Chapter 7 Natural Biogeochemistry and Dispersal

Alan Couch

Wed Sep 23 3:23:19 PM 2015

## Loading required package: knitr  
## Loading project configuration  
## Autoloading helper functions  
## Running helper script: otolithTransectPlot.R  
## Running helper script: otoPartChem.R  
## Autoloading cache  
## Autoloading data  
## Loading data set: allOtoChemData  
## Loading data set: DMac14.1567DistMatrix  
## Loading data set: DMac14.1567snps  
## Loading data set: DMac14.1567snps2  
## Loading data set: qslAllLarvaInfo  
## Loading data set: qslAllLarvaInfoold  
## Loading data set: qslAllLarvaInfoold2  
## Loading data set: qslDartCovariates  
## Loading data set: qslLarvaeAgePlus  
## Munging data  
## Running preprocessing script: 01MungeGeneticsData.R  
## Running preprocessing script: 02MungeChemAverages.R

This document includes methods, results and possibly some discussion dot points for the biogeochemistry chapter.

source("http://addictedtor.free.fr/packages/A2R/lastVersion/R/code.R")# load code of A2R function  
library(ggplot2)  
library(ggdendro)  
library(ape)  
library(dendextend)  
library(Hmisc)  
library(ade4)

## Method

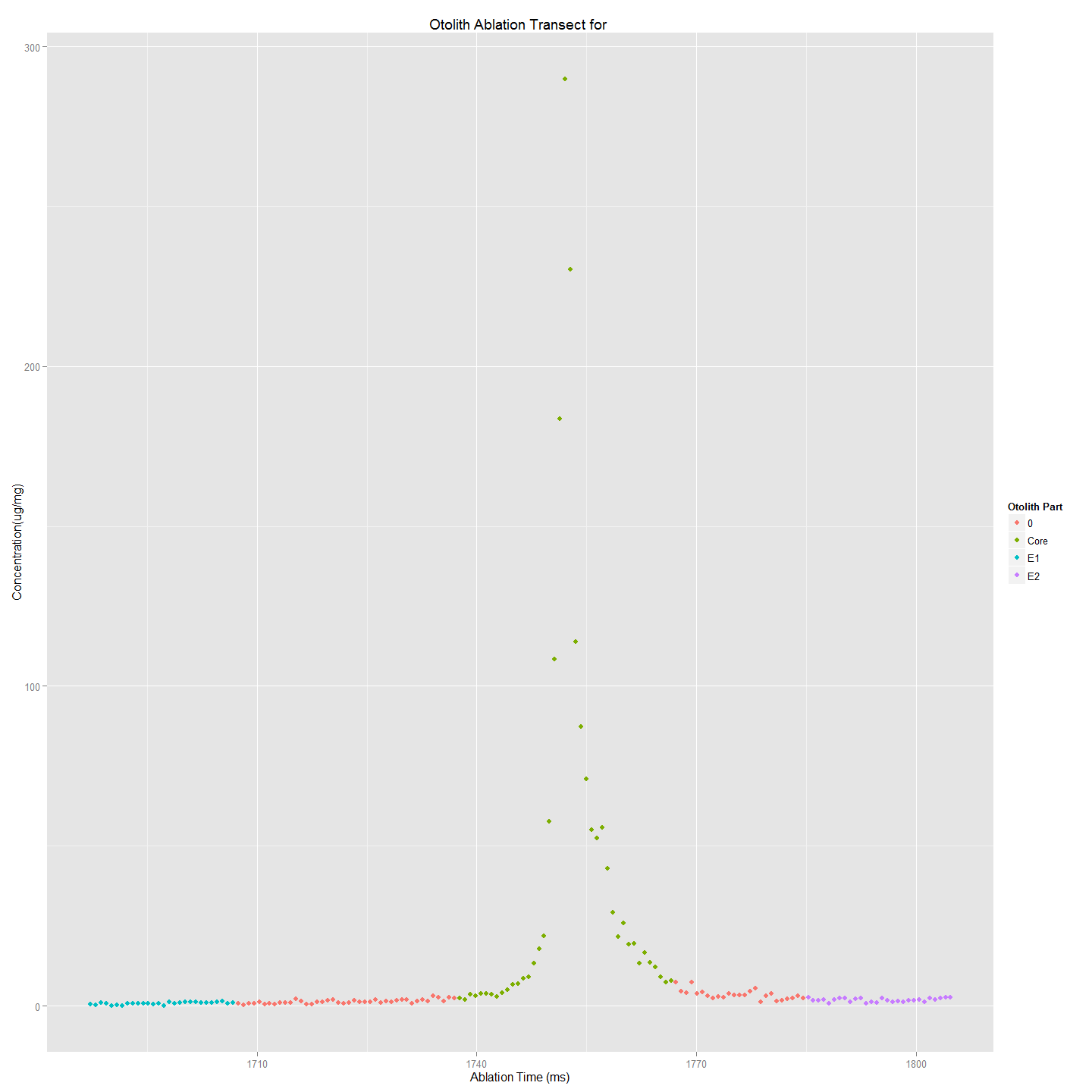
The first effort is based on the location of collection. Later, if there is a corelation between 'chemotype distance' and geographic distance we will look at the possibility of using an offset downstream of the origin of the mother. It is not likely possible as with larvae we could use larval 'drift' age days to get infrormation about each larvae. We can do this with the mothers but can we do something with the clade groups?

