Clustering_Rareification_Heatmap

Trevor Eakes

May 8, 2016

Rarefaction Methods for the Kuroshio

Rarefaction is a method for comparing diversity between sample groups which may have different coverage, area, or even distributions. It estimates the relationship between how diversity such as richness or the 1st order Hill number accumulates as more individuals/samples are collected. Recent development of this package and the idea behind rarefaction now allow for using abundance in adition to frequency. Previously I could only use frequency with this idea which is statistically less powerful.

Research on rarefaction conducted by: Dr. Anne Chao National Tsing Hua University Ecological Statistics, biodiversity

First to prepare the data

Note: You must first run the dataframe cunstruction Rmd chunks. The file is titled DiversityDataframeConstruction.Rmd

You will also need the following packages downloaded: gplots ggplots2 iNEXT

Dataframe creation and visualization

Note, other graphs may be created from the data easily by changing the inputs in this script. Only relavent graphs have been chosen for printing.

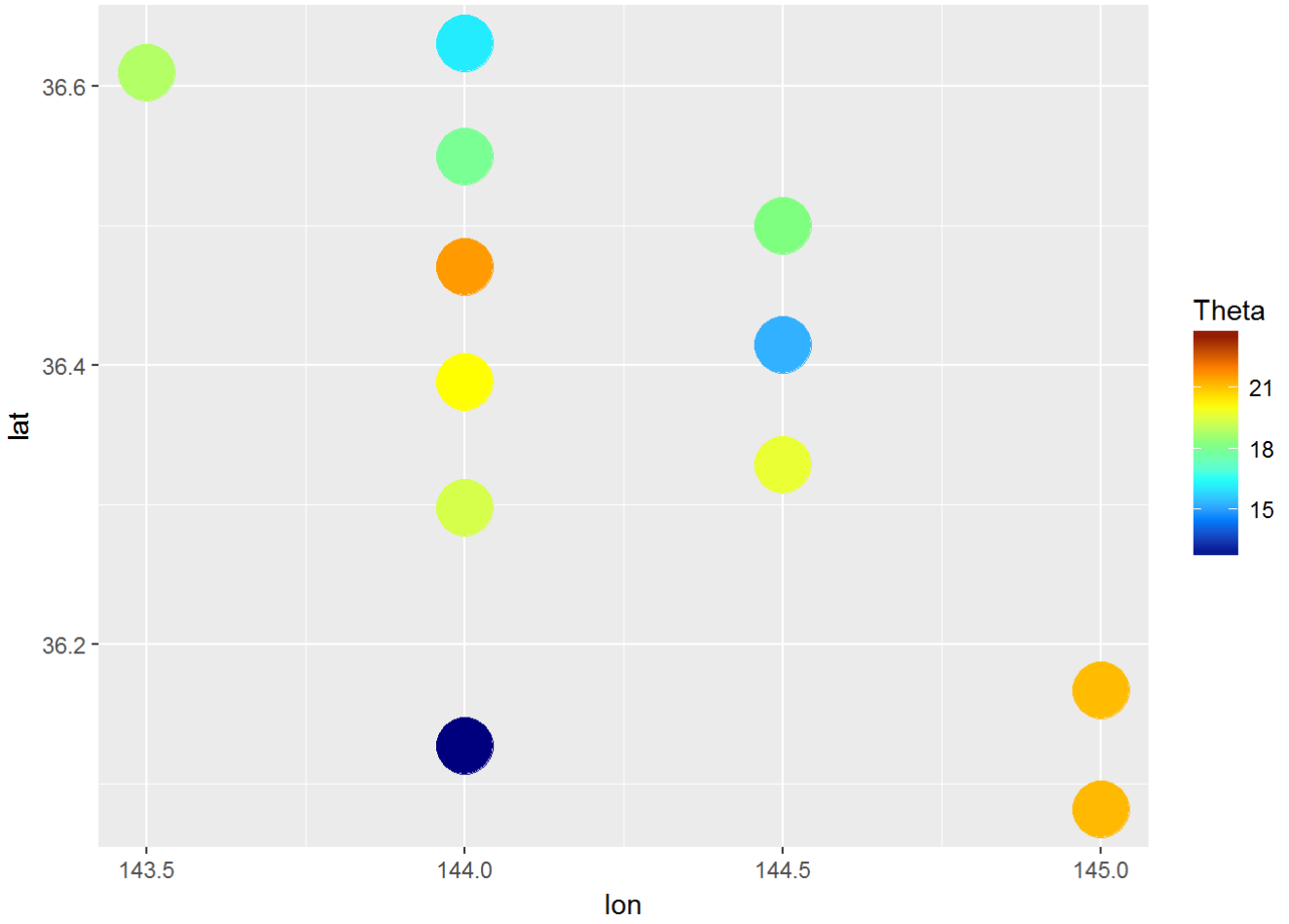
```
centitest <- read.csv('centitest.csv') #Pull up processed phytoplankton cell
#counts csv from directory
Adiv.abiotic <- read.csv('Adiv.abiotic.csv')# Pull up all
#other related environmental and diversity data
centitest.Clade <- cbind(clades=Adiv.abiotic$cluster, centitest)
#adding the clade t centitest

#Clustering the Fillament manually by salinity and position
Adiv.abioticR <- cbind("rows"=c(1:nrow(Adiv.abiotic)), Adiv.abiotic)
# add rows to keep track of things
Fillament <- Adiv.abioticR[which(Adiv.abioticR[, "DfromF"]>0),]
#Take stations north of the front


#More manual editing, WILL NEED TO ADJUST THIS AS EASY,
#MAYBE SET UP OBJECTIVE FILTERING ALGORYTHM
Fillament <- Fillament[which(Fillament[,"S"]>33.85),] #Take stations
#Above 33.9 salinity
Fillament <- Fillament[which(Fillament["lat"]>35.99),]
Fillament <- Fillament[which(Fillament["station"]!=2),]
# must manually input single station number you want removed each time
Fillament <- Fillament[which(Fillament["station"]!=3),]
Fillament <- Fillament[which(Fillament["station"]!=2),]
Fillament <- Fillament[which(Fillament["depth..m."]<80),]

#vISUALIZE FILLAMENT STATIONS
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
jet.colors <- colorRampPalette(c("#00007F", "#007FFF", "cyan", "#7FFF7F", "yellow",
"#FF7F00", "#7F0000"))

ggplot(Fillament, aes(x=lon, y=lat, colour=Theta))+
geom_point(size=10, position=position_dodge(), stat="identity")+
scale_color_gradientn(colours=jet.colors(7), space="rgb", guide="colourbar")
```

