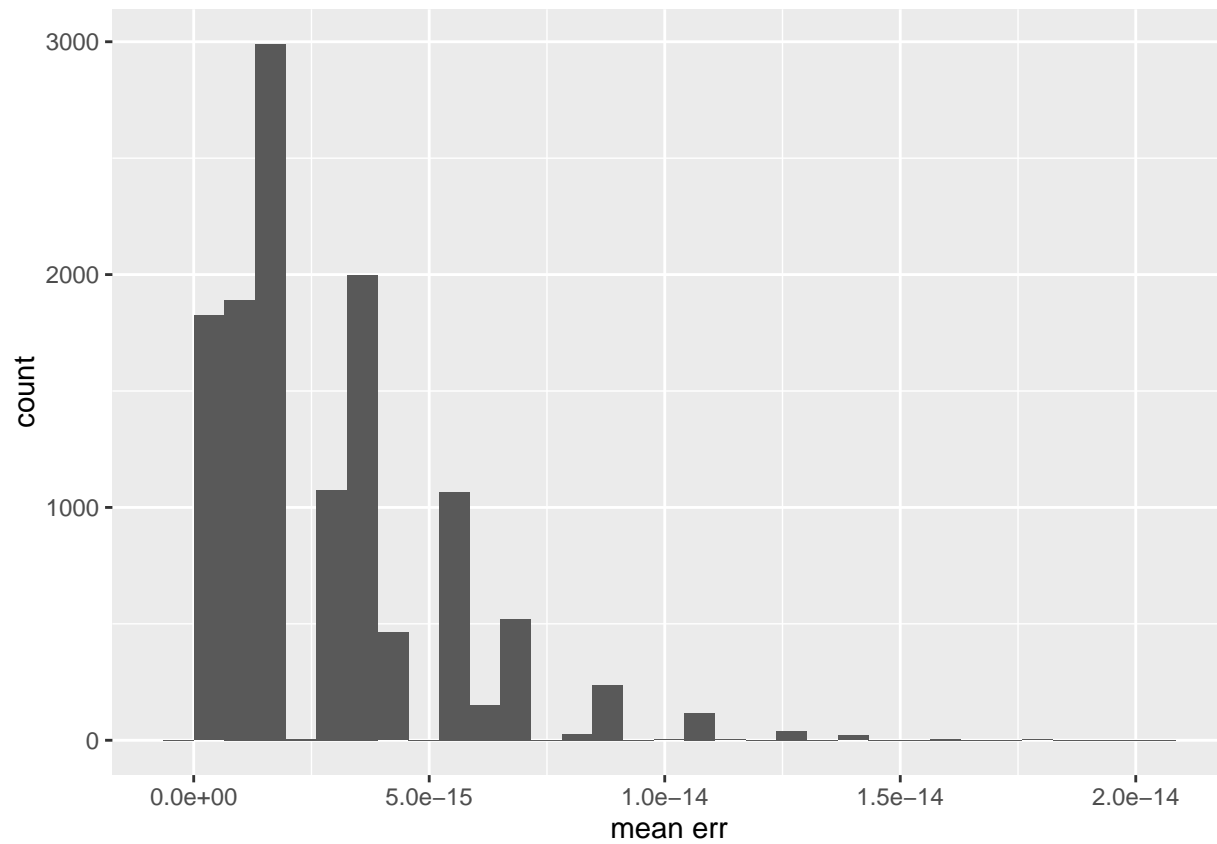
```
library(microbenchmark)
library(profr)
library(ggplot2)
library(Rcpp)
eset.1<-system.file("extdata","eset.1.RData",package="speedSage")
eset.2<-system.file("extdata","eset.2.RData",package="speedSage")
load(eset.1)
load(eset.2)
ncol(eset.1) #split by label
```

```
## [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")
sourceCpp(file="/home/anthonycolombo/Documents/qusage/qusage_repos/qusage_speed/R/sigmaSingle.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]]))
```