### Identify Important Chemistry Variables for Prediction of Site

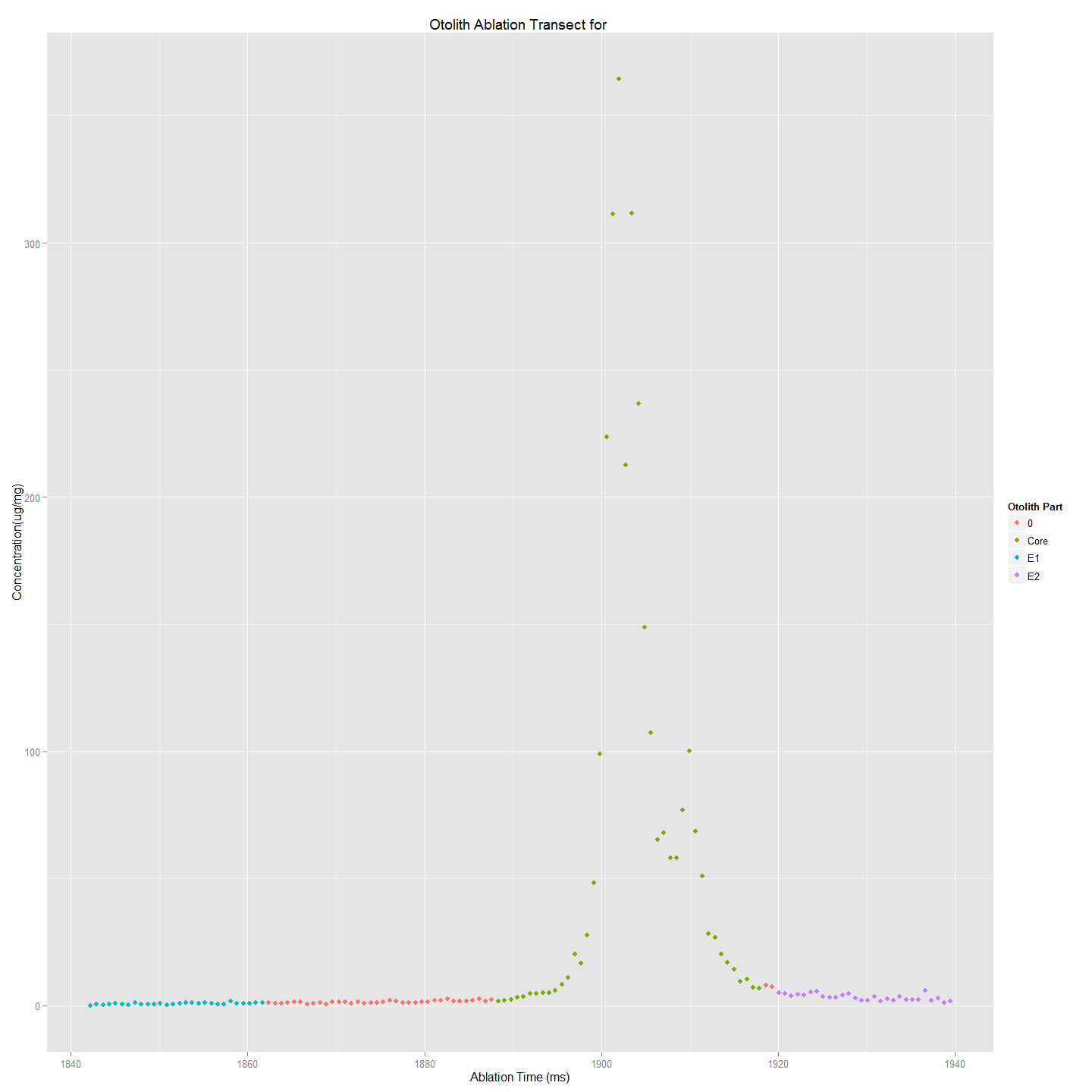
In order to identify important variables in a multivariate dataset one can utilize machine learning methods. There are many different machine learning algorithms for different tasks. One common task is to decide if a feature vector belongs to a certain class. This can be done with a random forest classifier. In order to do so, one has to train the classifier with training data first. Then the classifier can be used to predict the class of other feature vectors.<http://proven-inconclusive.com/blog/machine_learning_methods_to_identify_important_variables.html>. There is no need for other tests, such as cross-validation, to get an unbiased estimate of the test set error as each tree is created with a different bootstrap sample [2].

The classifier saves information on feature importance ("importance=TRUE"). We can use this information in order to identify potentially import variables in the data set.

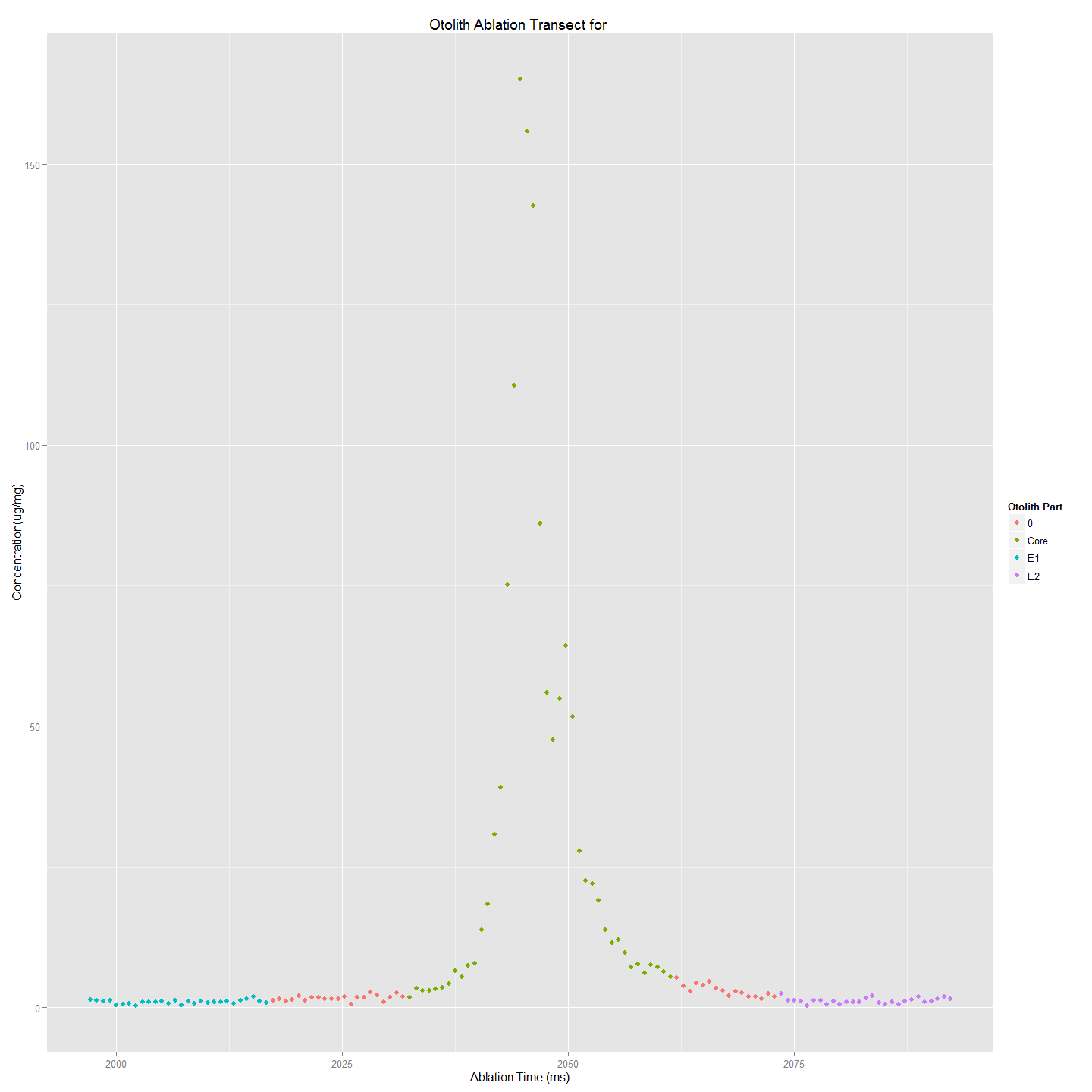
## [[1]]