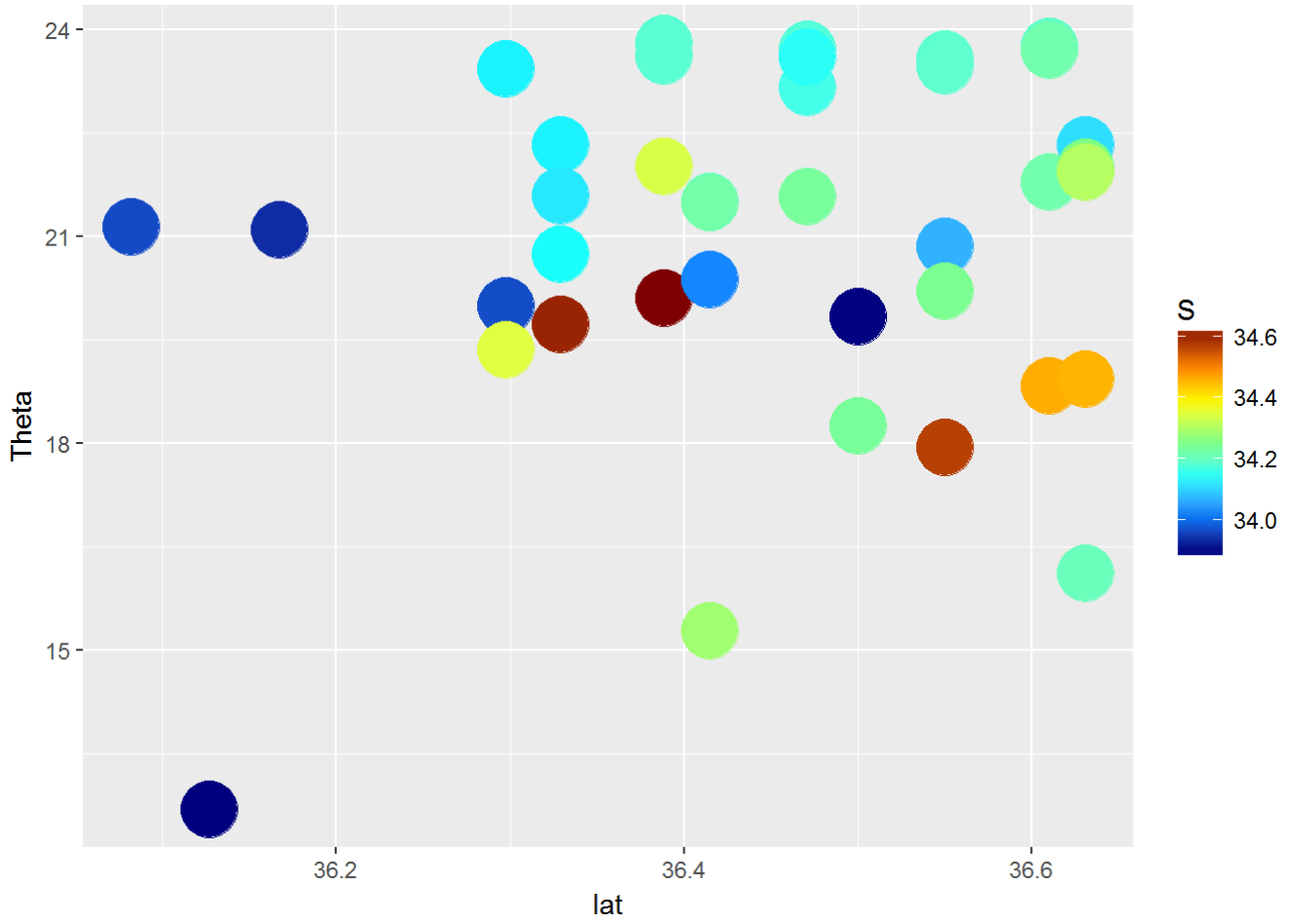
```
## Warning: Non Lab interpolation is deprecated
```

```
## Warning: Width not defined. Set with `position_dodge(width = ?)`
```



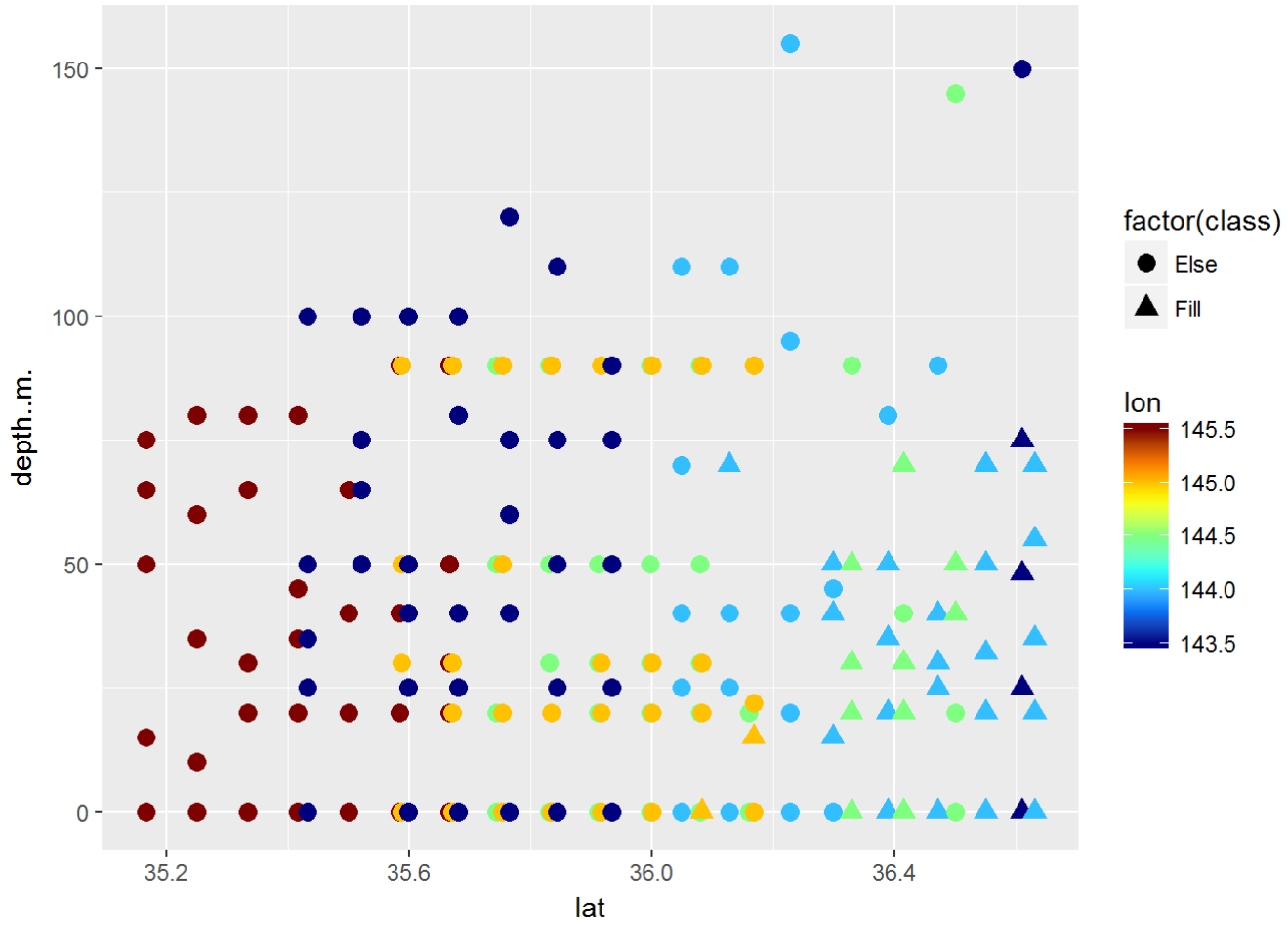

```
#
ggplot(Fillament, aes(x=lat, y=Theta, colour=S)) + geom_point(size=10,              position=pos
ition_dodge(), stat="identity")+
scale_color_gradientn(colours=jet.colors(7), space="rgb", guide="colourbar")
```

```
## Warning: Non Lab interpolation is deprecated

## Warning: Width not defined. Set with `position_dodge(width = ?)`
```




```
#VISUALIZE WHERE FILLAMENT IS RELATIVE TO ALL OTHER SAMPLES
Fill <- data.frame(rows=Fillament$rows, class=rep("Fill", length(Fillament$rows)))
Else <- Adiv.abioticR[-c(Fillament$rows), ]
Else <- data.frame(rows=Else$rows,  class=rep("Else", length(Else$rows)))
class <- rbind(Else, Fill)
class <- class[order(class$rows),]
Adiv.abioticR <- cbind(Adiv.abiotic, class)
#
ggplot(Adiv.abioticR, aes(x=lat, y=depth..m., colour=lon, shape=factor(class)))+
geom_point(size=3,position=position_dodge(), stat="identity")+
scale_color_gradientn(colours=jet.colors(7), space="rgb", guide="colourbar")
```

```
## Warning: Non Lab interpolation is deprecated

## Warning: Width not defined. Set with `position_dodge(width = ?)`
```



