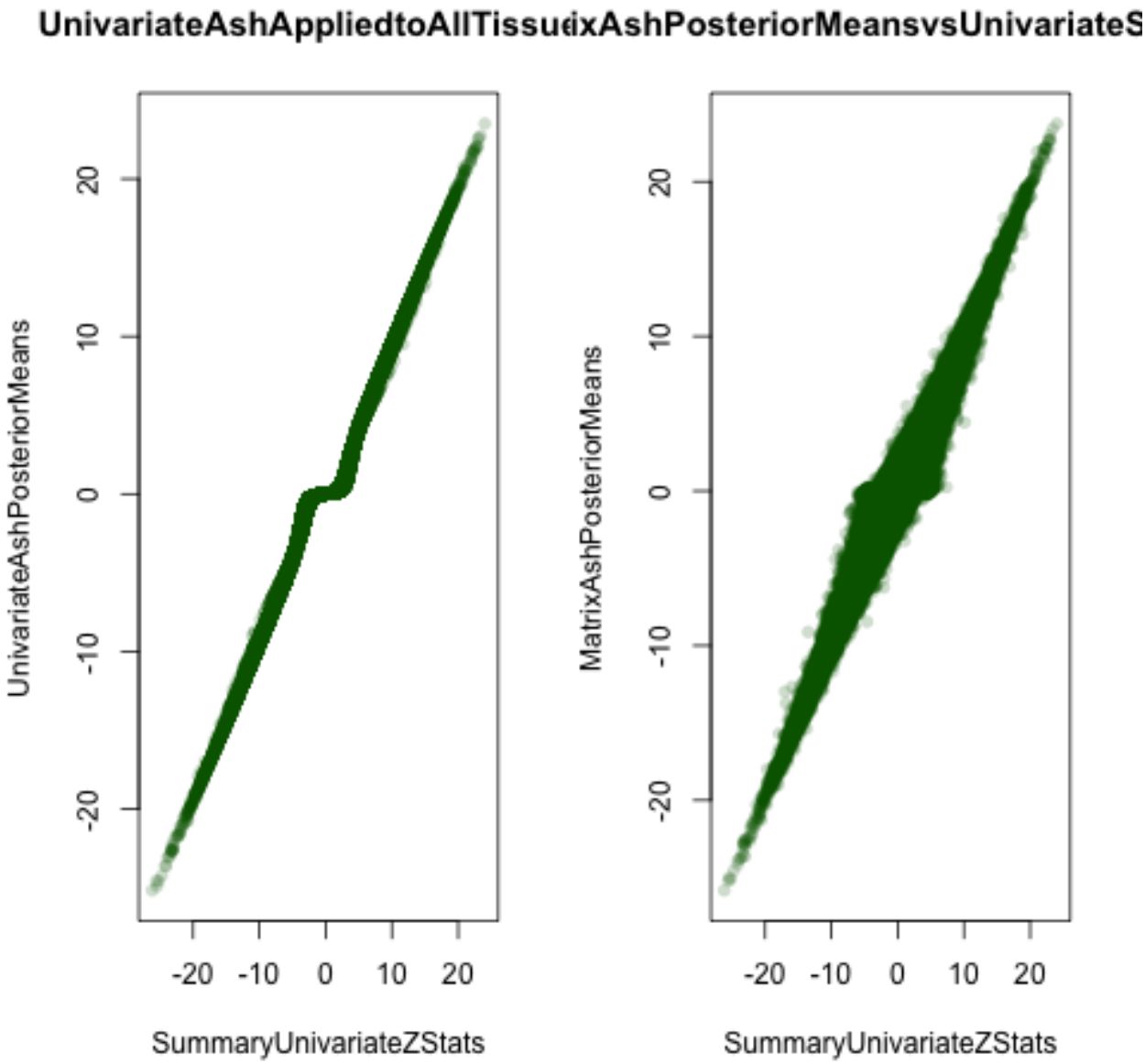
ComparingwithUnivariateAsh

In this document I compare with univariate ash

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




```
## [1] "/Users/sarahurbut/Dropbox/UnivariateAsh"
```

