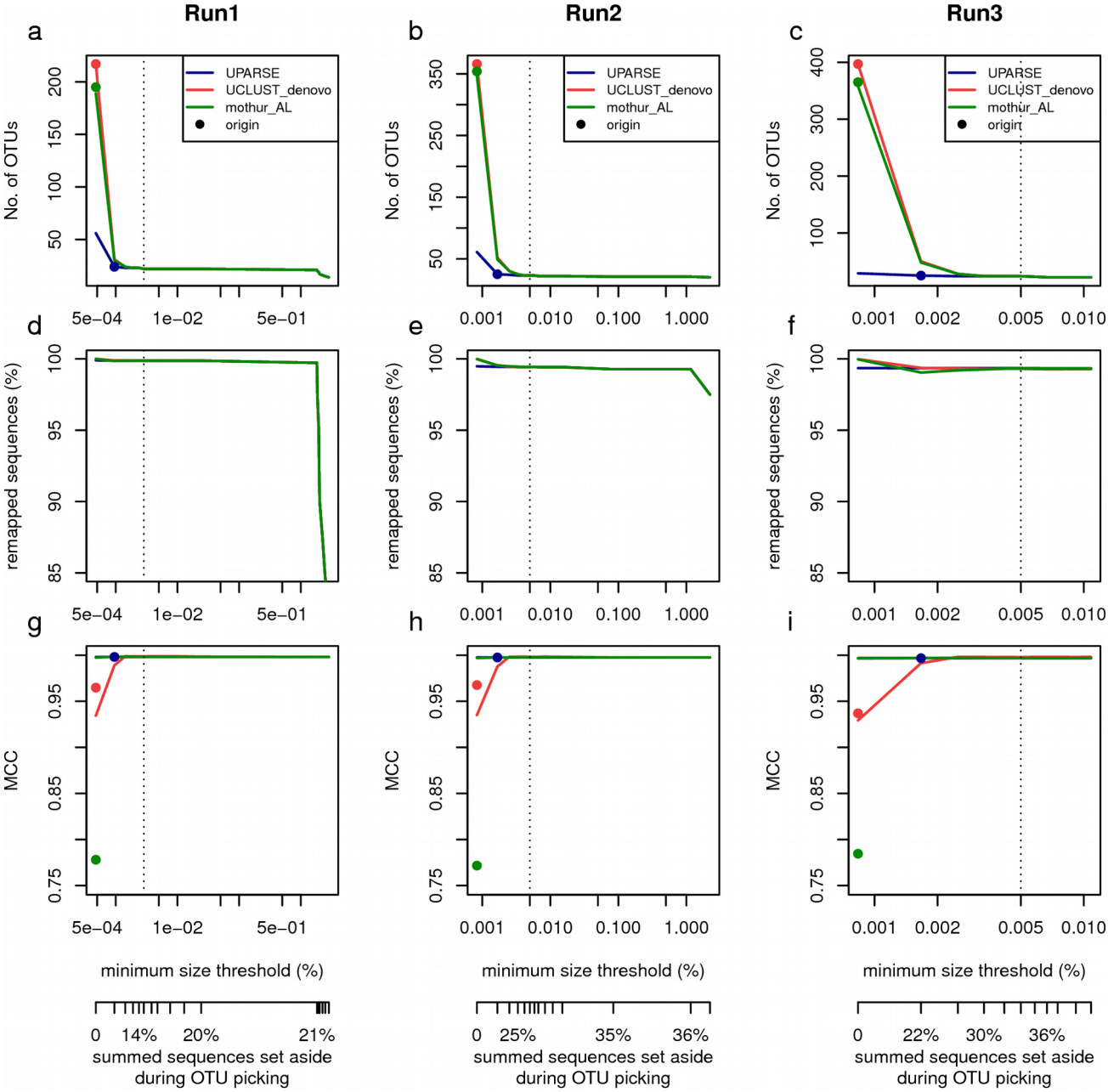

Figure S1 The OTUs obtained by AOR approach in Mock data. (a-c) The number of OTUs decreased to 22 at thresholds; (d-f) the total ratio of sequences remapped back to OTUs also maintained at >99%; (g-i) the MCC values increased to >0.95, indicating ideal OTU delineation quality. The alternative x axis at the bottom indicates how many sequences did not attending initial OTU delineation at each threshold levels. After OTU delineation, qualified unique sequences were remapped to OTUs with 97% similarity threshold. Dots indicate the original results of corresponding OTU delineation methods.

