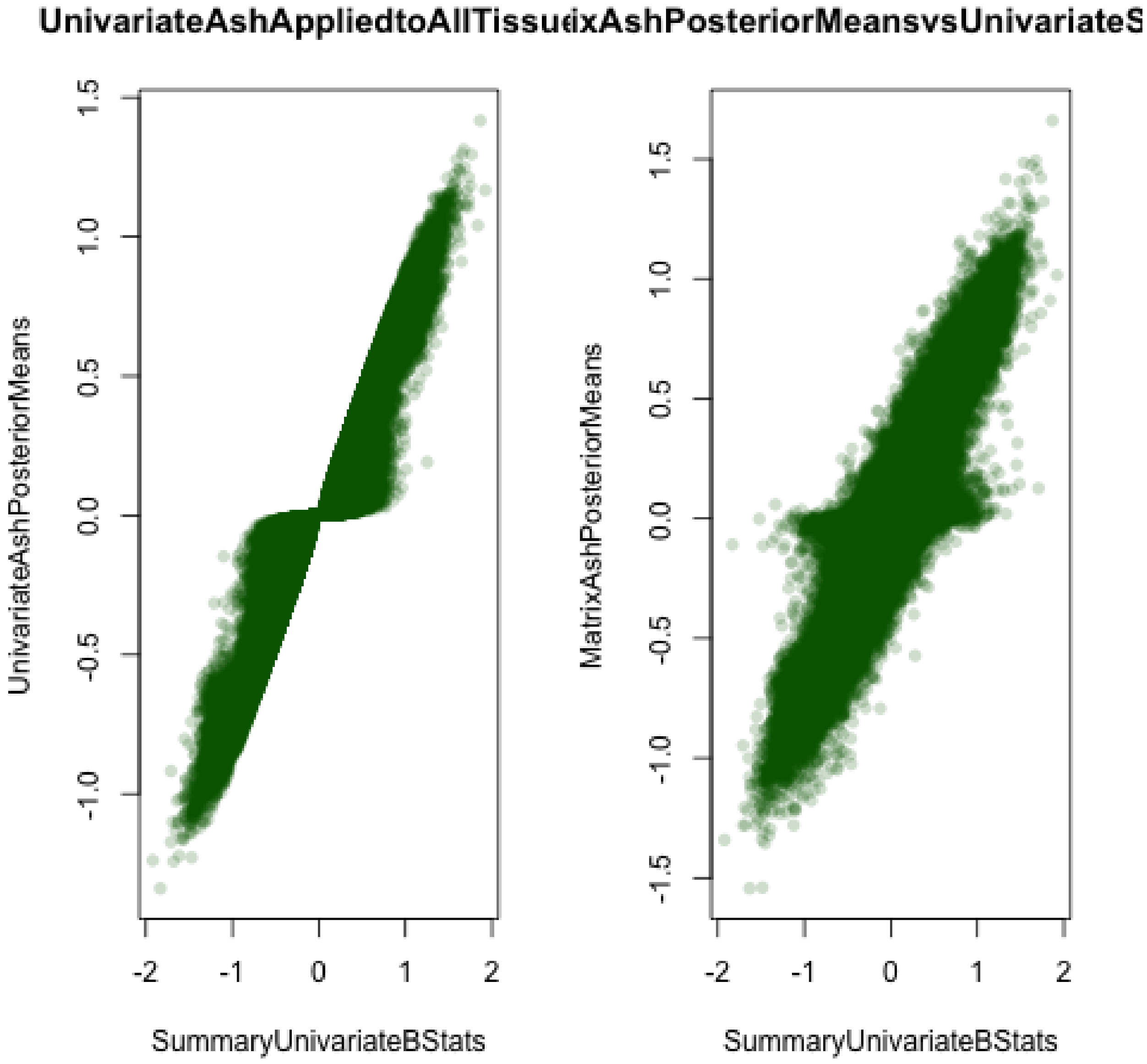
ComparingwithUnivariateAsh_maxb

In this document I compare with univariate ash

```
## pdf
##   2
```



```
## [1] "/Users/sarahurbut/gtexresults_matrixash/Analysis"
```

We can consider the number of statistics at each threshold using both methods:

```
thresh=0.05
sum(lfsr.mash<thresh)
```

```
## [1] 393414
```

```
sum(lfsr.ash<thresh)
```

```
## [1] 91755
```