Setting up the fillament data and preparing for rarefaction

```
  SplitData <- function(data, column, header){
  #data=dataframe, column=the the number of
  #the column desired to be used as a numerical factor,
#header is the title of the data frames without the numbers at the end.
#MUST BE IN PARATHESES
libr <- setNames(split(data, data[, column]), paste0(header, unique(data[,column])))
#Seperates the dataframe into multiple dataframes based on the factor
  return(libr) #Exports both s and each individual, new data frame to the global environment
  }
fillament <- centitest.Clade[Fillament$rows,]
#Grab data for only fillament stations
centitest.Clade <- centitest.Clade[-Fillament$rows,]
#remove fillament stations from centitest.Clade
SplitD <- SplitData(centitest.Clade, 1, 'dfcluster')
#Split up the data into the remaining clasified water masses
dfcluster1 <- SplitD$dfcluster1
dfcluster2 <-SplitD$dfcluster2
dfcluster3 <- SplitD$dfcluster3
#sums species in each cluster
Clus1 <- as.numeric(apply(dfcluster1[,6:ncol(dfcluster1)], 2, function(x) sum(x>0)))
#gets rid of the 0's, add the number of rows to the front to match the package format
clus1 <- c(nrow(dfcluster1), Clus1[Clus1>0])#And so on
Clus2 <- as.numeric(apply(dfcluster2[,6:ncol(dfcluster2)], 2, function(x) sum(x>0)))
clus2 <- c(nrow(dfcluster2), Clus2[Clus2>0])
Clus3 <- as.numeric(apply(dfcluster3[,6:ncol(dfcluster3)], 2, function(x) sum(x>0)))
clus3 <- c(nrow(dfcluster3), Clus3[Clus3>0])
Clus4 <- as.numeric(apply(fillament[,6:ncol(fillament)], 2, function(x) sum(x>0)))
clus4 <- c(nrow(fillament), Clus4[Clus4>0])

#For abundance, ABUNDANCE MOST INFORMATIVE
clusA1 <- as.numeric(apply(dfcluster1[,6:ncol(dfcluster1)], 2, function(x) sum(x)))
#sum species abundances
clusA2 <- as.numeric(apply(dfcluster2[,6:ncol(dfcluster2)], 2, function(x) sum(x)))
clusA3 <- as.numeric(apply(dfcluster3[,6:ncol(dfcluster3)], 2, function(x) sum(x)))
clusA4 <- as.numeric(apply(fillament[,6:ncol(fillament)], 2, function(x) sum(x)))

#Making Rarefaction lists for the clusters with row names
Rarefaction <- list("Deep"=clus1, "Fillament"=clus4, "Kuroshio"=clus2, "Oyashio"=clus3)
RarefactionAbund <- list("Deep"=clusA1, "Fillament"=clusA4, "Kuroshio"=clusA2, "Oyashio"=clusA3)
```