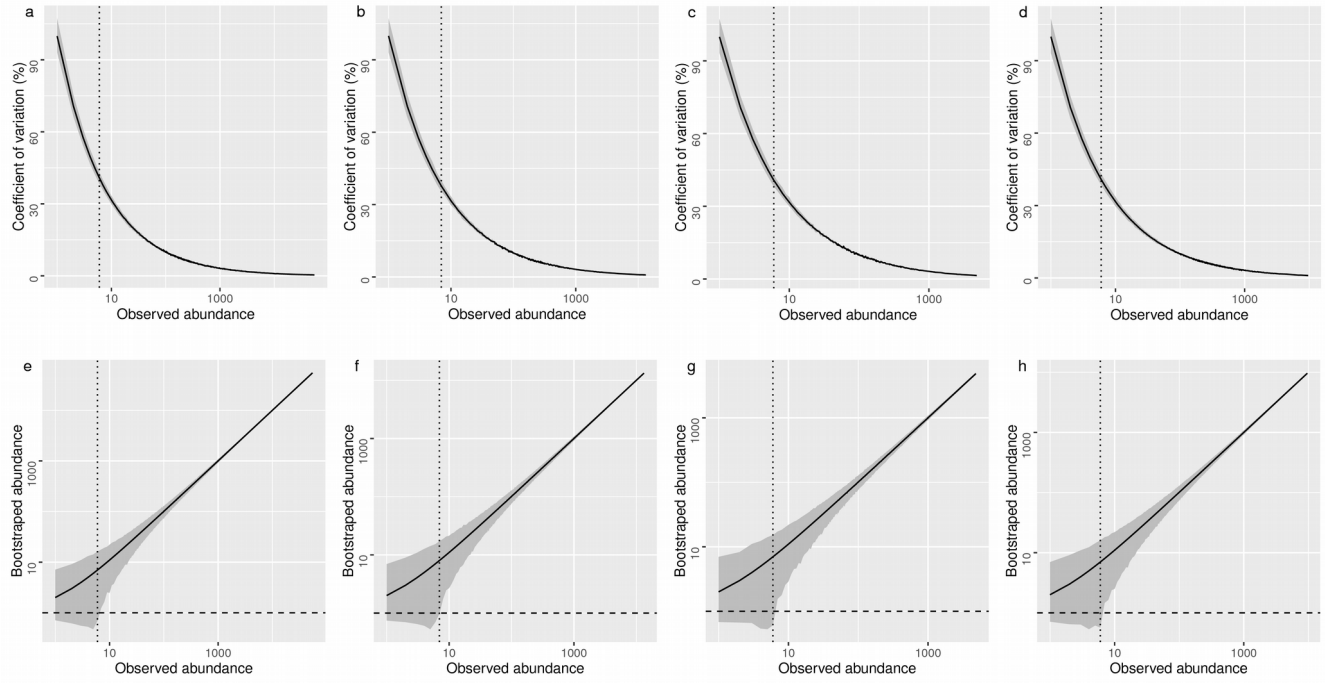

Figure S2 Coefficient of variation(a-d) and the 99% confidential intervals of bootstrapped abundance (e-h) in (a, e) PWS, (b, f) Ultra, (c, g) River and (d, h) Water data. The Coefficient of variation decreased quickly along with the sequences' abundances. The distribution of bootstrapped abundance included zero when the abundances were really low. Dashed vertical lines showed the abundance thresholds for OTU delineation.