We can see that Matrix ASH identifies 4.2876573 more associations than using univariate ash. Furthemore, if we consider the number of genes with at least one LFSR less than threshold. Using

```
gene.func=function(lfsr,thresh){
sigmat=lfsr<thresh
sum(rowSums(sigmat)!=0)}

gene.func(lfsr.mash,thresh=0.05)
```

```
## [1] 14146
```

```
gene.func(lfsr.ash,thresh=0.05)
```

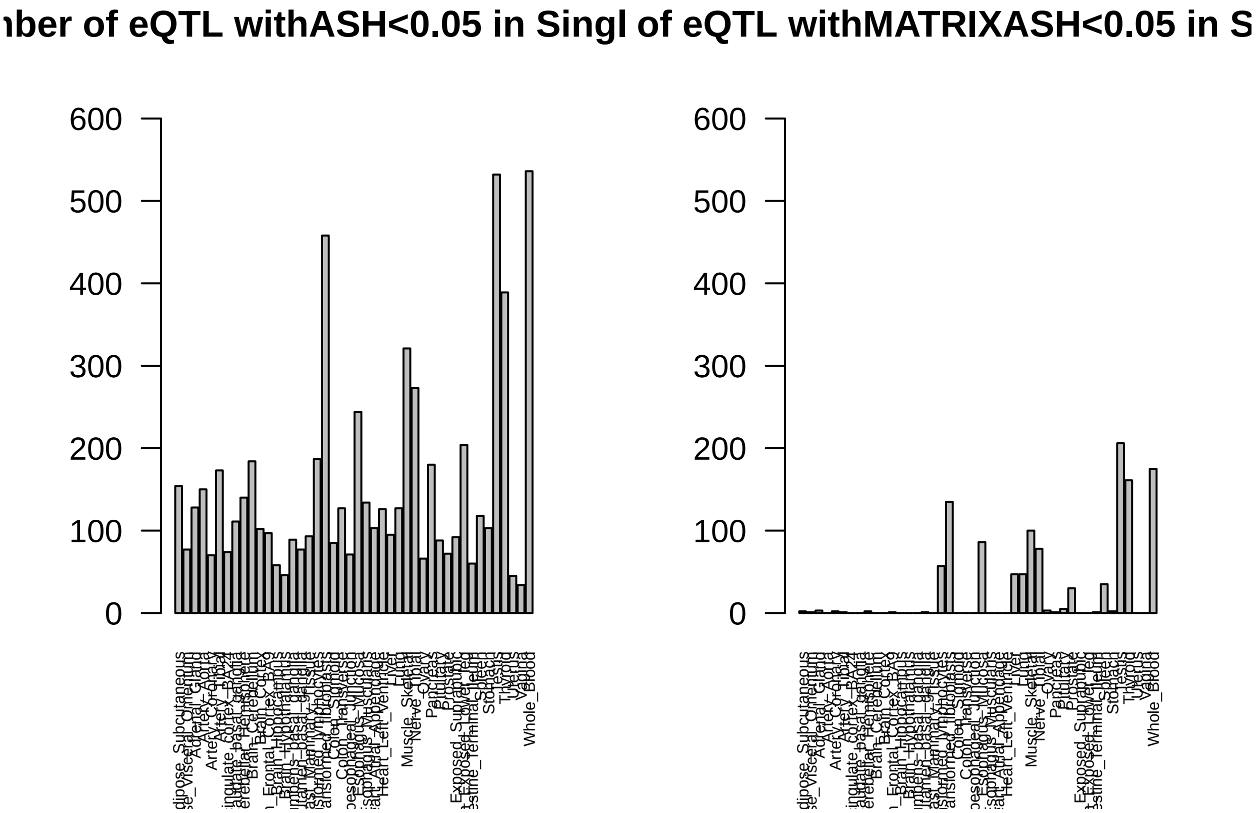
```
## [1] 14645
```

```
sum(lfsr.mash<thresh&lfsr.ash>thresh)
```

```
## [1] 303106
```

Let's consider how are bias towards or against tissue speicificy changes:

```
par(mfrow=c(1,2))
plot_singletontissues(lfsr.ash,thresh,method="ASH")
plot_singletontissues(lfsr.mash,thresh,method="MATRIXASH")
```



Let's also count how many associations we might count as inconsistent by simply counting the number of times the signs differed in a vector of posterior means for a given gene snp pair:

```
inconsistent.func=function(posterior.means,lfsr,thresh=0.05){
h=apply(posterior.means,1,function(p){
  pos=sum(p>0);neg=sum(p<0);pos*neg!=0})
sum(h=="TRUE")}

inconsistent.func(pm.ash.beta,lfsr.ash)
```

```
## [1] 14557
```

```
inconsistent.func(pm.mash.beta,lfsr.mash)
```

```
## [1] 9597
```