List of stations and their depths

```
#Summarizing Function, returns mean, standard deviation, range, and # of observations
summarize <- function(data) {
Means <- apply(data, 2, function(x) mean(x))
SD <- apply(data, 2, function(x) sd(x))
Min<-apply(data, 2, function(x) min(x))
Max<-apply(data, 2, function(x) max(x))
Obs<- apply(data, 2, function(x) length(x))
summry <- cbind(Means, SD, Min, Max, Obs)
rownames(summry) <- colnames(data)
return(summry)
}
#
WaterM <- cbind(clades=rep(4, length(Fillament$rows)), X=Fillament$rows)
#add code for fillament
WaterM <- rbind(WaterM, centitest.Clade[, 1:2]) #combine with other clade data
WaterM <- WaterM[order(WaterM[,2]),] #Reorder by row number
Adiv.abioticC <- cbind(WaterM, Adiv.abiotic) #add back to the dataframe

#Summary
CladeAdiv <- SplitData(Adiv.abioticC, "clades", "Adiv")
DeepAdiv <- CladeAdiv$Adiv1
FillAdiv <- CladeAdiv$Adiv2
OyaAdiv <- CladeAdiv$Adiv3
KuroAdiv <- CladeAdiv$Adiv4
(DeepSmry <- summarize(DeepAdiv[,c("Chlorophyll", "Theta", "S", "depth..m.", "Richness",
"Cellcount", "ShannonWiener", "sigPoDen")]))
```

```
##                     Means         SD      Min        Max Obs
## Chlorophyll            NA         NA       NA         NA  48
## Theta          18.3559125  2.4355552 11.25500  25.024200  48
## S              33.5706229  0.2444359 32.85600  34.060000  48
## depth..m.      24.0000000 22.8715190  0.00000  90.000000  48
## Richness        3.6875000  1.4165624  1.00000   7.000000  48
## Cellcount     103.8958333 48.4896302 10.00000 194.000000  48
## ShannonWiener   0.3857303  0.2631098  0.00000   1.200087  48
## sigPoDen       24.0658798  0.6159876 22.15003  25.855902  48
```

```
(FillSmry <- summarize(FillAdiv[,c("Chlorophyll", "Theta", "S", "depth..m.", "Richness",
"Cellcount", "ShannonWiener", "sigPoDen")]))
```

```
##                    Means         SD        Min       Max Obs
## Chlorophyll    0.3909465  0.3266222  0.1078705   1.15900  21
## Theta         13.8657619  2.8997825 10.1485000  21.32200  21
## S             33.9886429  0.2017718 33.7990000  34.52050  21
## depth..m.     86.4285714 40.7474802  0.0000000 155.00000  21
## Richness       3.0952381  1.7001401  1.0000000   7.00000  21
## Cellcount     94.0000000 48.3993802  8.0000000 171.00000  21
## ShannonWiener  0.3319775  0.3331621  0.0000000   1.29427  21
## sigPoDen      25.3973834  0.7040247 23.4938311  26.15067  21
```

```
(OyaSmry <- summarize(OyaAdiv[,c("Chlorophyll", "Theta", "S", "depth..m.", "Richness",
"Cellcount", "ShannonWiener", "sigPoDen")]))
```

```
##                    Means         SD         Min        Max Obs
## Chlorophyll    0.5156794  0.3399605  0.09728218   1.770231  38
## Theta         20.7773026  2.6202457 12.69550000  23.804000  38
## S             34.1861842  0.2225731 33.47550000  34.634000  38
## depth..m.     32.5000000 22.4604758  0.00000000  75.000000  38
## Richness       4.4736842  2.0890557  1.00000000   9.000000  38
## Cellcount     89.7368421 40.3362115  4.00000000 215.000000  38
## ShannonWiener  0.6066035  0.4368765  0.00000000   1.419742  38
## sigPoDen      23.9075230  0.6779800 23.08081716  25.594077  38
```

```
(KuroSmry <- summarize(KuroAdiv[,c("Chlorophyll", "Theta", "S", "depth..m.", "Richness",
"Cellcount", "ShannonWiener", "sigPoDen")]))
```

```
##                    Means         SD     Min        Max Obs
## Chlorophyll    0.4976036  0.4262860  0.0380   2.634941  75
## Theta         23.0429480  2.4017330 14.9920  25.483000  75
## S             34.3470893  0.1823246 34.0880  34.746500  75
## depth..m.     44.2000000 33.6118544  0.0000 120.000000  75
## Richness       3.7600000  1.8368598  1.0000  10.000000  75
## Cellcount    115.2533333 72.0217860 14.0000 250.000000  75
## ShannonWiener  0.3930859  0.3470796  0.0000   1.396864  75
## sigPoDen      23.4047892  0.7416939 22.5756  25.605548  75
```

```
#
#Number of Species which appear once or twice for each cluster
Kurolets <- length(which(Clus1<3 & Clus1>0)) #Kuroshio samples
Oyalets <- length(which(Clus2<3 & Clus2>0))  #Oyashio samples
Deeplets <- length(which(Clus3<3 & Clus3>0)) #Deep samples
Fillamentlets <- length(which(Clus4<3 & Clus4>0)) #Fillament
#
#Fillament Station info
(Fillamentinfo <- Adiv.abiotic[Fillament$rows, c("depth..m.", "station", "lat", "lon") ])
```

```
##     depth..m. station     lat   lon
## 1           0       1 36.6102 143.5
## 2          25       1 36.6102 143.5
## 3          48       1 36.6102 143.5
## 4          75       1 36.6102 143.5
## 48         70      10 36.1270 144.0
## 56         15      12 36.2977 144.0
## 57         40      12 36.2977 144.0
## 59         50      12 36.2977 144.0
## 60          0      13 36.3885 144.0
## 61         20      13 36.3885 144.0
## 62         35      13 36.3885 144.0
## 63         50      13 36.3885 144.0
## 65          0      14 36.4710 144.0
## 66         25      14 36.4710 144.0
## 67         30      14 36.4710 144.0
## 68         40      14 36.4710 144.0
## 70          0      15 36.5502 144.0
## 71         20      15 36.5502 144.0
## 72         32      15 36.5502 144.0
## 73         50      15 36.5502 144.0
## 74         70      15 36.5502 144.0
## 75          0      16 36.6308 144.0
## 76         20      16 36.6308 144.0
## 77         35      16 36.6308 144.0
## 78         55      16 36.6308 144.0
## 79         70      16 36.6308 144.0
## 82         40      17 36.5000 144.5
## 83         50      17 36.5000 144.5
## 85          0      18 36.4148 144.5
## 86         20      18 36.4148 144.5
## 87         30      18 36.4148 144.5
## 89         70      18 36.4148 144.5
## 90          0      19 36.3287 144.5
## 91         20      19 36.3287 144.5
## 92         30      19 36.3287 144.5
## 93         50      19 36.3287 144.5
## 142         0      32 36.0818 145.0
## 147        15      33 36.1672 145.0
```

```
#Other water masses info
(KuroshioInfo <-Adiv.abiotic[dfcluster1$X,c("depth..m.", "station", "lat", "lon") ])
```

```
##     depth..m. station     lat   lon
## 5         150       1 36.6102 143.5
## 10         90       2 35.9347 143.5
## 15        110       3 35.8443 143.5
## 43         70       9 36.0485 144.0
## 44        110       9 36.0485 144.0
## 49        110      10 36.1270 144.0
## 53         95      11 36.2273 144.0
## 54        155      11 36.2273 144.0
## 84        145      17 36.5000 144.5
## 103        20      22 35.9967 144.5
## 104        30      22 35.9967 144.5
## 108        20      23 35.9117 144.5
## 109        30      23 35.9117 144.5
## 113        50      24 35.8310 144.5
## 117        50      25 35.7430 144.5
## 121        50      26 35.5870 145.0
## 136        30      30 35.9160 145.0
## 140        30      31 35.9997 145.0
## 144        30      32 36.0818 145.0
## 148        22      33 36.1672 145.0
## 152        30      34 35.6667 145.5
```

```
(OyashioInfo <- Adiv.abiotic[dfcluster2$X,c("depth..m.", "station", "lat", "lon") ])
```

```
##      depth..m. station     lat   lon
## 6            0       2 35.9347 143.5
## 12          25       3 35.8443 143.5
## 13          50       3 35.8443 143.5
## 14          75       3 35.8443 143.5
## 16           0       4 35.7645 143.5
## 17          40       4 35.7645 143.5
## 18          60       4 35.7645 143.5
## 19          75       4 35.7645 143.5
## 20         120       4 35.7645 143.5
## 21           0       5 35.6807 143.5
## 22          25       5 35.6807 143.5
## 23          40       5 35.6807 143.5
## 24          80       5 35.6807 143.5
## 25         100       5 35.6807 143.5
## 26           0       6 35.5985 143.5
## 27          25       6 35.5985 143.5
## 28          40       6 35.5985 143.5
## 29          50       6 35.5985 143.5
## 30         100       6 35.5985 143.5
## 31          50       7 35.5208 143.5
## 32          65       7 35.5208 143.5
## 33          75       7 35.5208 143.5
## 34         100       7 35.5208 143.5
## 35           0       8 35.4323 143.5
## 36          25       8 35.4323 143.5
## 37          35       8 35.4323 143.5
## 38          50       8 35.4323 143.5
## 39         100       8 35.4323 143.5
## 64          80      13 36.3885 144.0
## 69          90      14 36.4710 144.0
## 94          90      19 36.3287 144.5
## 118         90      25 35.7430 144.5
## 119          0      26 35.5870 145.0
## 122         90      26 35.5870 145.0
## 123          0      27 35.6712 145.0
## 124         20      27 35.6712 145.0
## 125         30      27 35.6712 145.0
## 126         90      27 35.6712 145.0
## 127          0      28 35.7537 145.0
## 128         20      28 35.7537 145.0
## 129         50      28 35.7537 145.0
## 130         90      28 35.7537 145.0
## 133         90      29 35.8347 145.0
## 153         50      34 35.6667 145.5
## 154         90      34 35.6667 145.5
## 155          0      35 35.5833 145.5
## 156         20      35 35.5833 145.5
## 157         40      35 35.5833 145.5
## 158         90      35 35.5833 145.5
## 159          0      36 35.5000 145.5
## 160         20      36 35.5000 145.5
## 161         40      36 35.5000 145.5
```

```
## 162         65      36 35.5000 145.5
## 163          0      37 35.4167 145.5
## 164         20      37 35.4167 145.5
## 165         35      37 35.4167 145.5
## 166         45      37 35.4167 145.5
## 167         80      37 35.4167 145.5
## 168          0      38 35.3333 145.5
## 169         20      38 35.3333 145.5
## 170         30      38 35.3333 145.5
## 172         80      38 35.3333 145.5
## 173          0      39 35.2500 145.5
## 174         10      39 35.2500 145.5
## 175         35      39 35.2500 145.5
## 176         60      39 35.2500 145.5
## 177         80      39 35.2500 145.5
## 178          0      40 35.1667 145.5
## 179         15      40 35.1667 145.5
## 180         50      40 35.1667 145.5
## 181         65      40 35.1667 145.5
## 182         75      40 35.1667 145.5
## NA          NA      NA       NA    NA
## NA.1        NA      NA       NA    NA
## NA.2        NA      NA       NA    NA
```

```
(DeepInfo <-Adiv.abiotic[dfcluster3$X,c("depth..m.", "station", "lat", "lon") ])
```

```
##     depth..m. station     lat   lon
## 7          25       2 35.9347 143.5
## 8          50       2 35.9347 143.5
## 9          75       2 35.9347 143.5
## 11          0       3 35.8443 143.5
## 40          0       9 36.0485 144.0
## 41         25       9 36.0485 144.0
## 42         40       9 36.0485 144.0
## 45          0      10 36.1270 144.0
## 46         25      10 36.1270 144.0
## 47         40      10 36.1270 144.0
## 50          0      11 36.2273 144.0
## 51         20      11 36.2273 144.0
## 52         40      11 36.2273 144.0
## 55          0      12 36.2977 144.0
## 58         45      12 36.2977 144.0
## 80          0      17 36.5000 144.5
## 81         20      17 36.5000 144.5
## 88         40      18 36.4148 144.5
## 95          0      20 36.1602 144.5
## 96         20      20 36.1602 144.5
## 100        50      21 36.0790 144.5
## 101        90      21 36.0790 144.5
## 102         0      22 35.9967 144.5
## 105        50      22 35.9967 144.5
## 106        90      22 35.9967 144.5
## 107         0      23 35.9117 144.5
## 110        50      23 35.9117 144.5
## 111         0      24 35.8310 144.5
## 112        30      24 35.8310 144.5
## 114        90      24 35.8310 144.5
## 115         0      25 35.7430 144.5
## 116        20      25 35.7430 144.5
## 120        30      26 35.5870 145.0
## 131         0      29 35.8347 145.0
## 132        20      29 35.8347 145.0
## 134         0      30 35.9160 145.0
## 135        20      30 35.9160 145.0
## 137        90      30 35.9160 145.0
## 138         0      31 35.9997 145.0
## 139        20      31 35.9997 145.0
## 141        90      31 35.9997 145.0
## 142         0      32 36.0818 145.0
## 143        20      32 36.0818 145.0
## 146         0      33 36.1672 145.0
## 147        15      33 36.1672 145.0
## 149        90      33 36.1672 145.0
## 151        20      34 35.6667 145.5
## 171        65      38 35.3333 145.5
```