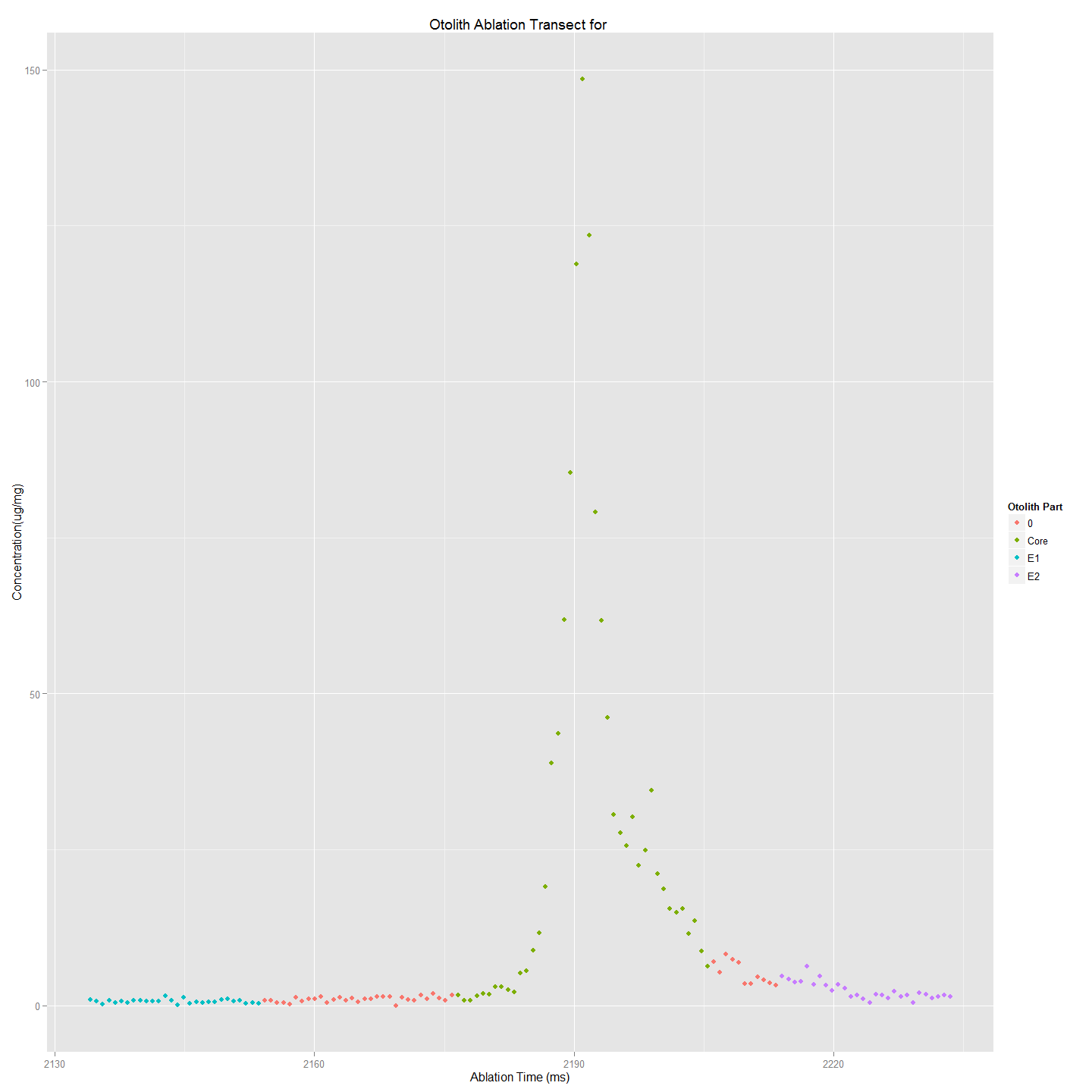
##   
## [[2]]