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)
```

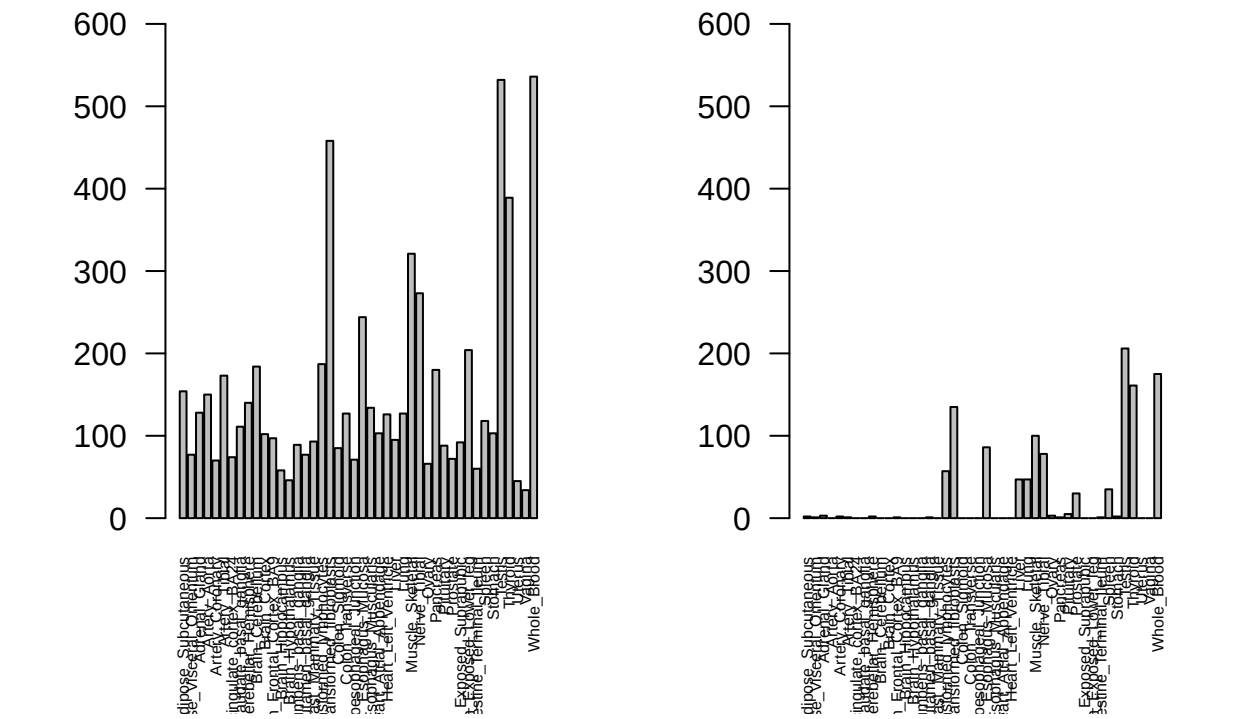
```
## [1] 14645
```

```
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,lfsr.ash)
```

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

```
inconsistent.func(pm.mash,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)),function(x){
l=lfsr.ash[x,];p=pm.ash[x,];plow=p[which(l<thresh)];##grab only those posterior means that are 'signifi
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] 984
```

```
z=sapply(seq(1:nrow(pm.ash)),function(x){
l=lfsr.mash[x,];p=pm.ash[x,];plow=p[which(l<thresh)];##grab only those posterior means that are 'signif
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)),function(x){
l=lfsr.mash[x,];p=pm.mash[x,];plow=p[which(l<thresh)];##grab only those posterior means that are 'signi
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:

```
z.stat=read.table("../Data/maxz.txt")
znorm=het.norm(effectsize = z.stat)
sum(znorm>0)/length(znorm)
```

```
## [1] 0.7180483
```