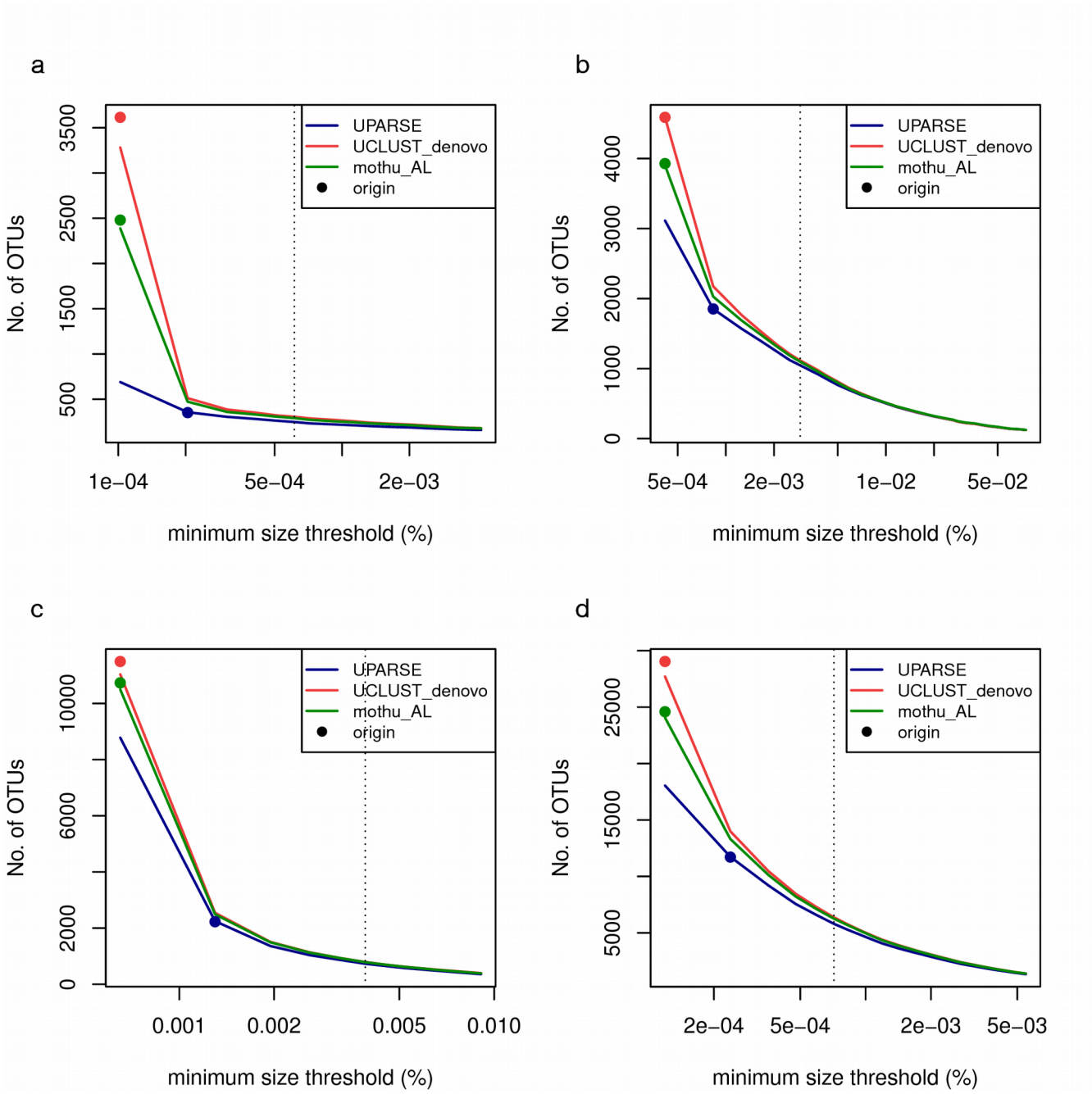

Figure S3 The OTUs obtained by AOR approach in (a) PWS, (b) Ultra, (c) River and (d) Water data sets. The vertical dashed lines indicates the threshold set by bootstrap resampling. Different pipelines obtained close number of OTUs at these thresholds. Dots indicate the original results of corresponding OTU delineation methods.