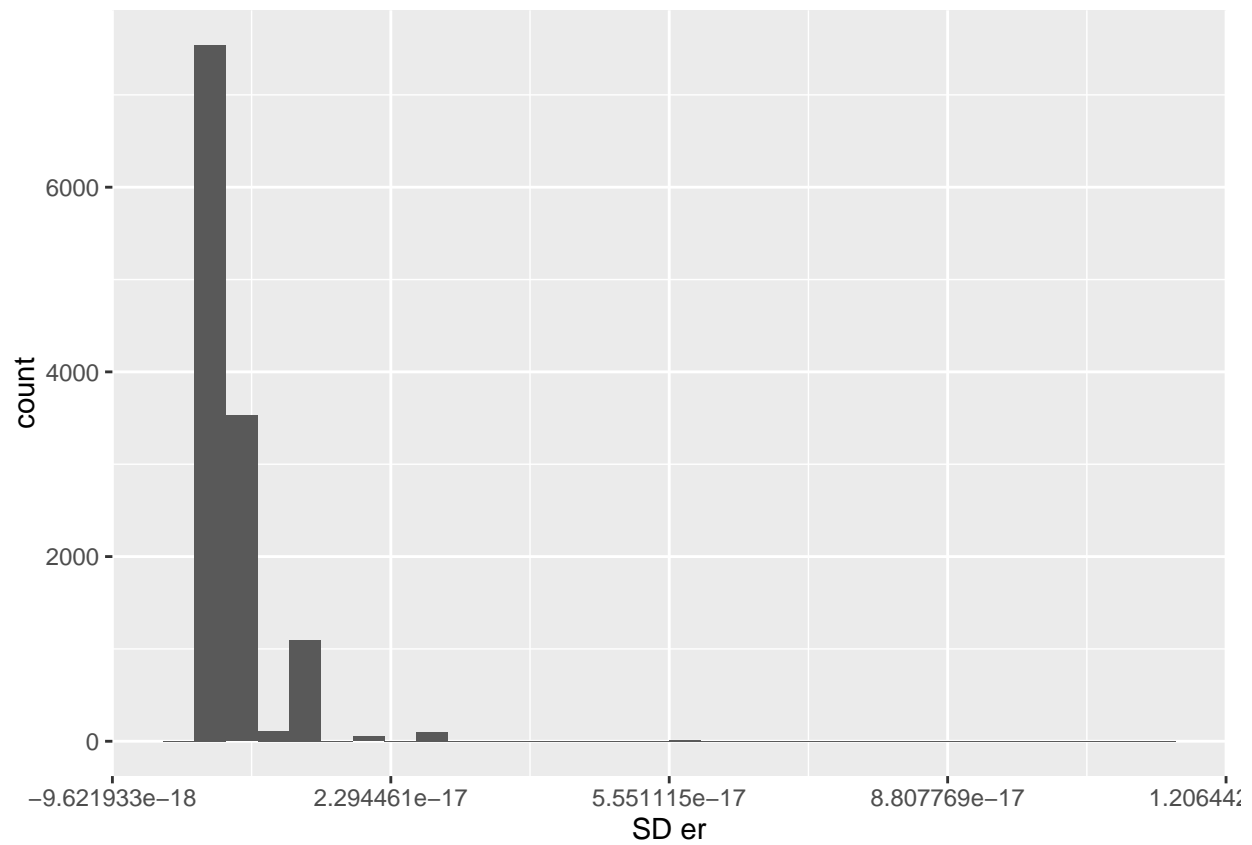
```
e3<-(abs(original[[3]]-arm[[3]]))
e4<-(abs(original[[4]]-arm[[4]]))
qplot(as.vector(e1), xlab="mean err")
```