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

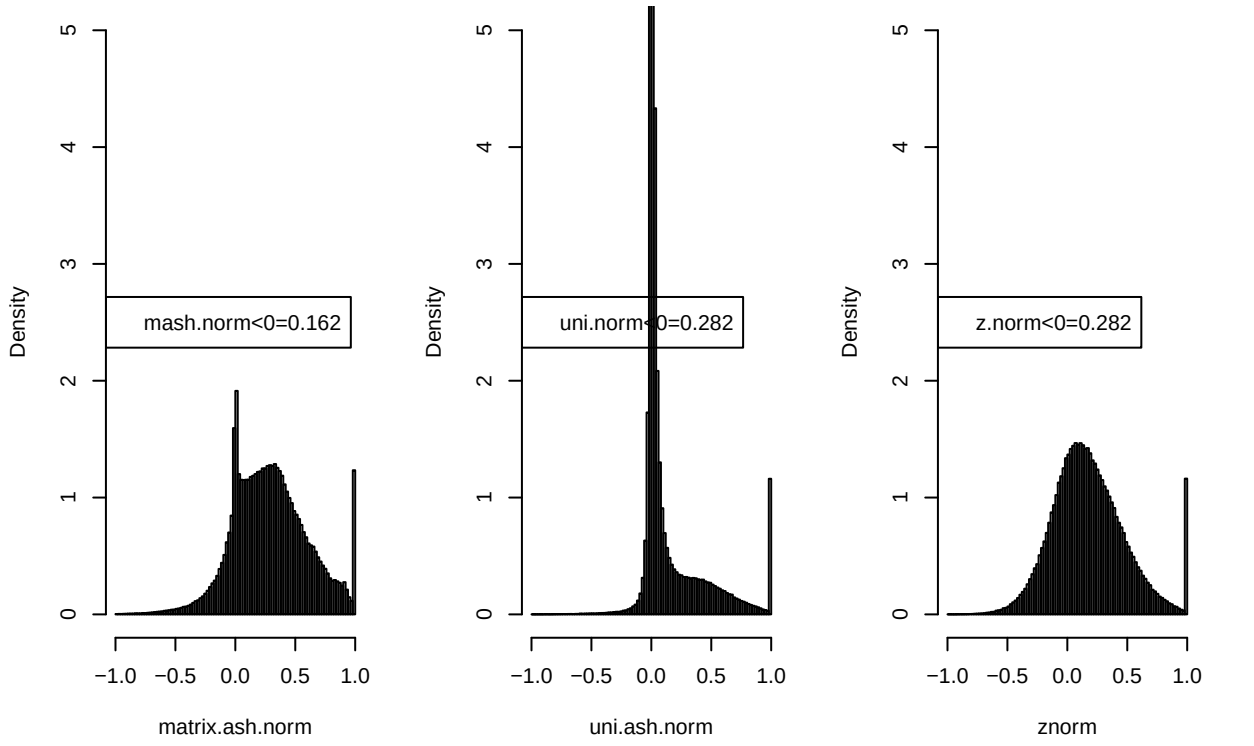
```
## [1] 0.8383845
```

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

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

```
## [1] 0.7180851
```

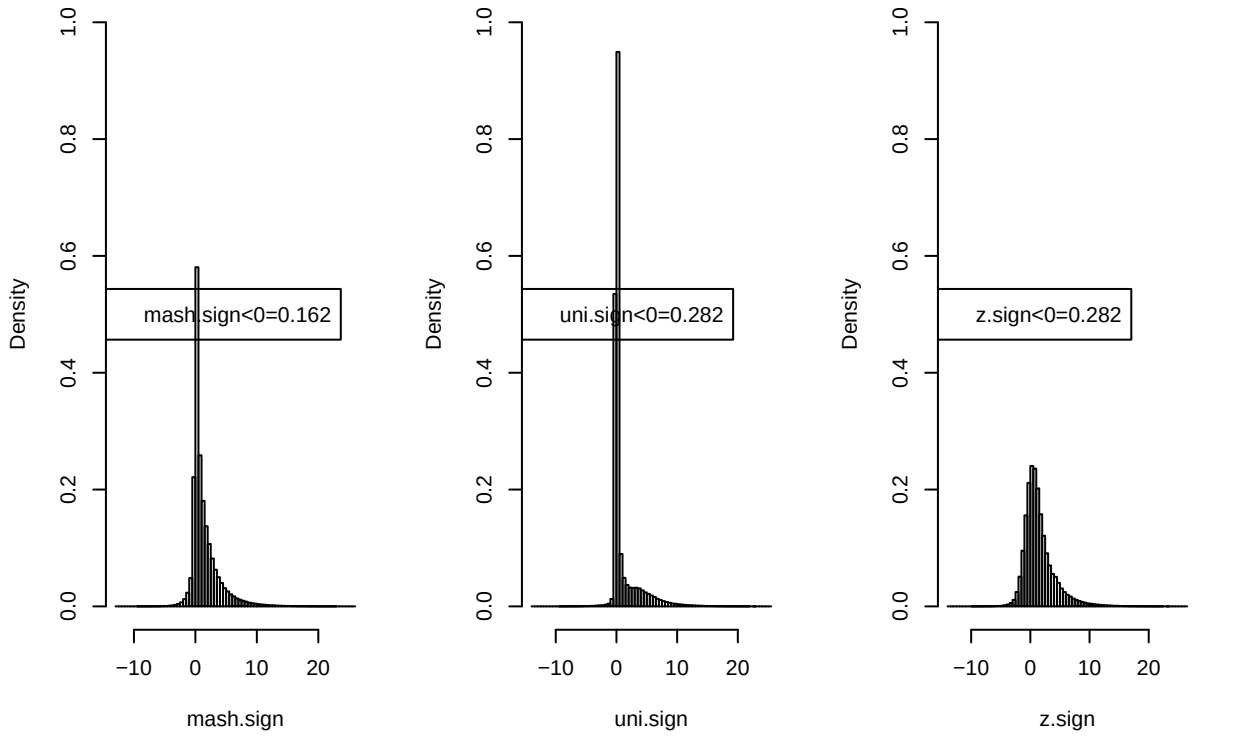
```
hist(uni.ash.norm,freq=FALSE,ylim=c(0,5),nclass=100,main="Z_jr|Data/Z_jr[which.max(abs(Z_jr|Data))],ASH
legend("left",legend=paste0("uni.norm<0=",round((sum(uni.ash.norm<0)/length(matrix.ash.norm)),3)))
hist(znorm,freq=FALSE,ylim=c(0,5),nclass=100,main="Z_jr|Data/Z_jr[which.max(abs(Z_jr|Data))],Z_raw")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




```
mash.sign=sign.norm(effectsize = pm.mash)
uni.sign=sign.norm(effectsize = pm.ash)
z.sign=sign.norm(effectsize = z.stat)

par(mfrow=c(1,3))
hist(mash.sign,freq=FALSE,ylim=c(0,1),nclass=100,main="Z_jr|Data/sign(Z_jr[which.max(abs(Z_jr|Data))],MA
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(0,1),nclass=100,main="Z_jr|Data/sign(Z_jr[which.max(abs(Z_jr|Data))],ASH
legend("left",legend=paste0("uni.sign<0=",round((sum(uni.ash.norm<0)/length(matrix.ash.norm)),3)))

hist(z.sign,freq=FALSE,ylim=c(0,1),nclass=100,main="Z_jr|Data/sign(Z_jr[which.max(abs(Z_jr|Data))],Z.raw
legend("left",legend=paste0("z.sign<0=",round((sum(znorm<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:

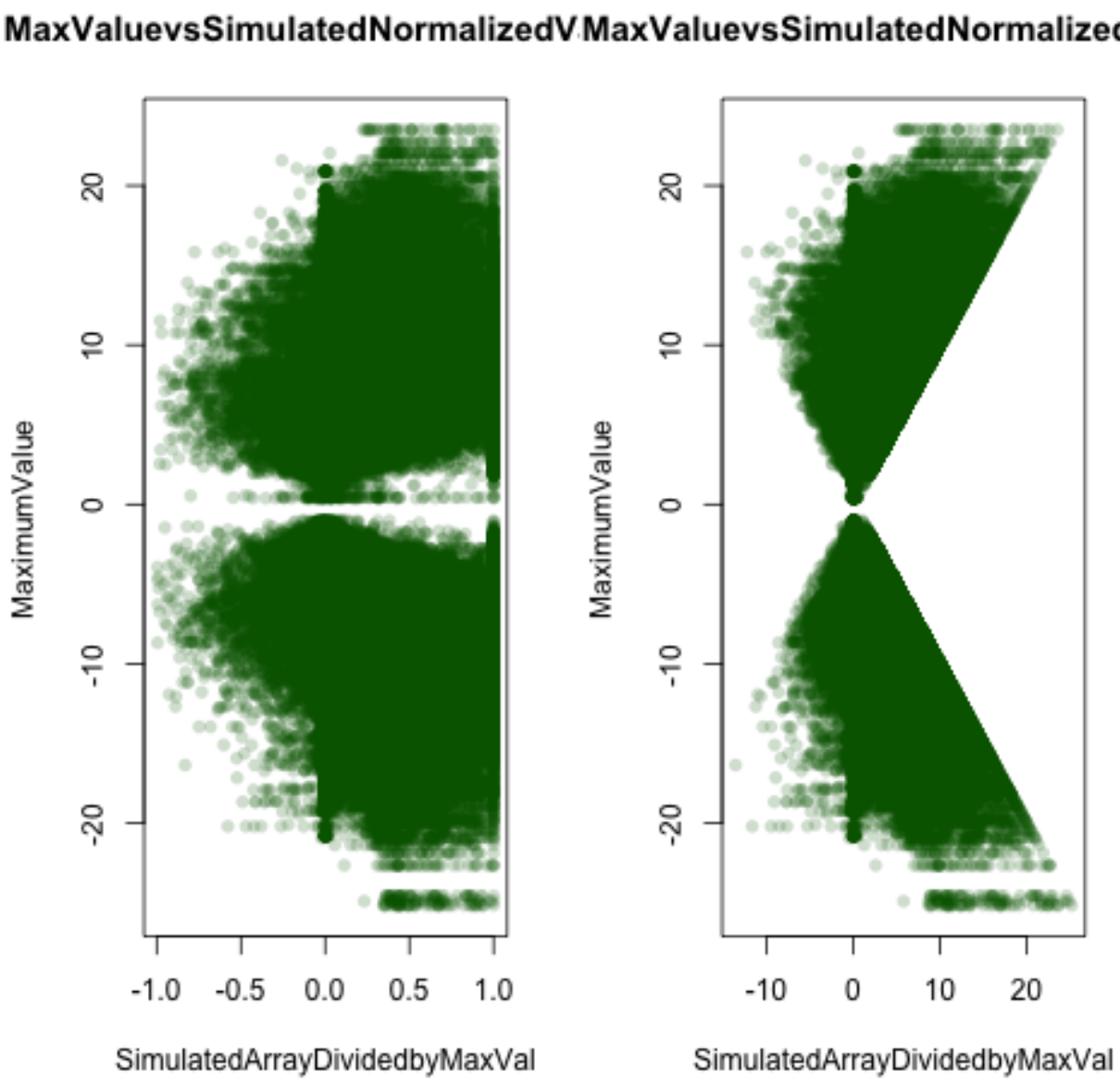
```
sum2=z.stat*pm.mash
sum(sum2<0)/length(unlist(sum2))
```