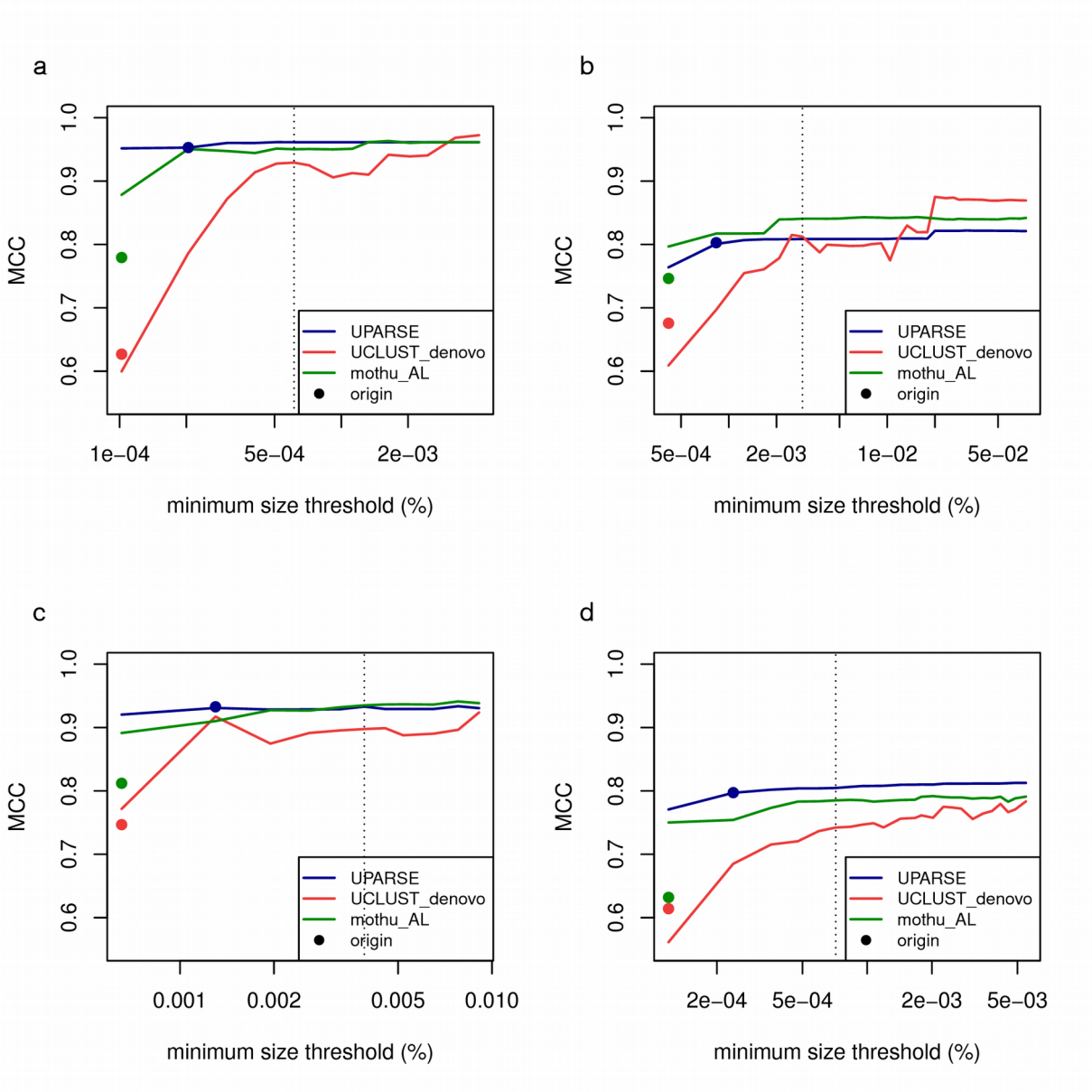

Figure S4 The MCC value in (a) PWS, (b) Ultra, (c) River and (d) Water data sets increased along with the threshold. After OTU delineation, all "qualified sequences" were remapped to OTUs with 97% similarity. Dots indicate the original results of corresponding OTU delineation methods.