Heatmap ploting

Finally, We visualize how the communities appear in each clustered water mass, this can tell us why the water masses are considered different and in what way they difer in species abundance and presence

This is an important figure for understanding the Kuroshio

Values are based off of phytoplankton incidence (# of times observed) which has been divided by the total number of samples in each group. We also use raw abundance data from each cluster to visualize actual cell counts.

```
require(gplots)# for heatmap.2
```

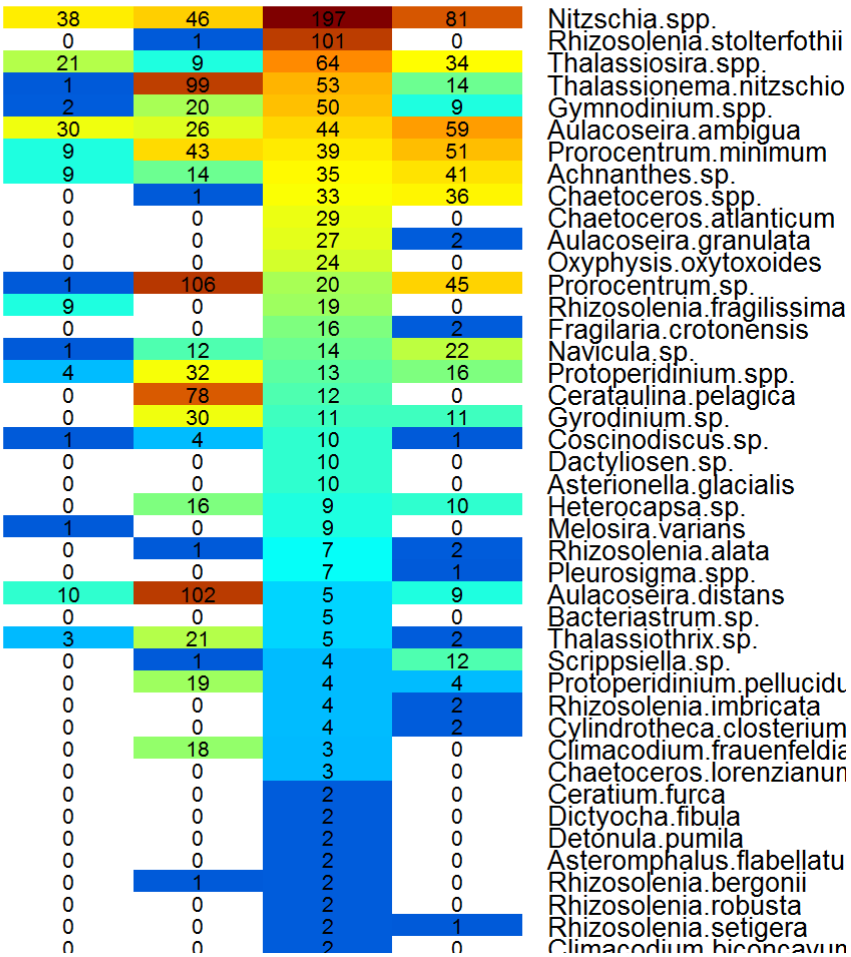
```
## Loading required package: gplots
```

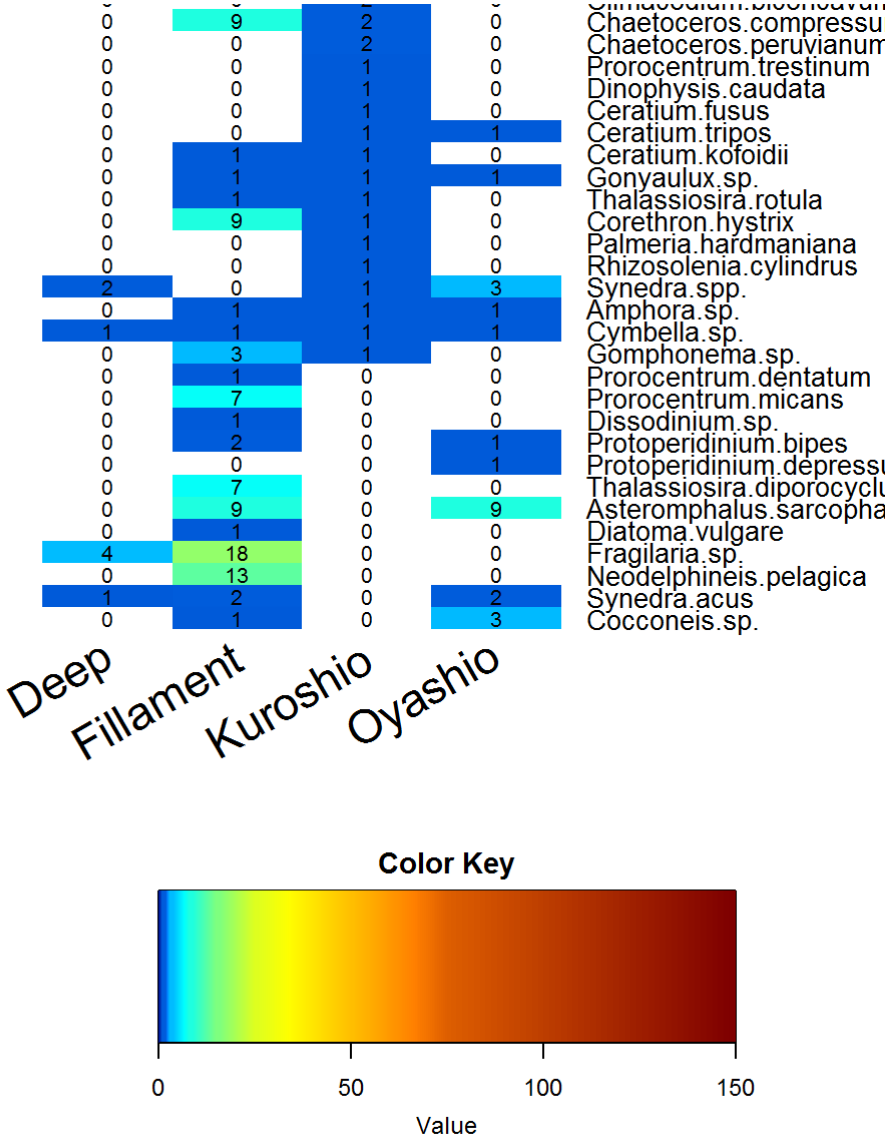
```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

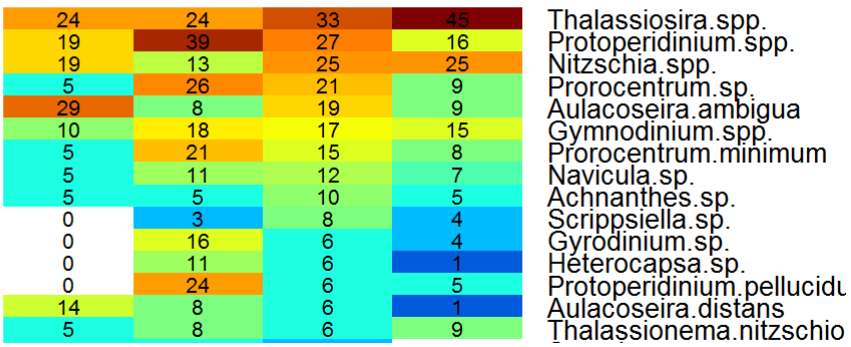
```
library(RColorBrewer)# for better color options