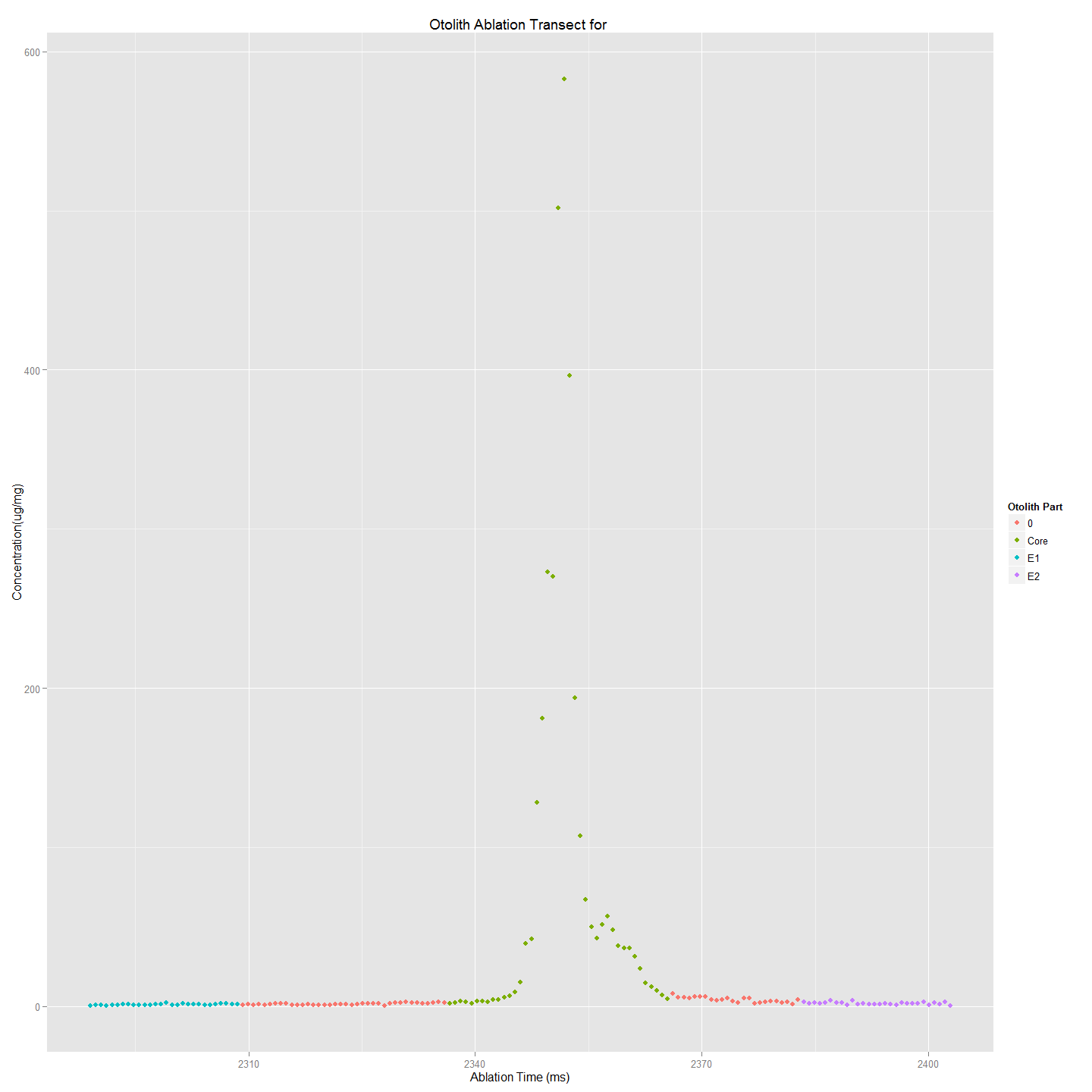
##   
## [[3]]



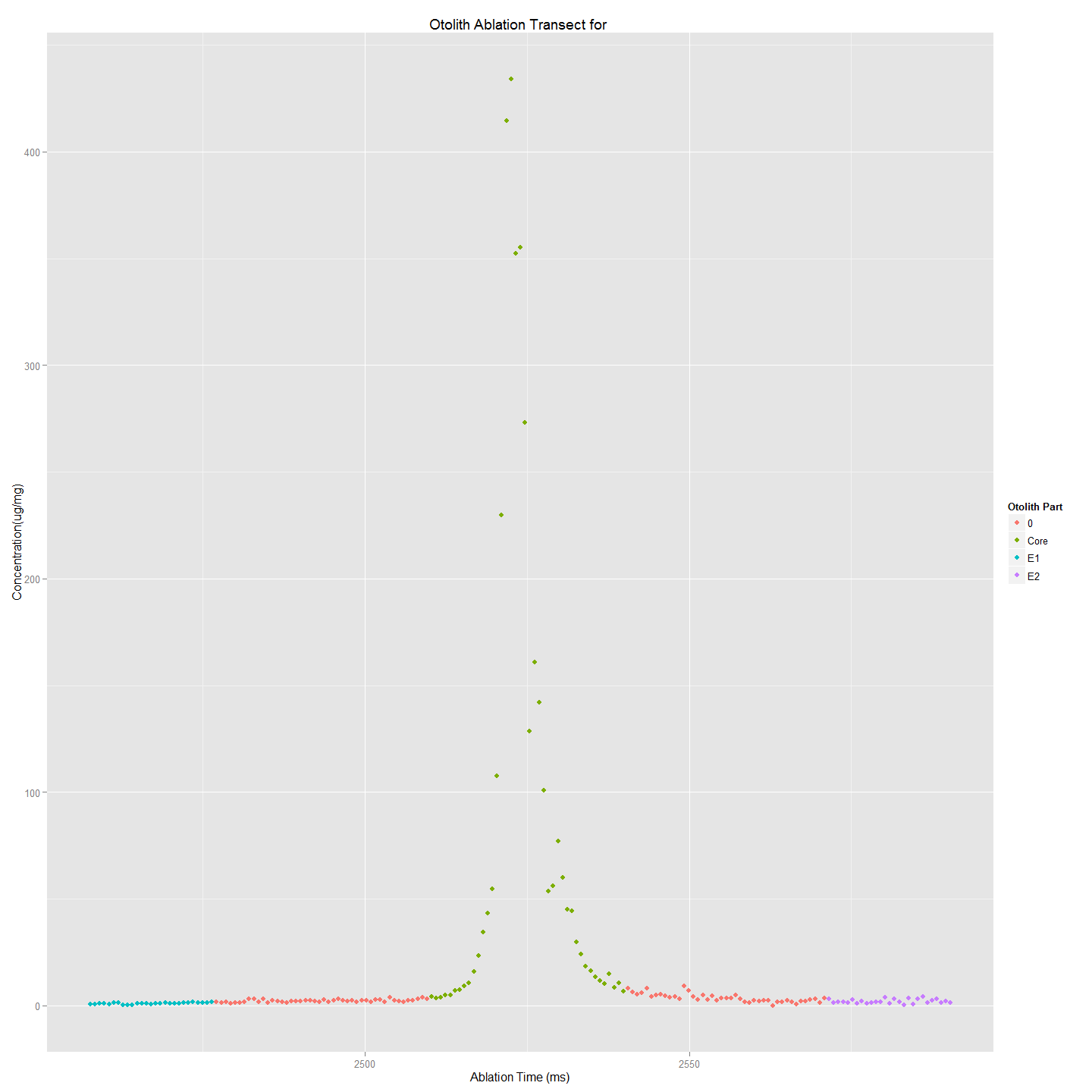
##   
## [[4]]