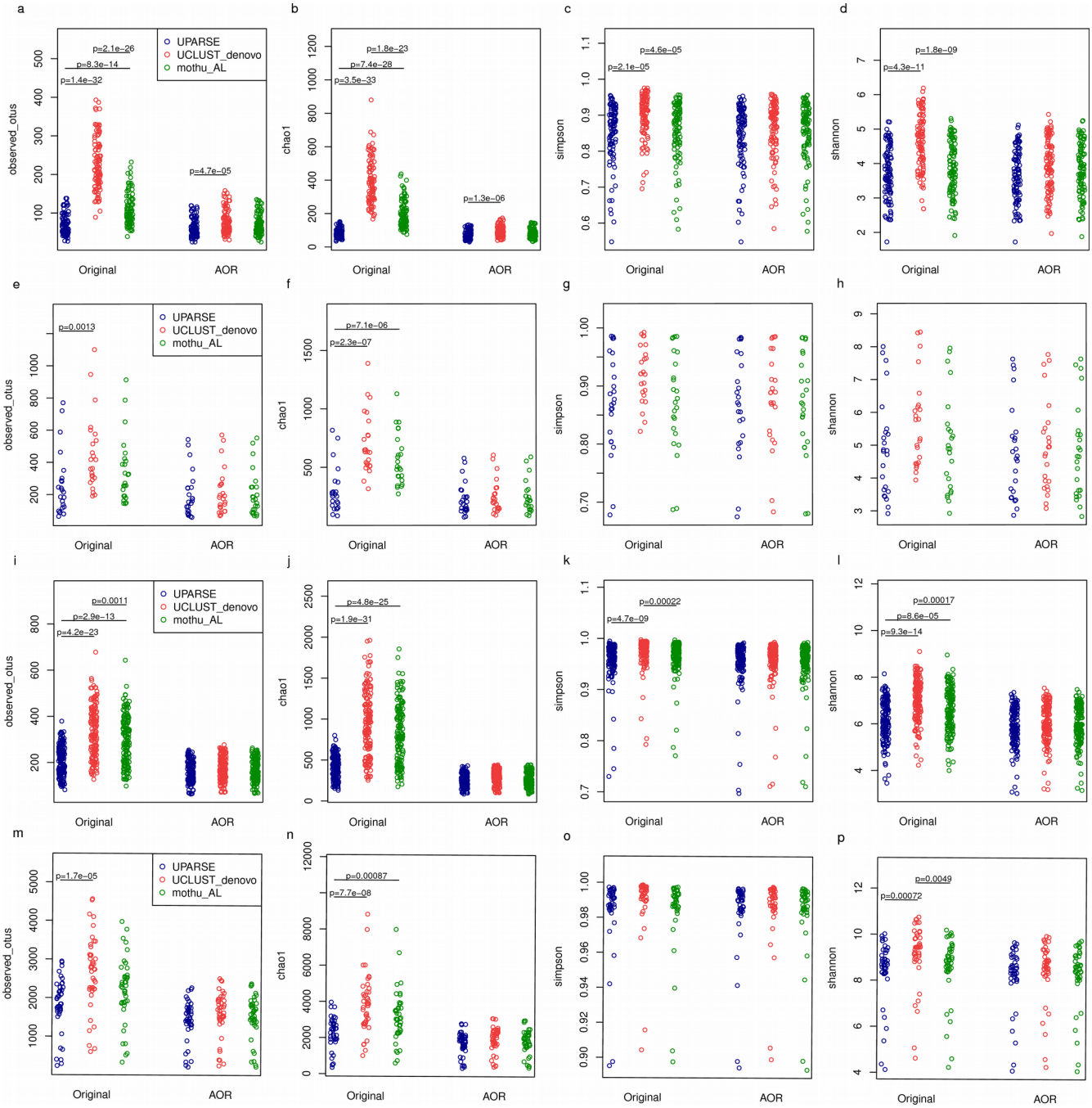

Figure S5 AOR resulted in less OTUs but comparable alpha diversity in PWS (a-d), Ultra (e-h), River (i-l) and Water (m-p) data. (a, e, i, m) Number of OTUs, (b, f, j, n) Chao1 indices, (c, g, k, o) Simpson indices and (d, h, l, p) Shannon indices per sample were calculated. Multiple comparison was performed using Wilcox test, p values were adjusted by FDR method.