#rounding count data
#For rounding frequency count data and standardizing it to the sample size
#Clus1, Clus2, Clus3, & Clus4 objects from the previous chunk contain the incidence
#information for each cluster.
#dfcluster# is the dataframe from which the centiliter abundance species sums
#for the water mass were calculated and has the number of samples in rows
clus1c <- round(Clus1/nrow(dfcluster1[,-c(1:2)]), 2)*100
clus2c <- round(Clus2/nrow(dfcluster2[,-c(1:2)]), 2)*100
clus3c <- round(Clus3/nrow(dfcluster3[,-c(1:2)]), 2)*100
clus4c <- round(Clus4/nrow(fillament[,-c(1:2)]), 2)*100


Incidence <- as.matrix(data.frame(cbind(Kuroshio=clus1c, Filament=clus4c,
Oyashio=clus3c, Deep=clus2c))) #combining into single data frame
Kuroshio_Phytoplankton <- read.csv("Kuroshio_Phytoplankton.csv")
#grab the original phytoplankton abundances dataframe
Kuroshio_Phytoplankton <- Kuroshio_Phytoplankton[-c(97:99, 186:190), -c(1,2)]
#Editing base data frame, getting rid of uneaded columns and rows

#Raw Abundance Data
HeatAbund <- as.matrix(as.data.frame(RarefactionAbund))
#changing to a dataframe
row.names(HeatAbund)<-colnames(Kuroshio_Phytoplankton)
#changing row names of Incidence to their true Phytoplankton names
row.names(Incidence)<-colnames(Kuroshio_Phytoplankton)
#
HeatAbund<-HeatAbund[order(HeatAbund[,3],decreasing=TRUE),]
#ordering rows from greatest to smallest
Incidence<-Incidence[order(Incidence[,3],decreasing=TRUE),]
#color scheme
col <-jet.colors(398)
col <-c("white", col)
# (optional) defines the color breaks manually for a "skewed" color transition
col_breaks = c(seq(0, 1, length=50),
               seq(2, 3, length=50),
               seq(4,8,length=50),
               seq(9,15,length=50),
               seq(16, 25, length=50),
               seq(26, 50, length=50),
               seq(51, 75, length=50),
               seq(76, 150, length=50))
#Heatmap plotting function, just add matrix!
AbundHeatmap <- function(matrix, col, breaks) {
heatmap.2(matrix,
               cellnote = matrix, # same data set for cell labels
               offsetCol = 0.5,
               notecol="black",      # change font color of cell labels to black
               trace="none",      # turns off trace lines inside the heat map
               margins =c(8,8),     # widens margins around plot
               col=col,       # use on color palette defined earlier
               breaks= breaks,    # enable color transition at specified limits
```

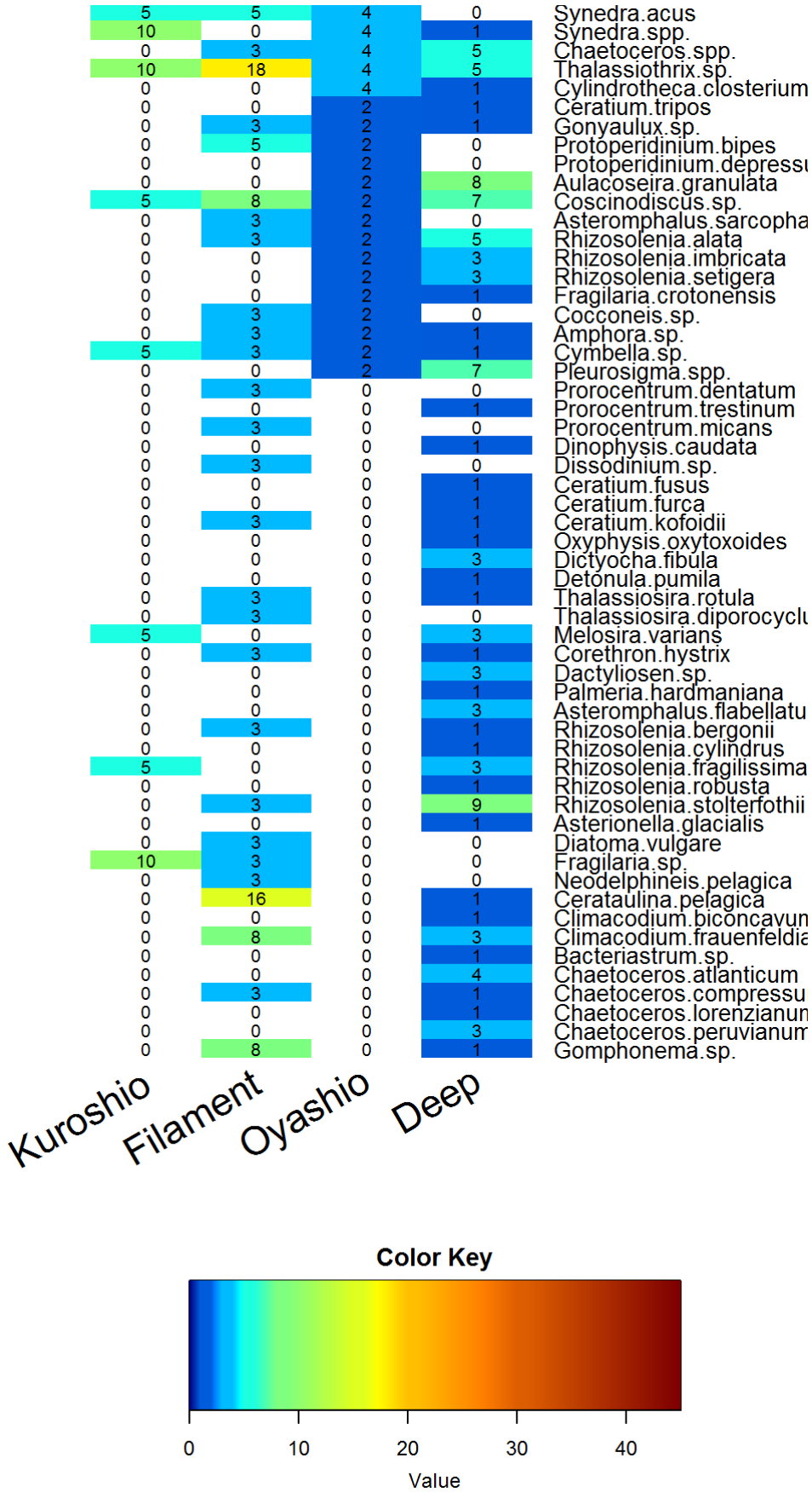
```
              dendrogram="none", # only draw a row dendrogram
              Colv=NA, # turn off column clustering
              cexCol=2.2,
              cexRow=1.3,
              srtCol=30,
              key=TRUE,
              key.title = NULL,
              symbreaks=TRUE,
              symkey=FALSE,
              density.info="none",
              densadj = 0.25,
              Rowv = FALSE,
              #denscol="black",
              keysize=.5,
              #( "bottom.margin", "left.margin", "top.margin", "left.margin" )
              #key.par=list(mar=c(7,.5,3,1)),
               #lmat -- added 2 lattice sections (5 and 6) for padding
              lmat=rbind( c(0, 3, 0), c(2,1,0), c(0,4,0) ),
          lhei=c(.5, 5, 1),
          lwid = c(.5,3,.5))}
par(mar = rep(1, 4))
AbundHeatmap(HeatAbund, col=col, breaks=col_breaks) #Raw Abundance data
```