##   
## [[5]]



##   
## [[6]]



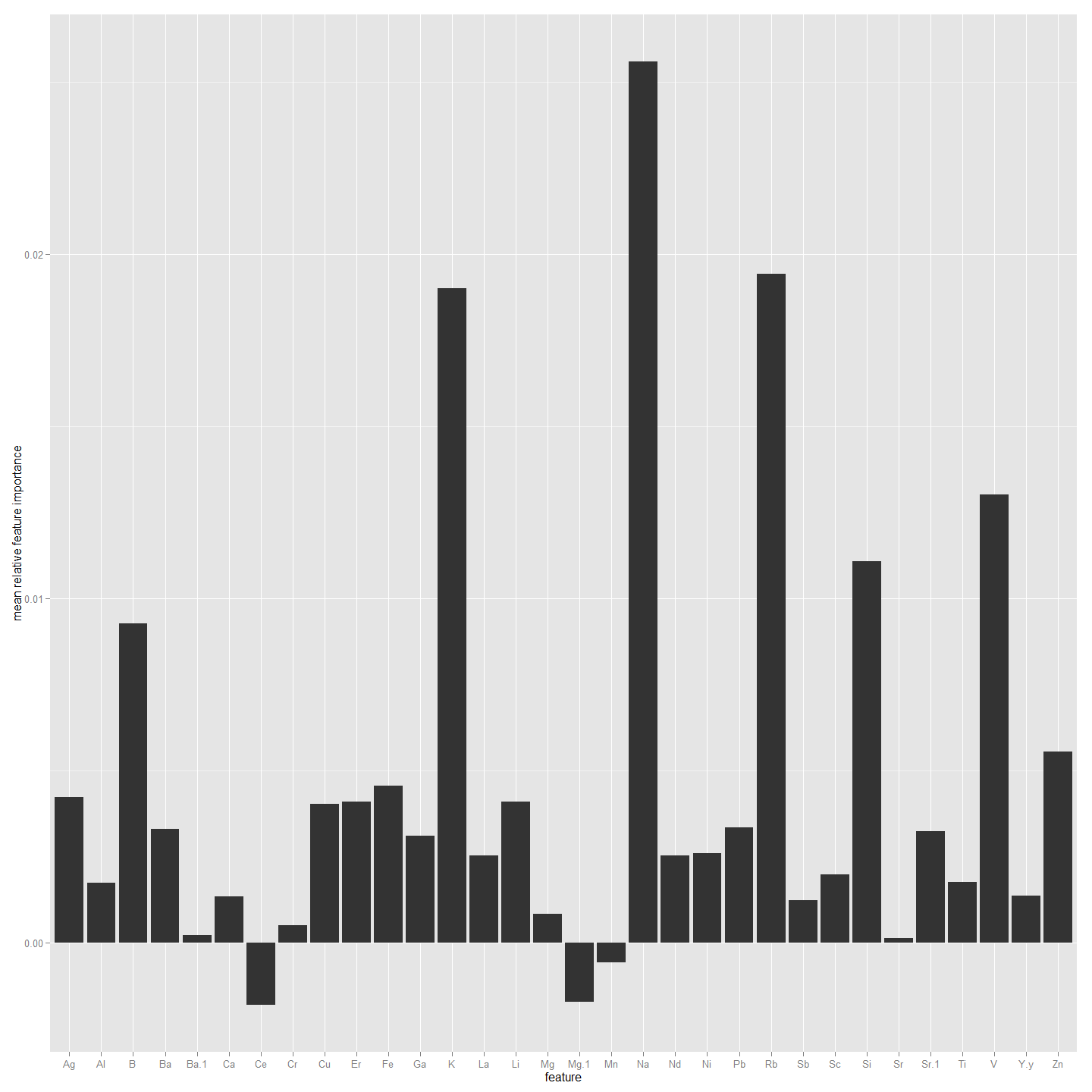
### Results

#### First with Elements Alone (Six OtoParts)

require(dplyr)  
#This identifies the important Chemistry factors that could be used to predict site the larva are from.  
#First using Core as the values. So need to combine first  
ImportantVars<-ChemAnalCore  
  
ImportantVars<- select(ImportantVars, SiteName, Li:Pb)#ImportantVars<-ImportantVars[c(84,120:151)] #  
ImportantVars<-ImportantVars[complete.cases(ImportantVars),] #remove any nulls  
colnames(ImportantVars)[1]<-"SiteName"  
ImportantVars$SiteName<-as.factor(ImportantVars$SiteName)  
  