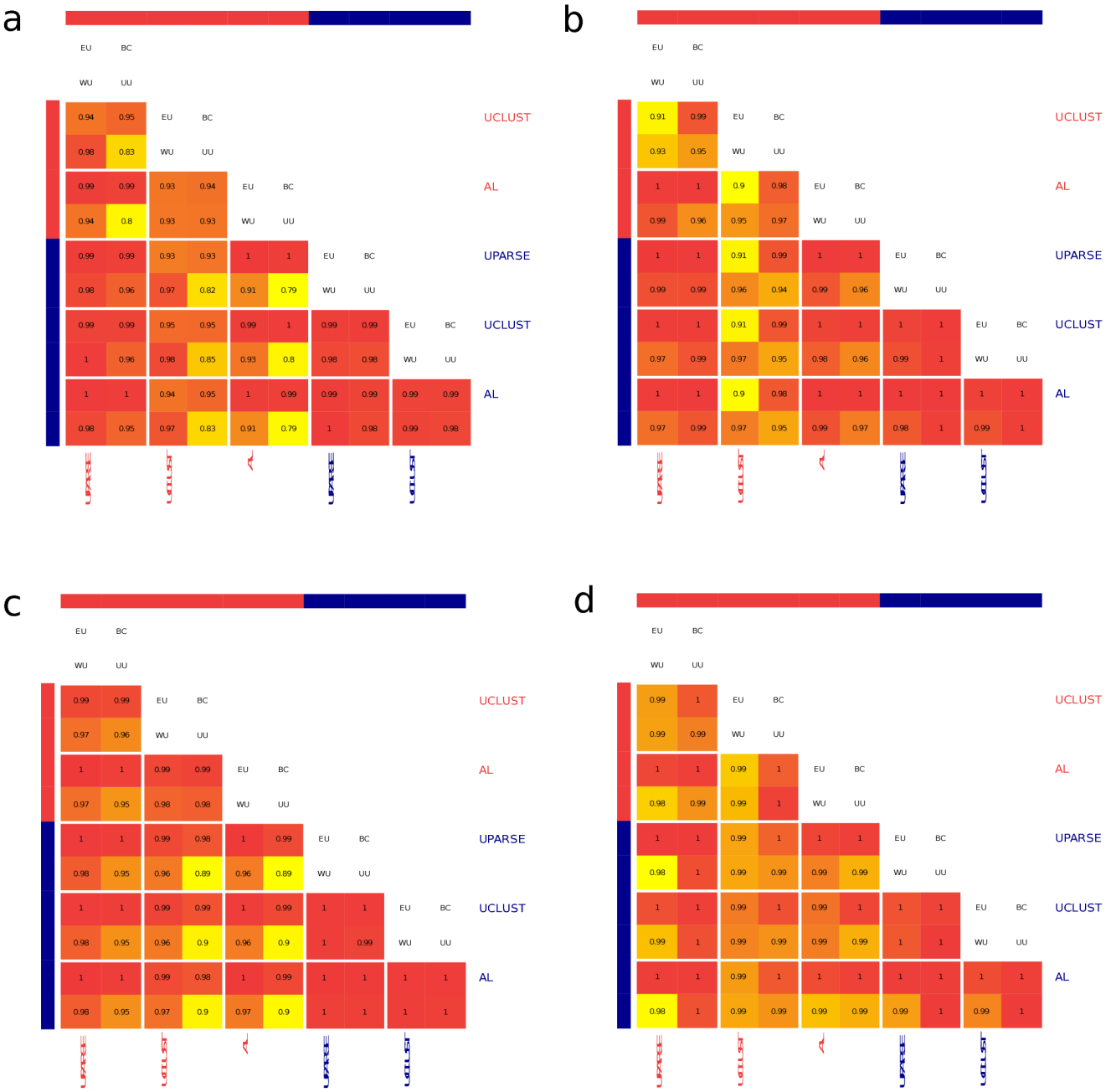

Figure S6 AOR resulted in more consistent beta diversity among methods in (a) PWS, (b) Ultra, (c) River and (d) Water data. Mantel r Statistics were obtained by comparing beta diversity distance matrices between each pair of analysis methods with (Red) original results, (Blue) AOR approach incorporated.