If we restrict our analysis to only those considered significant at an lfsr threshold using ash computed LFSR, we have a smaller number, but this is simply because so many fewer associations are called significant using ash. Using the MASH lfsrs, the results are identical to using the univariate z statistics. WE can compare to the matrix ash posterior means as well.

```
 thresh=0.05



z=sapply(seq(1:nrow(pm.ash.beta)),function(x){
```

```
l=lfsr.mash[x,];p=pm.ash.beta[x,];plow=p[which(l<thresh)];##grab only those posterior means that are 's
if(length(plow)==0){return("FALSE")}##for ones who show no significants, they can't be heterogenous
else{pos=sum(plow>0);neg=sum(plow<0);pos*neg!=0}})
sum(z==TRUE)
```

```
## [1] 9655
```

```
z=sapply(seq(1:nrow(pm.mash.beta)),function(x){
l=lfsr.mash[x,];p=pm.mash.beta[x,];plow=p[which(l<thresh)];##grab only those posterior means that are '
if(length(plow)==0){return("FALSE")}##for ones who show no significants, they can't be heterogenous
else{pos=sum(plow>0);neg=sum(plow<0);pos*neg!=0}})
sum(z==TRUE)
```

```
## [1] 3180
```

“‘

We can also consider a histogram of normalized posterior means:

```
maxb=read.table("../Data/maxbetahats.txt")
bnorm=het.norm(effectsize = maxb)
sum(bnorm>0)/length(bnorm)
```

```
## [1] 0.7144276
```

```
uni.ash.norm=het.norm(effectsize = pm.ash.beta)
sum(uni.ash.norm>0)/length(uni.ash.norm)
```

```
## [1] 0.7170696
```

```
matrix.ash.norm=het.norm(effectsize = pm.mash.beta)
sum(matrix.ash.norm>0)/length(matrix.ash.norm)
```

```
## [1] 0.8330297
```

We might also be interested in the proportion that are greater than a particular threshold of max effect. This

```
sum(bnorm>0.5)/length(bnorm)
```

```
## [1] 0.1899479
```

```
sum(uni.ash.norm>0.5)/length(uni.ash.norm)
```

```
## [1] 0.1163166
```

```
sum(matrix.ash.norm>0.5)/length(matrix.ash.norm)
```

```
## [1] 0.3540626
```