ImportantVars <- droplevels(ImportantVars)#Not sure why get error without this line.  
  
library(randomForest)  
forest <- randomForest(SiteName ~.,data=ImportantVars, importance=TRUE)  
  
forest

##   
## Call:  
## randomForest(formula = SiteName ~ ., data = ImportantVars, importance = TRUE)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 5  
##   
## OOB estimate of error rate: 35%  
## Confusion matrix:  
## Bullen Range Kambah Pool Lanyon Nerreman Tharwa Sandwash class.error  
## Bullen Range 0 7 0 1 0 1.00000000  
## Kambah Pool 1 42 0 0 0 0.02325581  
## Lanyon 0 2 0 1 0 1.00000000  
## Nerreman 0 9 0 10 0 0.47368421  
## Tharwa Sandwash 0 5 0 2 0 1.00000000

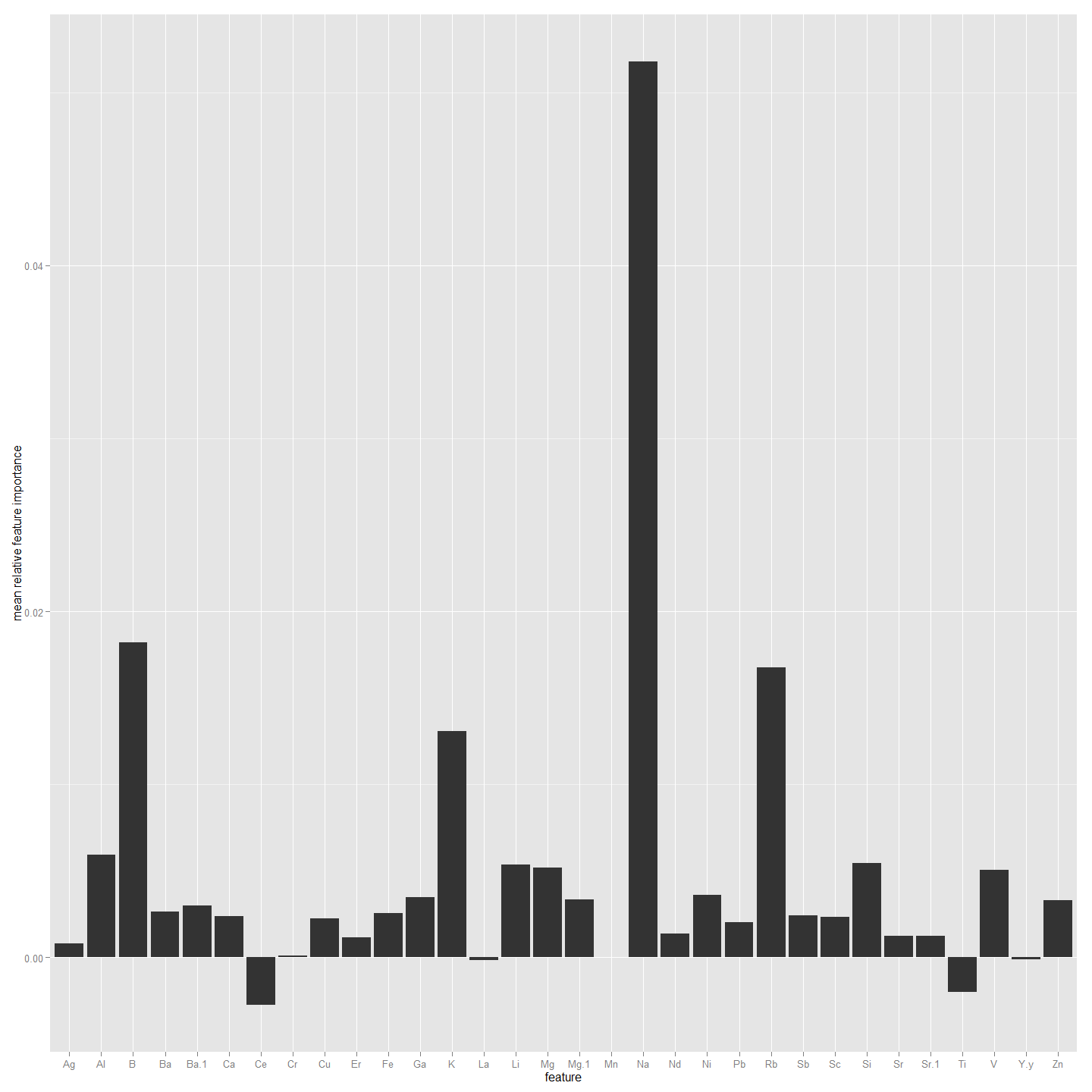
library(ggplot2)  
forest.importance = as.data.frame(importance(forest, scale=FALSE))  
forest.importance = forest.importance[,1:(ncol(forest.importance)-2)]  
forest.importance$mean = rowMeans(forest.importance)  
  
#forest.importance  
  
ggplot(forest.importance, aes(x=row.names(forest.importance), y=mean)) +  
 ylab('mean relative feature importance') +  
 xlab('feature') +  
 geom\_bar(stat='identity')



##################################################################################  
#Now to do the same but using Edge1 as the values. So need to combine first  
ImportantVars<-ChemAnalE1  
  
ImportantVars<- select(ImportantVars, SiteName, Li:Pb)#ImportantVars<-ImportantVars[c(84,120:151)] #  
ImportantVars<-ImportantVars[complete.cases(ImportantVars),] #remove any nulls  
colnames(ImportantVars)[1]<-"SiteName"  
ImportantVars$SiteName<-as.factor(ImportantVars$SiteName)  
  
ImportantVars <- droplevels(ImportantVars)#Not sure why get error without this line.  
  