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

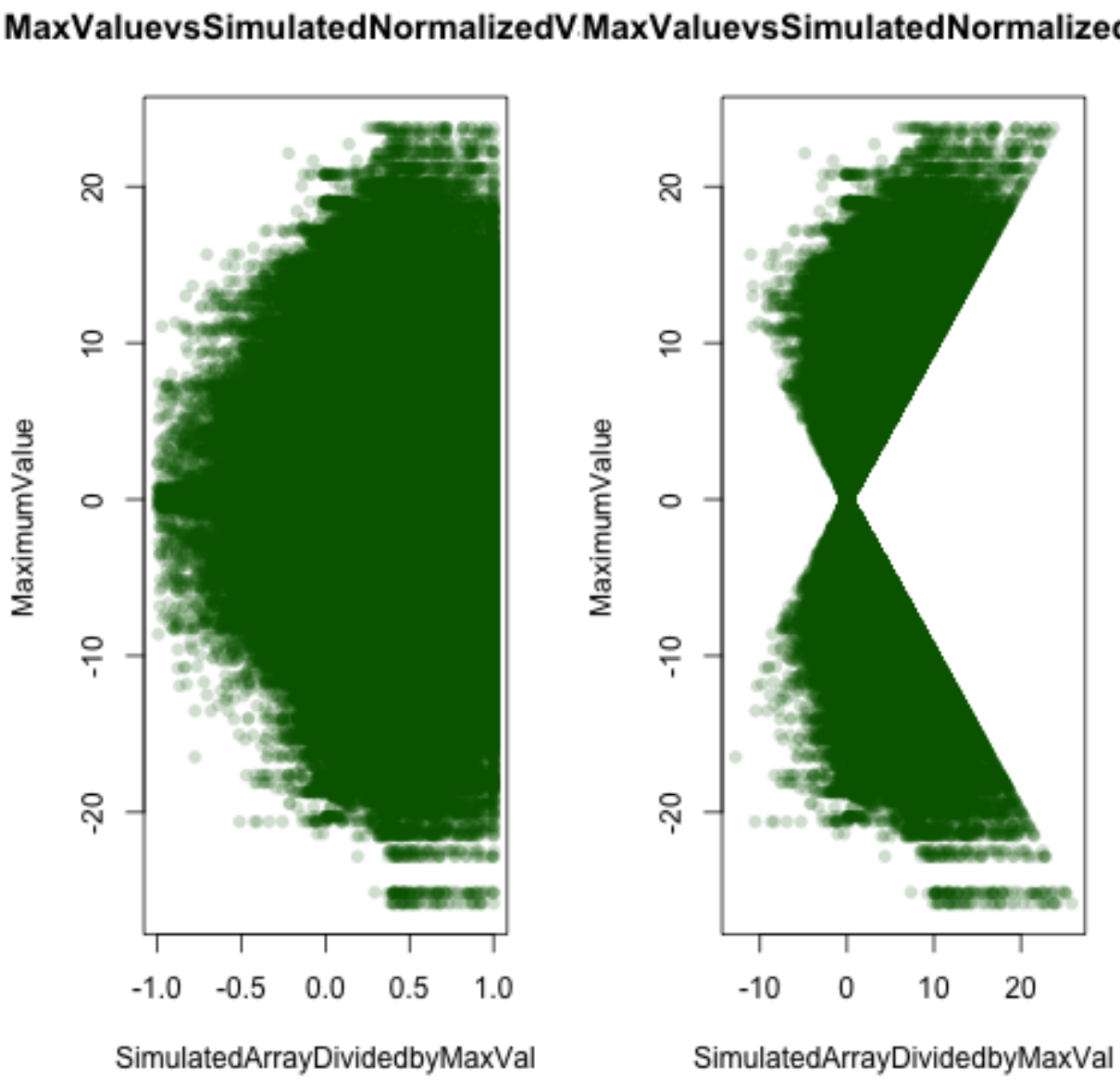
```
sumash=z.stat*pm.ash
sum(sumash<0)/length(unlist(sum2))
```