Table S3 The average error rates of the raw sequences reported by sequencing machine, QC sequences passing different quality control methods, the final qualified sequences for OTU delineation, and the qualified sequences pre-clustered with up to 1 difference per 100 bp.

	reported by machine	UPARSE	mothur	moira	S+BH+P	qualified	pre.cluster
run1	2.51%	0.19%	0.22%	0.20%	0.18%	0.07%	0.03%
run2	2.09%	0.47%	0.51%	0.49%	0.48%	0.13%	0.04%
run3	3.91%	0.46%	0.52%	0.50%	0.56%	0.17%	0.06%

Table S4 The number of sequences passed quality filtration using different methods.

	UPARSE	mothur	moira	S+BH+P	Raw sequences
Run1	254086	279699	278119	262288	315365
Run2	130940	147535	158214	148431	194967
Run3	190818	250932	275005	138179	408720

Table S5 The abundance threshold of unreliable sequences.

Data sets	Mock run1	Mock run2	Mock run3	Simulate	PWS	Ultra	River	Water
Abundance threshold of unreliable unique sequences	6	6	6	6	6	7	6	6
Relative abundance threshold of unreliable unique sequences	0.003%	0.005%	0.005%	0.0006%	0.0006%	0.003%	0.004%	0.0007%

resample_uniques_ci.r

```
#The following script is validated on R platform version 3.3.1. The package
“doParallel” should be installed a prior.

require(doParallel)

#Calculate whether the 99% confidential interval of unique sequence contains 0. If
so, the unique sequence is not statistically reliable.

n <- 1000 #Set the No. of replicates in bootstrap process.

data <- read.table("uniques.size") #uniques.size is a file contains only one
column, in which are the absolute abundances of each unique sequences.

data<-sort(data[,1],decreasing = T)

cl <- makeCluster(12) #Set the No. of proccessors used during parallel calculation.

registerDoParallel(cl)

set.seed(1234) #The seed of randomized resampling.

#length(data) is the number of unique sequences, sum(data) is the total sequences

resample <- replicate(n,sample(1:length(data),size = sum(data),replace = T,prob =
data))

#transform resample to factor() using table() to count the length(data) unique
sequences in sum(data) total sequences

count <- foreach(i=1:n,.combine = 'cbind',.multicombine = T) %dopar%
as.data.frame(table(factor(resample[,i],levels=1:length(data))))[,2]

rm(resample)

r.b <- rowMeans(count) #Mean of bootstrapped abundance

r.adj <- 2*data-r.b #Adjusted mean of bootstrapped abundance

r.quan <- parApply(cl,count,1,quantile,probs=c(0.005,0.5,0.995),names=F) #99%
percentile of bootstrapped abundance

r.quan <- t(r.quan)

ci.min <- r.adj-(r.b-r.quan[,1]) #99% confidantial interval of abundance

ci.max <- r.adj+(r.quan[,3]-r.b)

rmse <- sqrt(rowSums((data-count)^2)/n)

pval <- pt(data/rmse,df=n,lower.tail = F)

#The result of each unique sequence

write.table(cbind(data,data/sum(data)*100,r.quan,ci.min,ci.max,pval,data/
rmse),"resample.99percentile.txt",quote=F,row.names = F,col.names =
c("abund","relative_abund","lower_quan","median","higher_quan","lower_ci","higer_ci
```

```
","pval","signal/noise"),sep = "\t")

#The bootstrapped abundance

write.table(count,"resample.count.txt",quote=F,row.names = F,col.names = F,sep = "\
t")

#Determine the abundance range of reliable and unreliable sequences.

write.table(cbind(range(data[ci.min<=0]),range(data[ci.min<=0]/
sum(data)*100),range(data[ci.min>0]),range(data[ci.min>0]/
sum(data)*100)),file="resample.99percentile.range.txt",quote=F,sep="\
t",row.names=F,col.names=c("abund_ci<=0","relative_abund_ci<=0","abund_ci>0","relat
ive_abund_ci>0"))

stopCluster(cl)
```