library(randomForest)  
forest <- randomForest(SiteName ~.,data=ImportantVars, importance=TRUE)  
  
forest

##   
## Call:  
## randomForest(formula = SiteName ~ ., data = ImportantVars, importance = TRUE)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 5  
##   
## OOB estimate of error rate: 37.35%  
## Confusion matrix:  
## Bullen Range Kambah Pool Lanyon Nerreman Tharwa Sandwash class.error  
## Bullen Range 2 7 0 0 0 0.7777778  
## Kambah Pool 0 38 0 5 0 0.1162791  
## Lanyon 0 2 0 2 0 1.0000000  
## Nerreman 0 8 0 12 0 0.4000000  
## Tharwa Sandwash 0 7 0 0 0 1.0000000

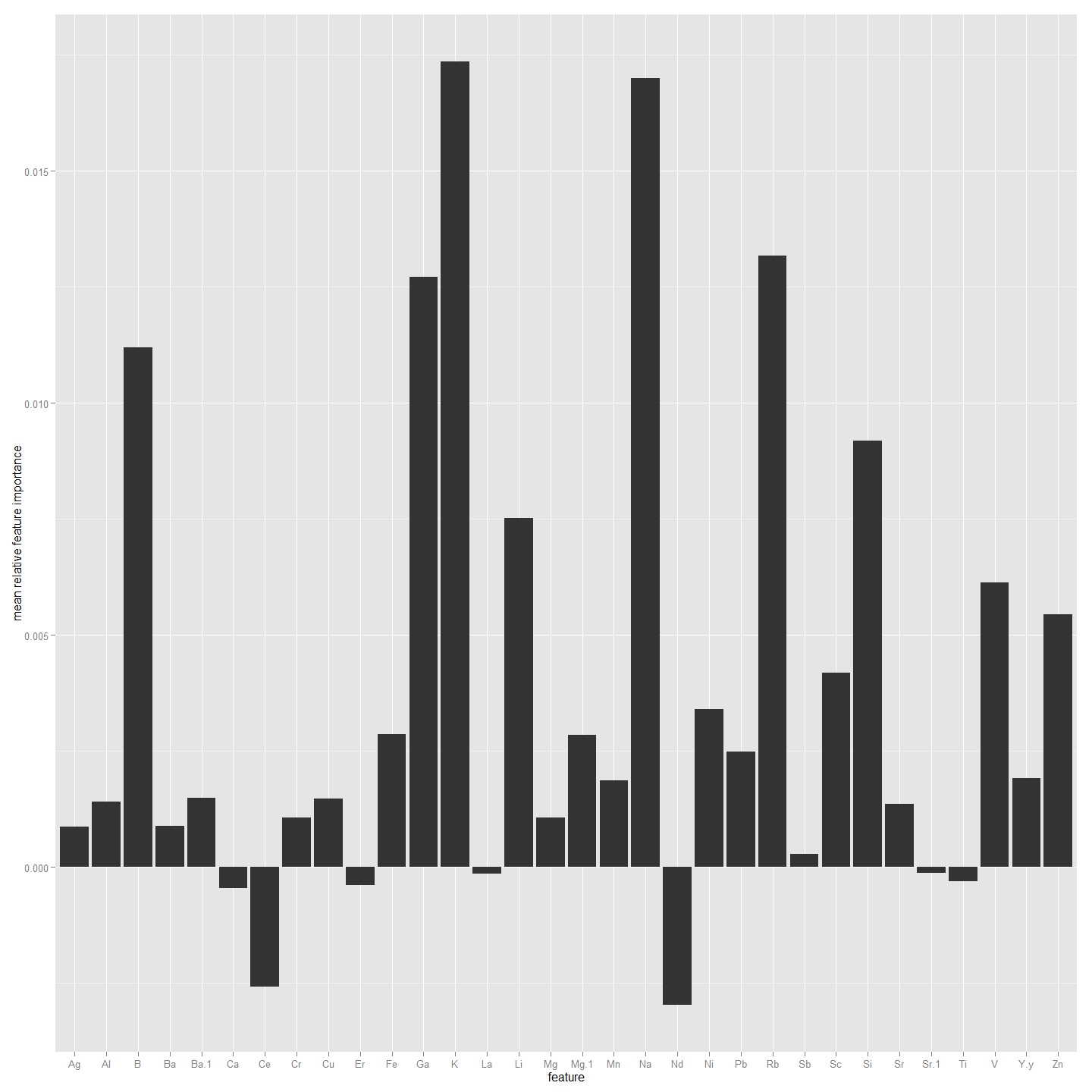
library(ggplot2)  
forest.importance = as.data.frame(importance(forest, scale=FALSE))  
forest.importance = forest.importance[,1:(ncol(forest.importance)-2)]  
forest.importance$mean = rowMeans(forest.importance)  
  
#forest.importance  
  
ggplot(forest.importance, aes(x=row.names(forest.importance), y=mean)) +  
 ylab('mean relative feature importance') +  
 xlab('feature') +  
 geom\_bar(stat='identity')



################################################################################################  
#Now to do the same but using Edge2 as the factor rather than SiteName. So need to combine first  
ImportantVars<-ChemAnalE2  
  
ImportantVars<- select(ImportantVars, SiteName, Li:Pb)#ImportantVars<-ImportantVars[c(84,120:151)] #  
ImportantVars<-ImportantVars[complete.cases(ImportantVars),] #remove any nulls  
colnames(ImportantVars)[1]<-"SiteName"  
ImportantVars$SiteName<-as.factor(ImportantVars$SiteName)  
  
ImportantVars <- droplevels(ImportantVars)#Not sure why get error without this line.  
  
library(randomForest)  
forest <- randomForest(SiteName ~.,data=ImportantVars, importance=TRUE)  
  
forest

##   
## Call:  
## randomForest(formula = SiteName ~ ., data = ImportantVars, importance = TRUE)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 5  
##   
## OOB estimate of error rate: 39.02%  
## Confusion matrix:  
## Bullen Range Kambah Pool Lanyon Nerreman Tharwa Sandwash class.error  
## Bullen Range 3 6 0 0 0 0.6666667  
## Kambah Pool 2 37 0 4 0 0.1395349  
## Lanyon 0 3 0 1 0 1.0000000  
## Nerreman 0 9 0 10 0 0.4736842  
## Tharwa Sandwash 0 6 0 1 0 1.0000000