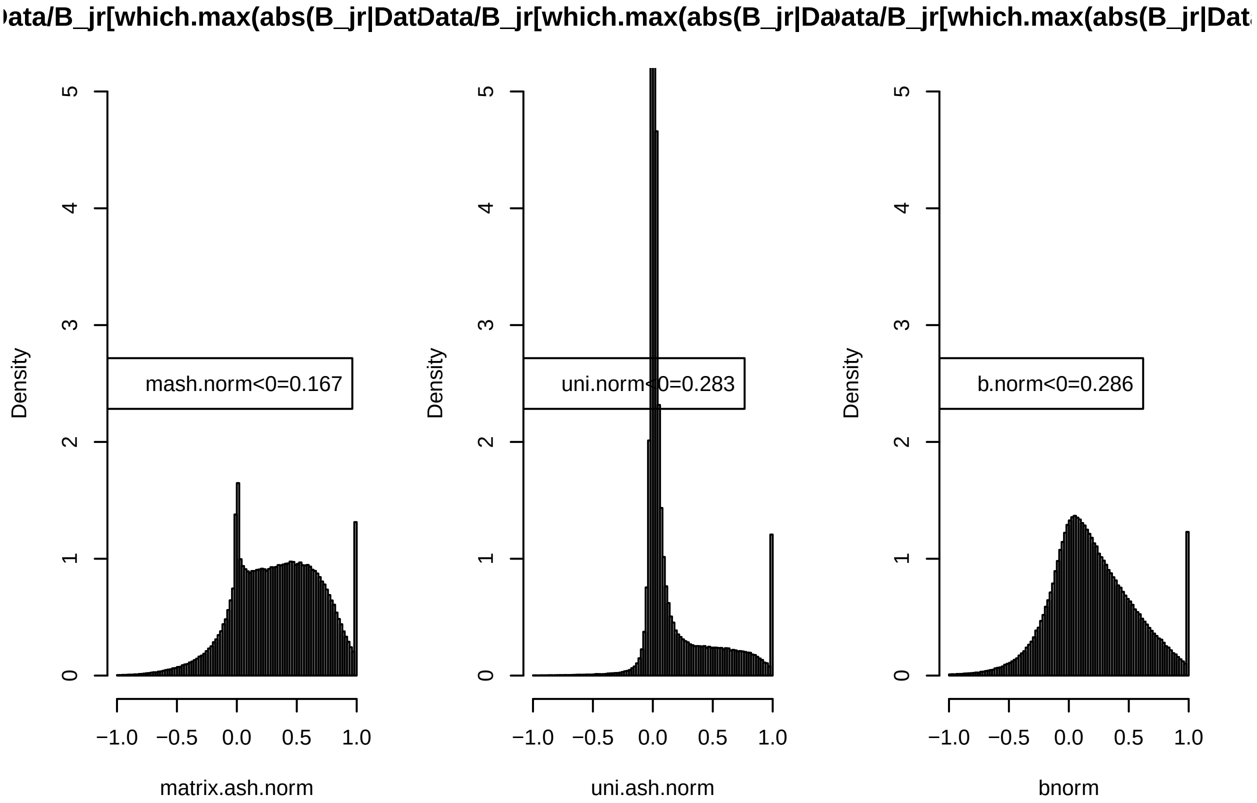
Now, let’s plot:

```r
par(mfrow=c(1,3))

hist(matrix.ash.norm,freq=FALSE,ylim=c(0,5),nclass=100,main="B_jr|Data/B_jr[which.max(abs(B_jr|Data))],M
legend("left",legend=paste0("mash.norm<0=",round((sum(matrix.ash.norm<0)/length(matrix.ash.norm)),3)))

hist(uni.ash.norm,freq=FALSE,ylim=c(0,5),nclass=100,main="B_jr|Data/B_jr[which.max(abs(B_jr|Data))],ASH
legend("left",legend=paste0("uni.norm<0=",round((sum(uni.ash.norm<0)/length(matrix.ash.norm)),3)))

hist(bnorm,freq=FALSE,ylim=c(0,5),nclass=100,main="B_jr|Data/B_jr[which.max(abs(B_jr|Data))],Z_raw")
legend("left",legend=paste0("b.norm<0=",round((sum(bnorm<0)/length(matrix.ash.norm)),3)))
```