```
#new breaks for Incidence
Inc_col_breaks = c(seq(0, 1, length=50), seq(2, 3, length=50),seq(4,5,length=50),
seq(6,8,length=50),seq(9, 15,
length=50),seq(16, 20, length=50), seq(21, 30, length=50), seq(31, 45, length=50))
AbundHeatmap(Incidence, col=col, breaks=Inc_col_breaks)
```





Rarefaction and graphing Rarefaction curves

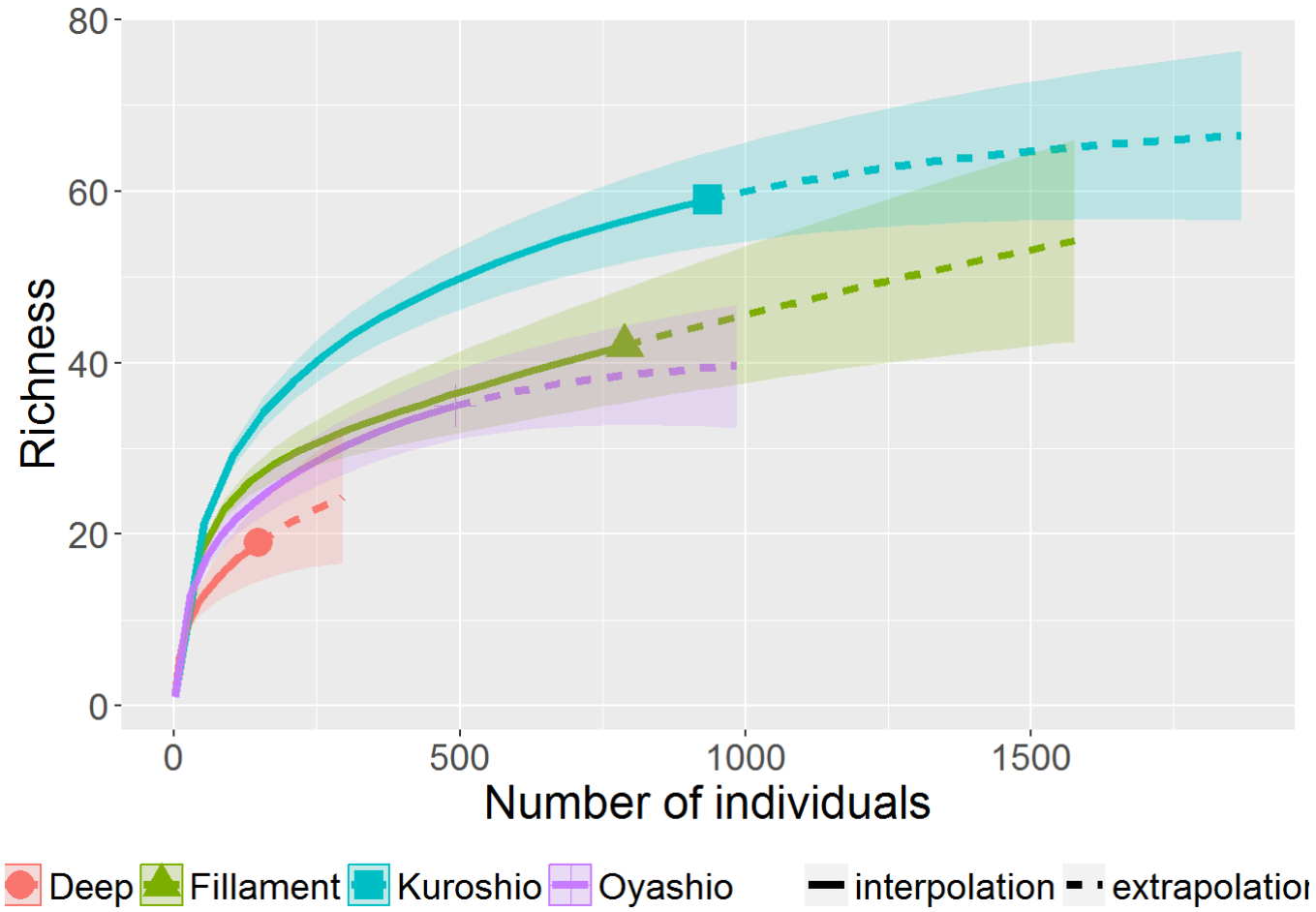
Note: Incidence rarefaction is turned on for improved speed running Rmd. Due to recent package updates incidence input data needs to be adjusted to match apropriate package format.

Aditional plotting may be done here using the INEXT package. This just shows the most relavent plots of interest

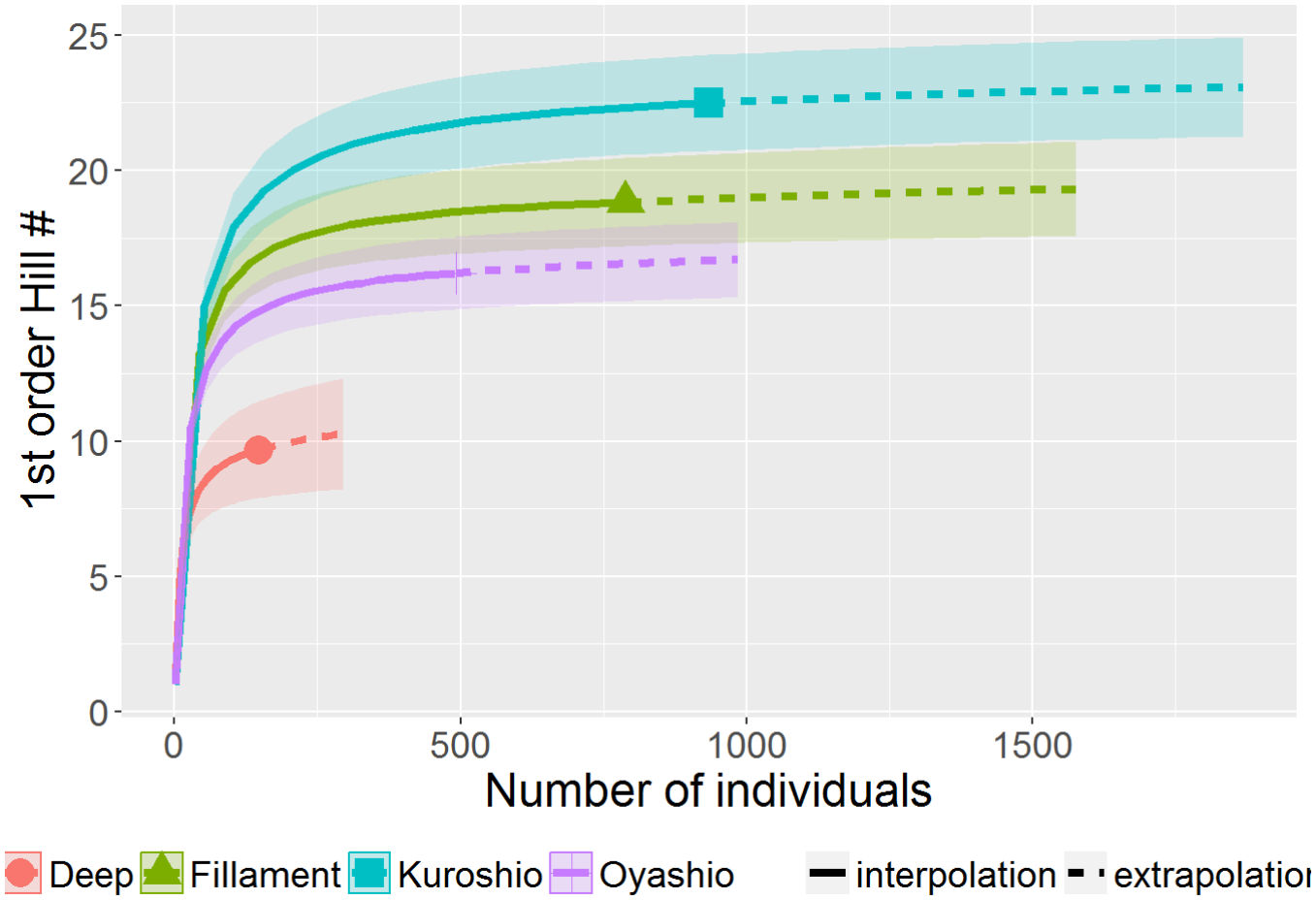
```
require(iNEXT) #Rarefaction package by Chao
```