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



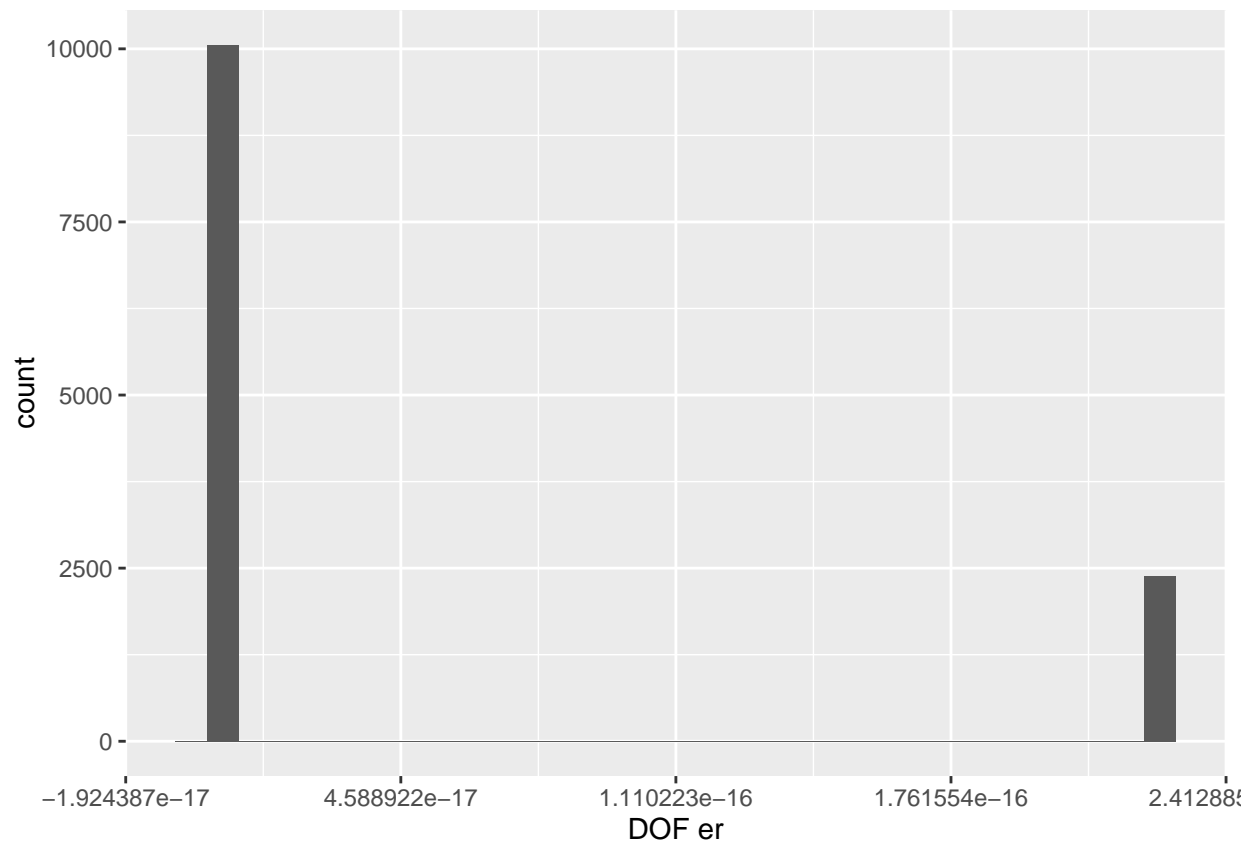
```
qplot(as.vector(e2), xlab="SD er")
```

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



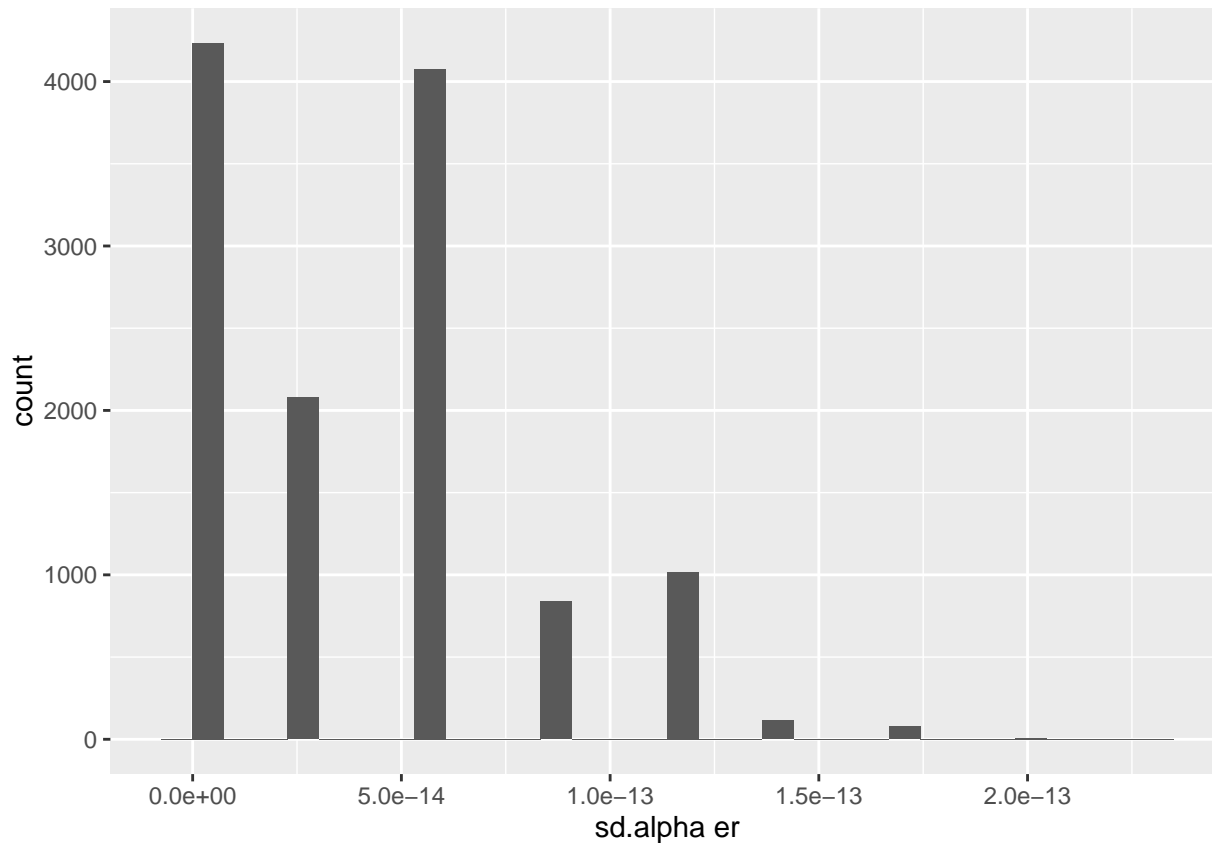
```
qplot(as.vector(e3), xlab="DOF er")
```