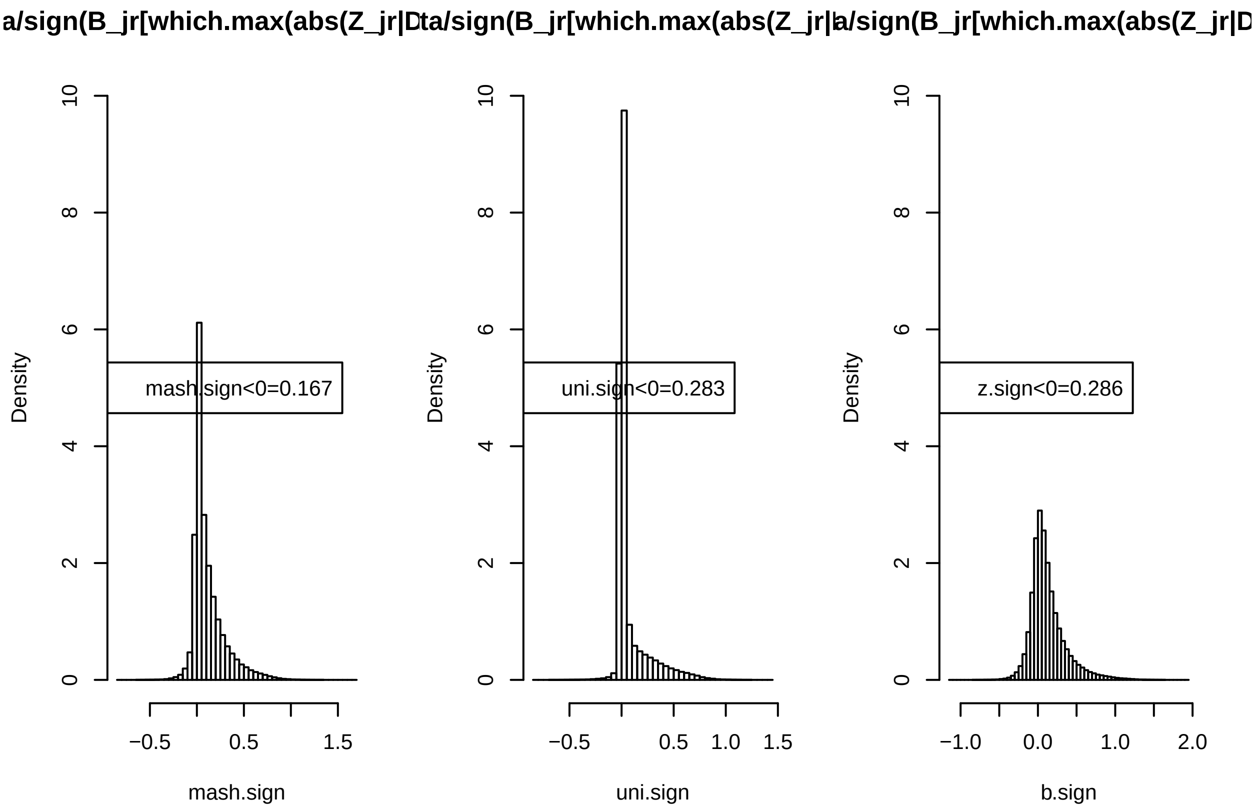


```r
mash.sign=sign.norm(effectsize = pm.mash.beta)
uni.sign=sign.norm(effectsize = pm.ash.beta)
b.sign=sign.norm(effectsize = maxb)

par(mfrow=c(1,3))
hist(mash.sign,freq=FALSE,ylim=c(0,10),main="B_jr|Data/sign(B_jr[which.max(abs(Z_jr|Data))],MASH",nclas
legend("left",legend=paste0("mash.sign<0=",round((sum(matrix.ash.norm<0)/length(matrix.ash.norm)),3)))

hist(uni.sign,freq=FALSE,ylim=c(0,10),main="B_jr|Data/sign(B_jr[which.max(abs(Z_jr|Data))],ASH",nclass=5
legend("left",legend=paste0("uni.sign<0=",round((sum(uni.ash.norm<0)/length(matrix.ash.norm)),3)))

hist(b.sign,freq=FALSE,ylim=c(0,10),main="B_jr|Data/sign(B_jr[which.max(abs(Z_jr|Data))],Z.raw)",nclass=
legend("left",legend=paste0("z.sign<0=",round((sum(bnorm<0)/length(matrix.ash.norm)),3)))
```



Understandably, there are no examples in which the sign of the posterior mean using univariate ash is flipped because there is no sharing of information across tissues, while with matrix ash, the sign changes about 18% of the time:

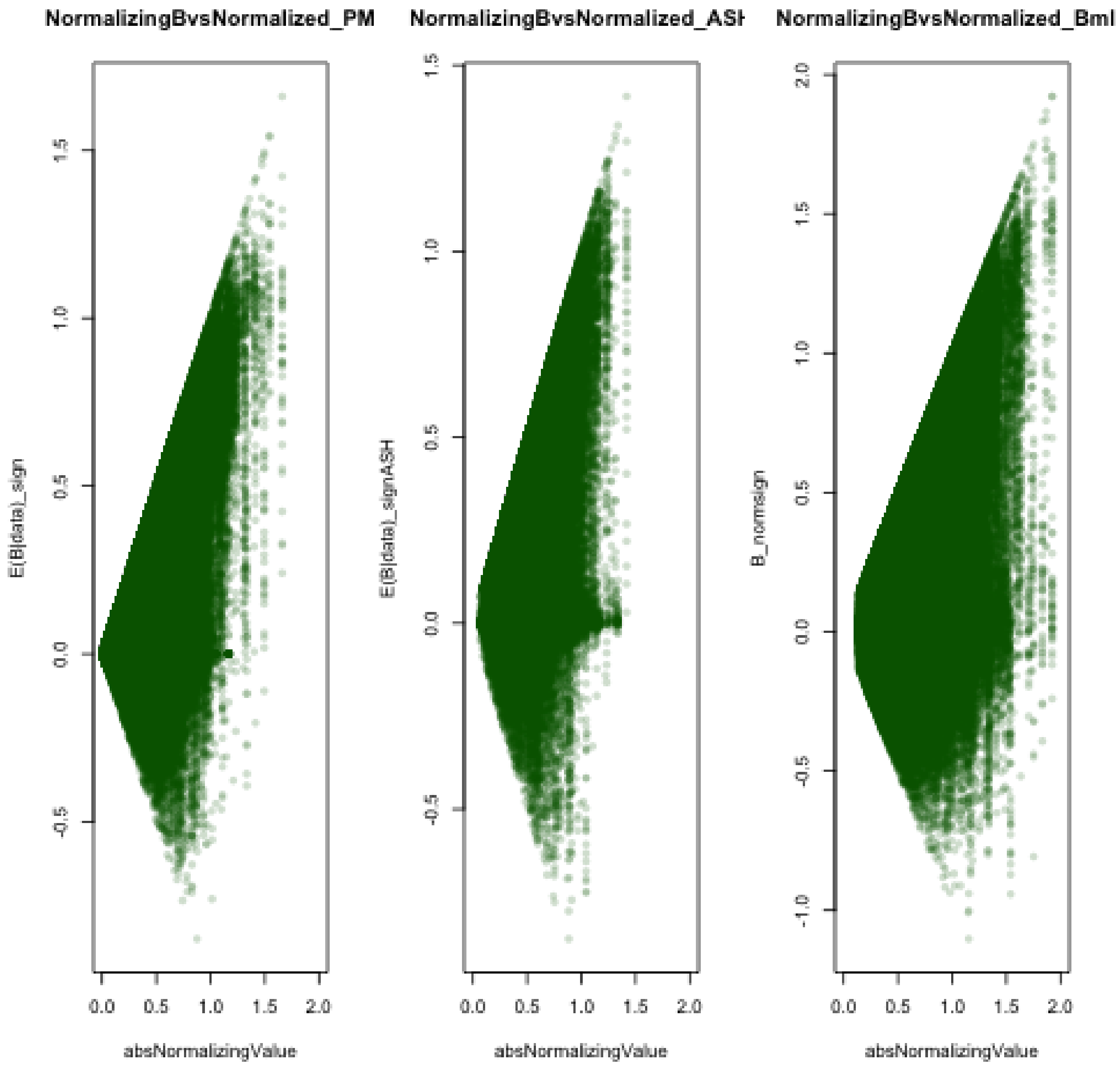
```
b.stat=maxb
sum2=b.stat*pm.mash.beta
sum(sum2<0)/length(unlist(sum2))
```