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

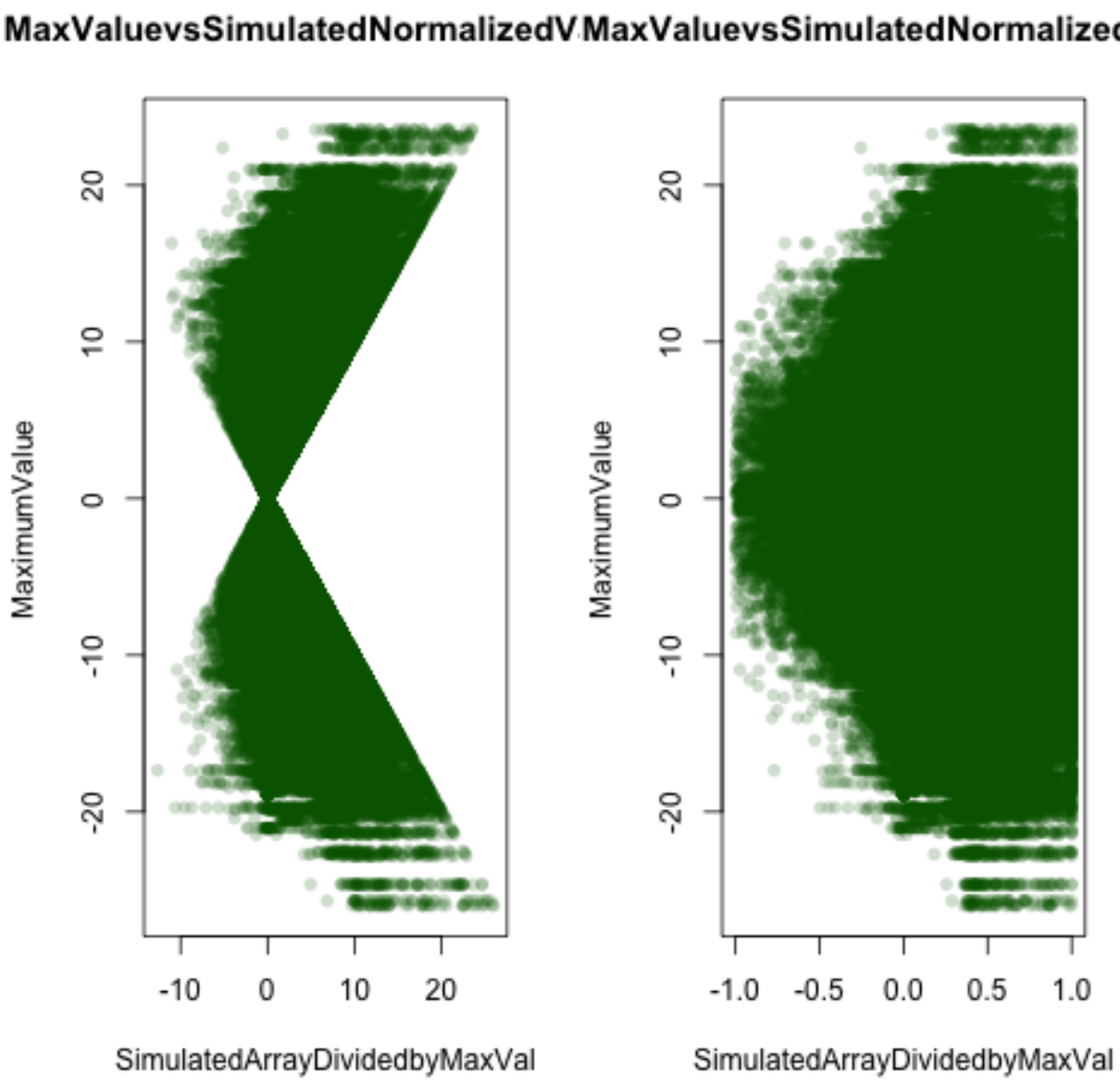
Let's consider the biplots of the values vs their maximums: First, I plot the biplot of the posterior means using univariate ash vs their maximums:



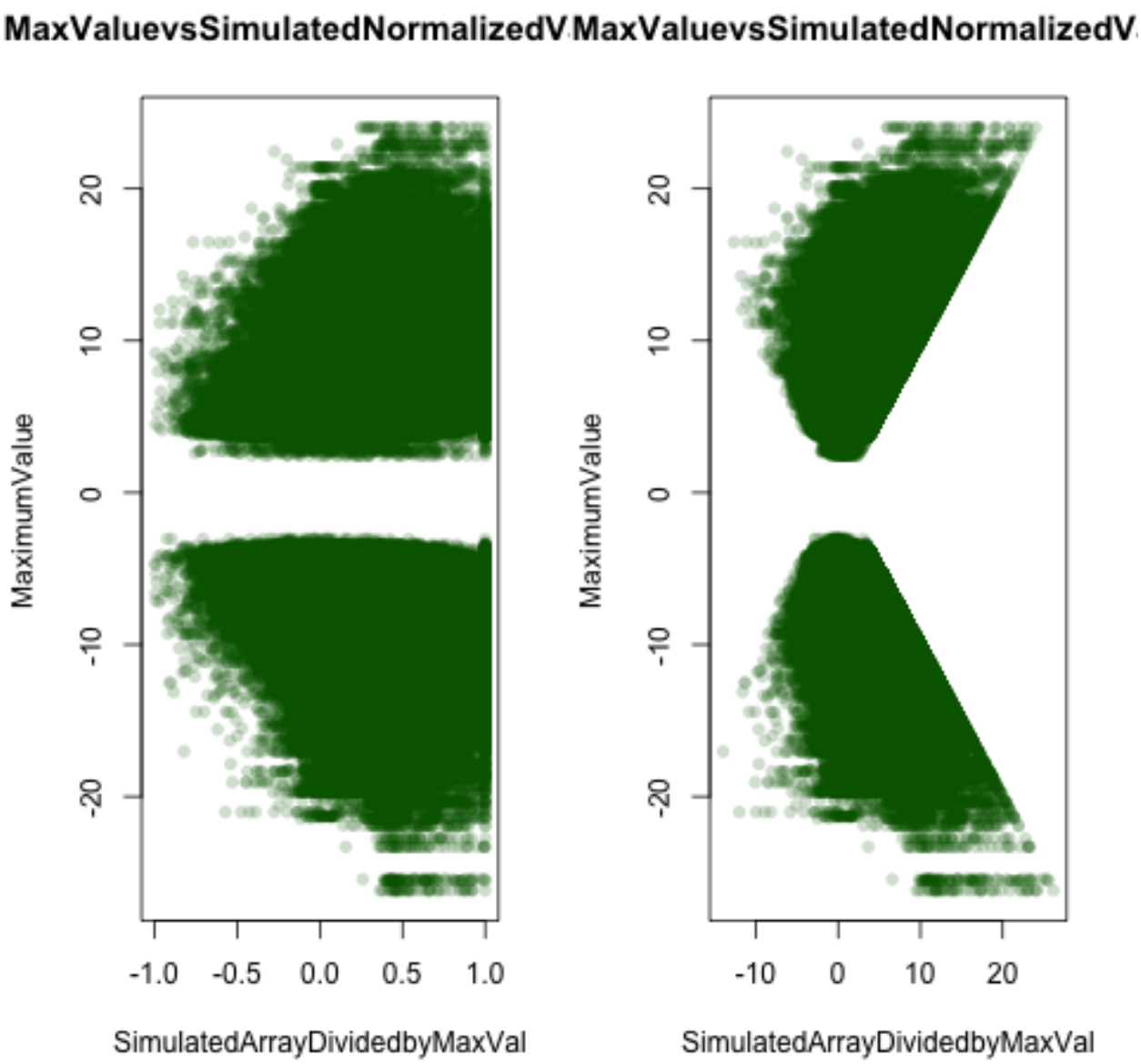
Then, I make the same plot using the posterior means using matrix ash vs their maximums:



I verify that this is the same as the simulated values with matrix vs their maximums:



And I also plot the univariate z stats vs their maximums:




```
rep.col<-function(x,n){
   matrix(rep(x,each=n), ncol=n, byrow=TRUE)
}
bi_plot_funcmaxz=function(max.val,znorm){
 a=rep.col(max.val,44)
  plot(znorm,a,main="MaxValuevsSimulatedNormalizedValue",xlab="SimulatedArrayDividedbyMaxVal",ylab="Maxi
  }

max.val=apply(pm.ash,1,function(x){
  x[which.max(abs(x))]
})


png("uni.ash.biplot.png")
```

```
par(mfrow=c(1,2))
bi_plot_funcmaxz(max.val = max.val,znorm = uni.ash.norm)
bi_plot_funcmaxz(max.val = max.val,znorm = uni.sign)
dev.off()



max.val=apply(pm.mash,1,function(x){
  x[which.max(abs(x))]
})

png("mash.biplot.png")
par(mfrow=c(1,2))
bi_plot_funcmaxz(max.val = max.val,znorm = matrix.ash.norm)
bi_plot_funcmaxz(max.val = max.val,znorm = mash.sign)
dev.off()



max.val=apply(z.stat,1,function(x){
  x[which.max(abs(x))]
})

png("z.biplot.png")
par(mfrow=c(1,2))
bi_plot_funcmaxz(max.val = max.val,znorm = znorm)
bi_plot_funcmaxz(max.val = max.val,znorm = z.sign)
dev.off()
```