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



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

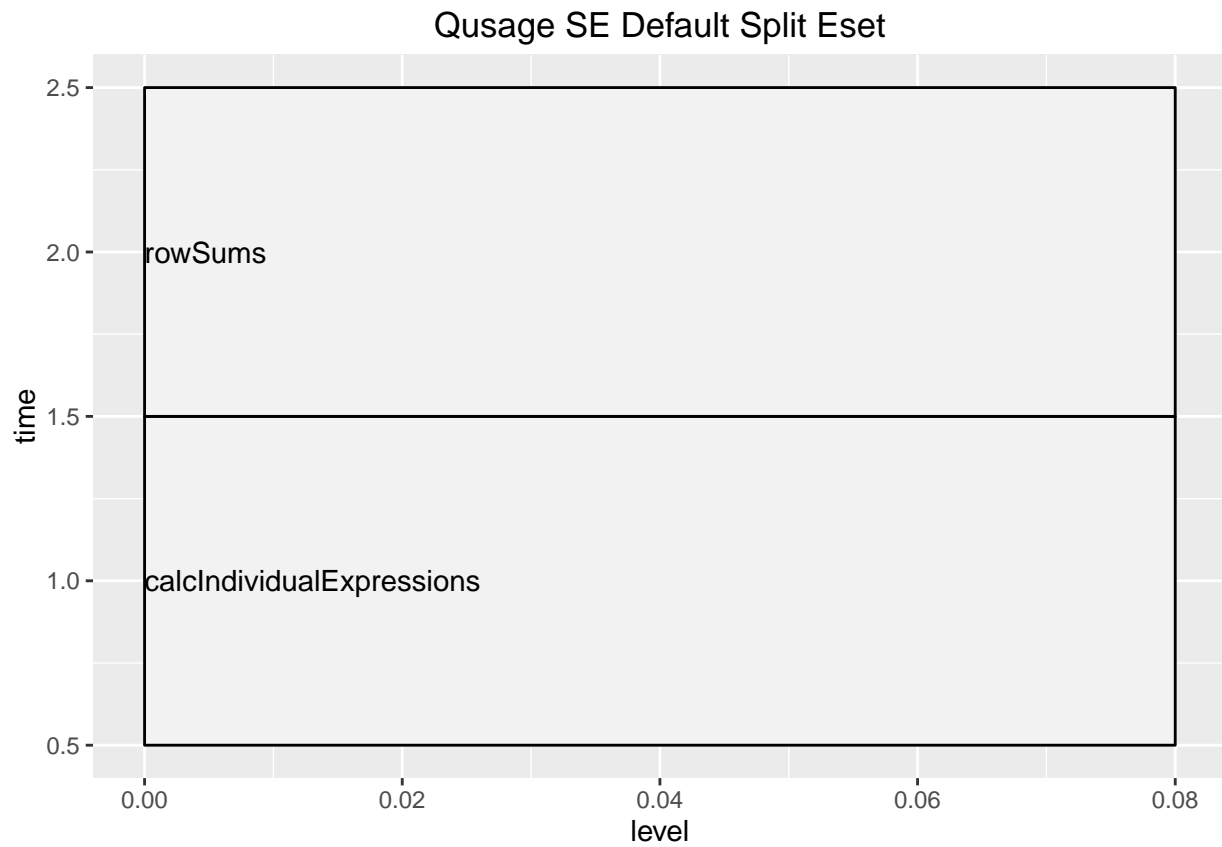
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
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
## 86.36741 86.85053 89.31846 88.49361 89.89950 151.05726   100   c
## 61.40887 61.81566 62.89255 62.10877 63.78194  67.08721   100   b
## 41.78373 42.09041 44.38603 42.86899 45.01508 106.62435   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")
```