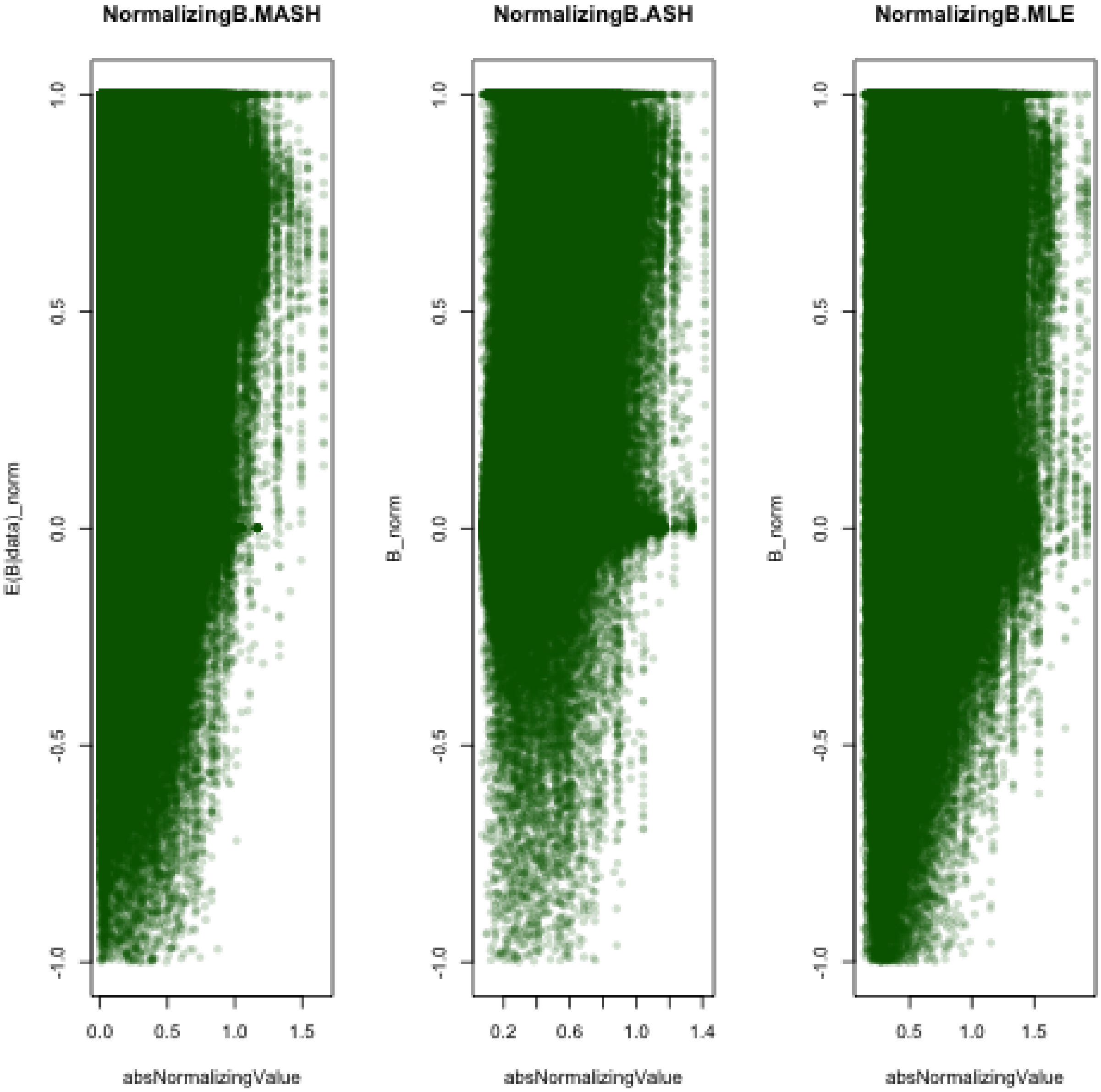
```
## [1] 0.1842254
```

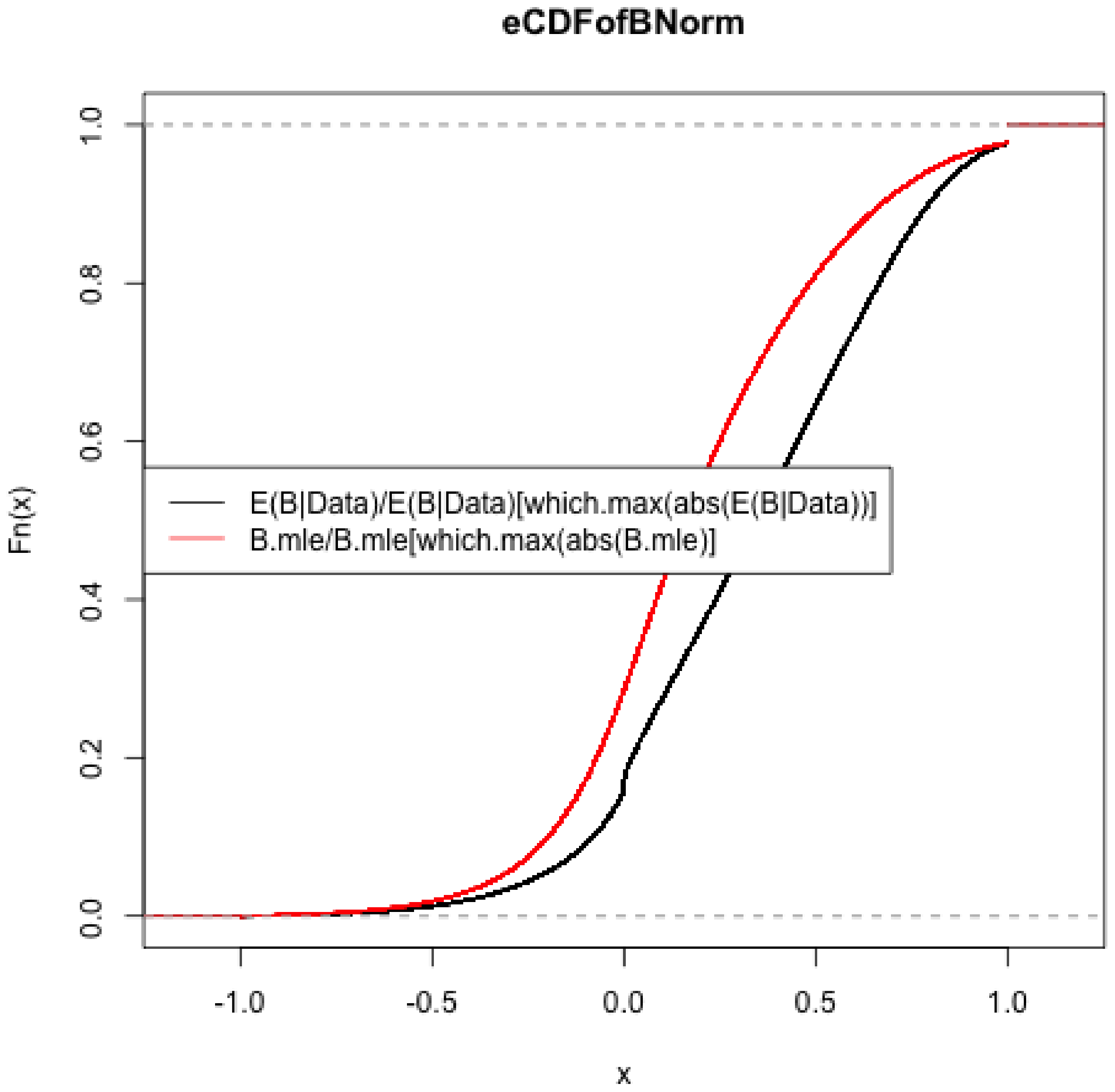
```
sumash=b.stat*pm.ash.beta
sum(sumash<0)/length(unlist(sum2))
```

```
## [1] 0
```