```
## Loading required package: iNEXT
```

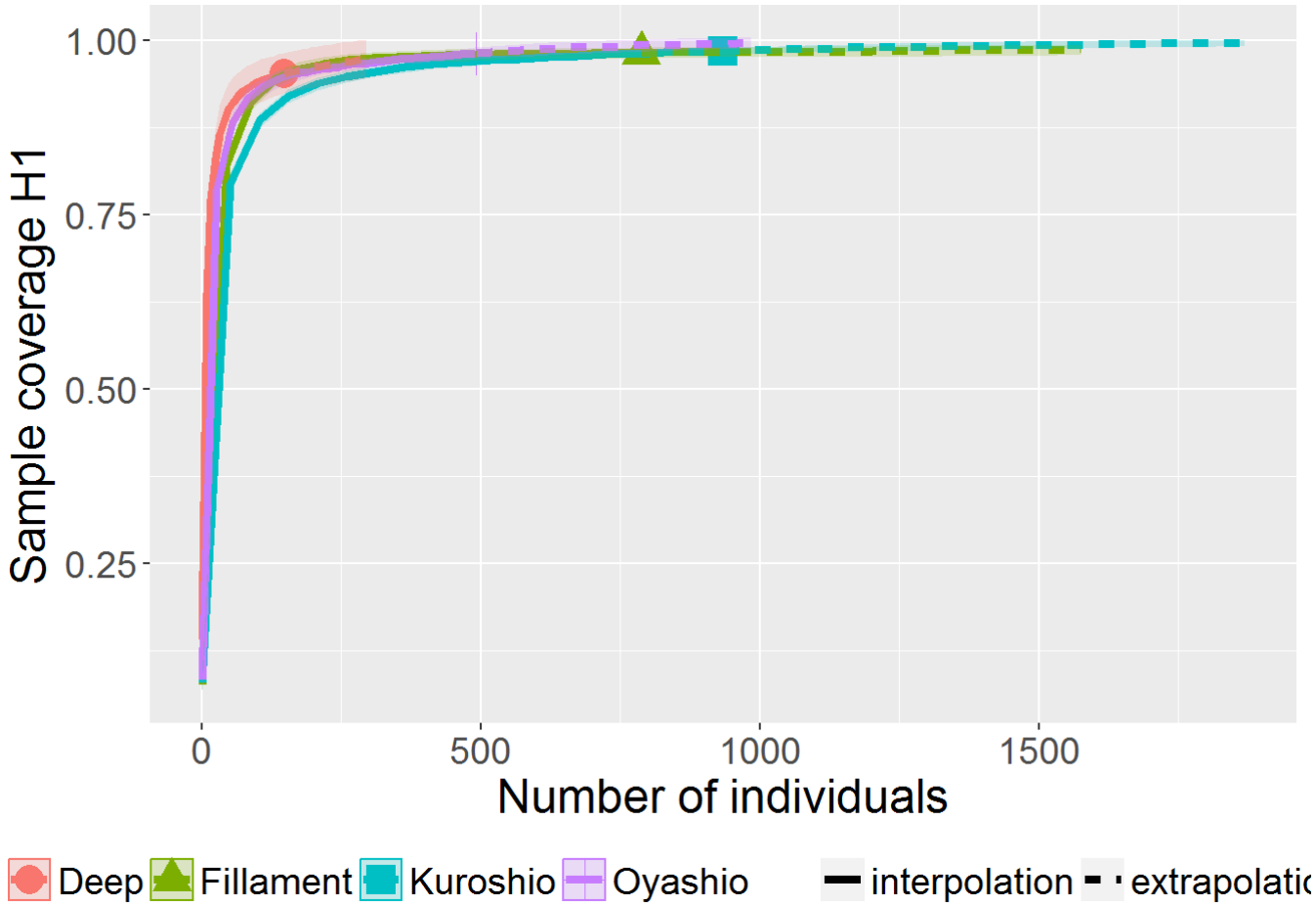
```
#INEXT creates rarefaction dataframe and metadata for a group of samples
A0 <- iNEXT(RarefactionAbund, q=0,  datatype="abundance") #richness
A1 <- iNEXT(RarefactionAbund, q=1,  datatype="abundance") #Shannon based diversity
#
#z0 <- iNEXT(Rarefaction, q=0,  datatype="Incidence_freq") #Incidence counts
#z1 <- iNEXT(Rarefaction, q=1,  datatype="incidence") #Incidence counts

#Rarefaction of abundance data showing richness
RarefactionRichness <- ggiNEXT(A0, type = 1, facet.var = "none", color.var = "site")
RarefactionRichness+ylab("Richness")
```




```
#Rarefaction of abundance data showing diversity (H1)
RarefactionH <- ggiNEXT(A1, type = 1, facet.var = "none", color.var = "site")
RarefactionH+ylab("1st order Hill #")
```




```
#Coverage plotted for 1st order hill number
RarefactionCoverage <- ggiNEXT(A1, type = 2, facet.var = "none", color.var = "site")
RarefactionCoverage+ylab("Sample coverage H1")
```




```
#Creates rarefaction graphs
```