Let's consider the biplots of the values vs their maximums: First, I plot the biplot of the Maximum values vs their Normalized using Posterior means computed with matrix ash, univariate ash, and univariate summary statistics.


NormalizingBvsNormalized_PM
E(B|data)_sign
absNormalizingValue
-0.5
0.0
0.5
1.0
1.5
0.0
0.5
1.0
1.5
2.0
NormalizingBvsNormalized_ASI
E(B|data)_signASH
absNormalizingValue
-0.5
0.0
0.5
1.0
1.5
0.0
0.5
1.0
1.5
2.0
NormalizingBvsNormalized_Bml
B_normsign
absNormalizingValue
-1.0
-0.5
0.0
0.5
1.0
1.5
2.0
0.0
0.5
1.0
1.5
2.0


NormalizingB.MASH
E(B|data)_norm
1.0
0.5
0.0
-0.5
-1.0
0.0
0.5
1.0
1.5
absNormalizingValue
NormalizingB.ASH
B_norm
1.0
0.5
0.0
-0.5
-1.0
0.2
0.6
1.0
1.4
absNormalizingValue
NormalizingB.MLE
B_norm
1.0
0.5
0.0
-0.5
-1.0
0.5
1.0
1.5
absNormalizingValue



```
R=ncol(pm.mash.beta)
colnames(pm.mash.beta)=colnames(maxb)

max.mash=t(apply(pm.mash.beta,1,function(x){
  rep(x[which.max(abs(x))],R)
}))

max.mle=t(apply(maxb,1,function(x){
  rep(x[which.max(abs(x))],R)
}))

max.ash=t(apply(pm.ash.beta,1,function(x){
  rep(x[which.max(abs(x))],R)
}))
```

```r
png("../Figures/normstuffeb.png")
par(mfrow=c(1,3))
plot(abs(max.mash),matrix.ash.norm,main="NormalizingB.MASH",xlab="absNormalizingValue",ylab="E(B|data)_
     col=rgb(0,100,0,50,maxColorValue=255), pch=16,xlim=c(0,1))
plot(abs(max.ash),uni.ash.norm,main="NormalizingB.ASH",ylab="B_norm",xlab="absNormalizingValue",col=rgb
plot(abs(max.mle),bnorm,main="NormalizingB.MLE",ylab="B_norm",xlab="absNormalizingValue",col=rgb(0,100,
dev.off()


png("../Figures/normstuffeb_sign.png")
par(mfrow=c(1,3))
plot(abs(max.mash),mash.sign,xlim=c(0,2),main="NormalizingBvsNormalized_PM",xlab="absNormalizingValue",
     col=rgb(0,100,0,50,maxColorValue=255), pch=16)
plot(abs(max.ash),uni.sign,main="NormalizingBvsNormalized_ASH",xlim=c(0,2),ylab="E(B|data)_signASH",xla
plot(abs(max.mle),b.sign,main="NormalizingBvsNormalized_Bmle",ylab="B_normsign",xlim=c(0,2),xlab="absNo
dev.off()


png("ecdf.png")
e=ecdf(ebnorm)
plot(e,main="eCDFofBNorm")
lines(ecdf(bnorm),col="red")
legend("left",lty=1,col=c("black","red"),legend=c("E(B|Data)/E(B|Data)[which.max(abs(E(B|Data))]","B.ml
dev.off()
```