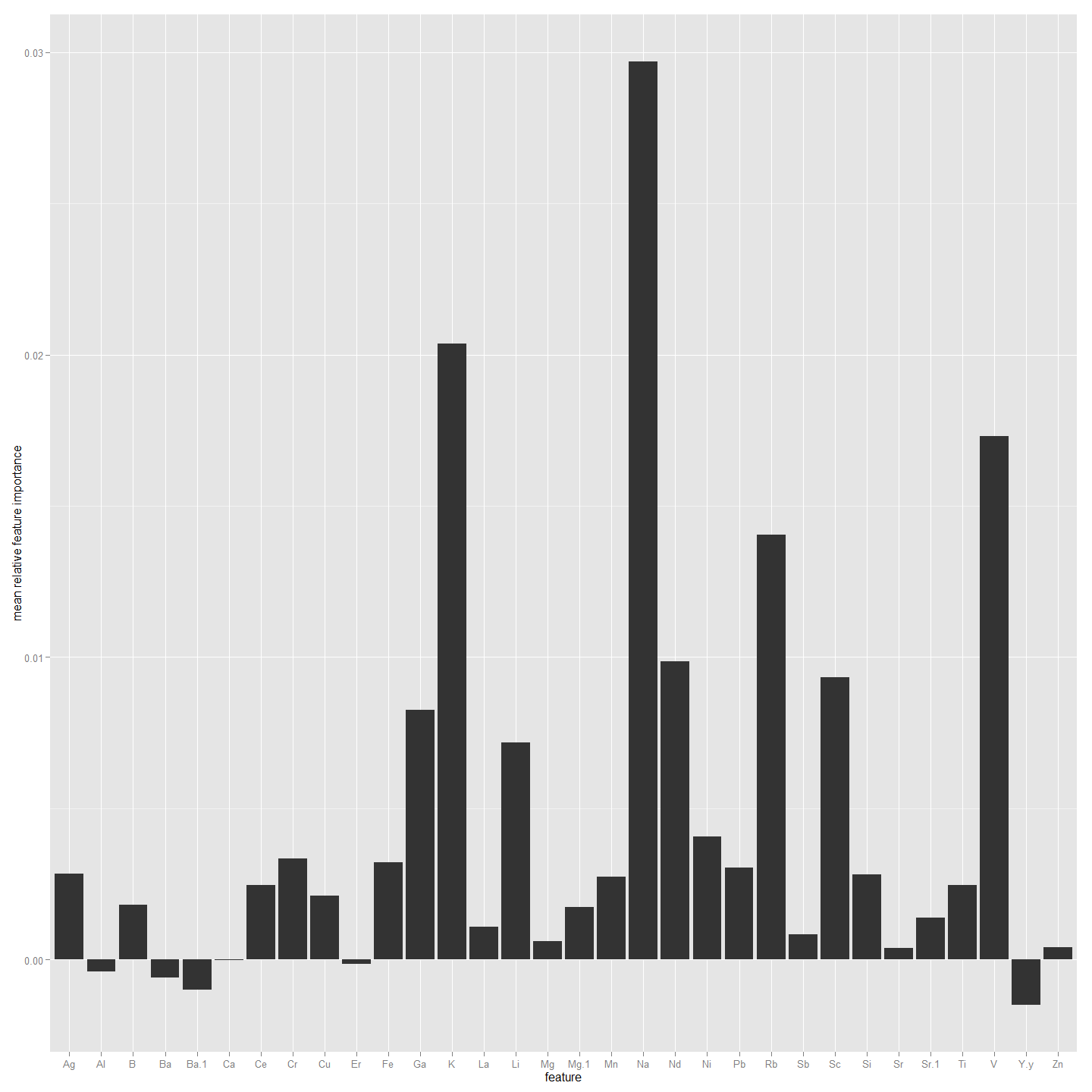
library(ggplot2)  
forest.importance = as.data.frame(importance(forest, scale=FALSE))  
forest.importance = forest.importance[,1:(ncol(forest.importance)-2)]  
forest.importance$mean = rowMeans(forest.importance)  
  
#forest.importance  
  
ggplot(forest.importance, aes(x=row.names(forest.importance), y=mean)) +  
 ylab('mean relative feature importance') +  
 xlab('feature') +  
 geom\_bar(stat='identity')



##################################################################################################  
#Now to do the same but using Edge0 values. So need to combine first  
ImportantVars<-ChemAnal0  
  
ImportantVars<- select(ImportantVars, SiteName, Li:Pb)#ImportantVars<-ImportantVars[c(84,120:151)] #  
ImportantVars<-ImportantVars[complete.cases(ImportantVars),] #remove any nulls  
colnames(ImportantVars)[1]<-"SiteName"  
ImportantVars$SiteName<-as.factor(ImportantVars$SiteName)  
  
ImportantVars <- droplevels(ImportantVars)#Not sure why get error without this line.  
  
library(randomForest)  
forest <- randomForest(SiteName ~.,data=ImportantVars, importance=TRUE)  
  
forest

##   
## Call:  
## randomForest(formula = SiteName ~ ., data = ImportantVars, importance = TRUE)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 5  
##   
## OOB estimate of error rate: 26.67%  
## Confusion matrix:  
## Bullen Range Kambah Pool Lanyon Nerreman Tharwa Sandwash class.error  
## Bullen Range 0 3 0 1 0 1.0000000  
## Kambah Pool 1 31 0 4 0 0.1388889  
## Lanyon 0 0 0 2 0 1.0000000  
## Nerreman 0 4 0 13 0 0.2352941  
## Tharwa Sandwash 0 1 0 0 0 1.0000000

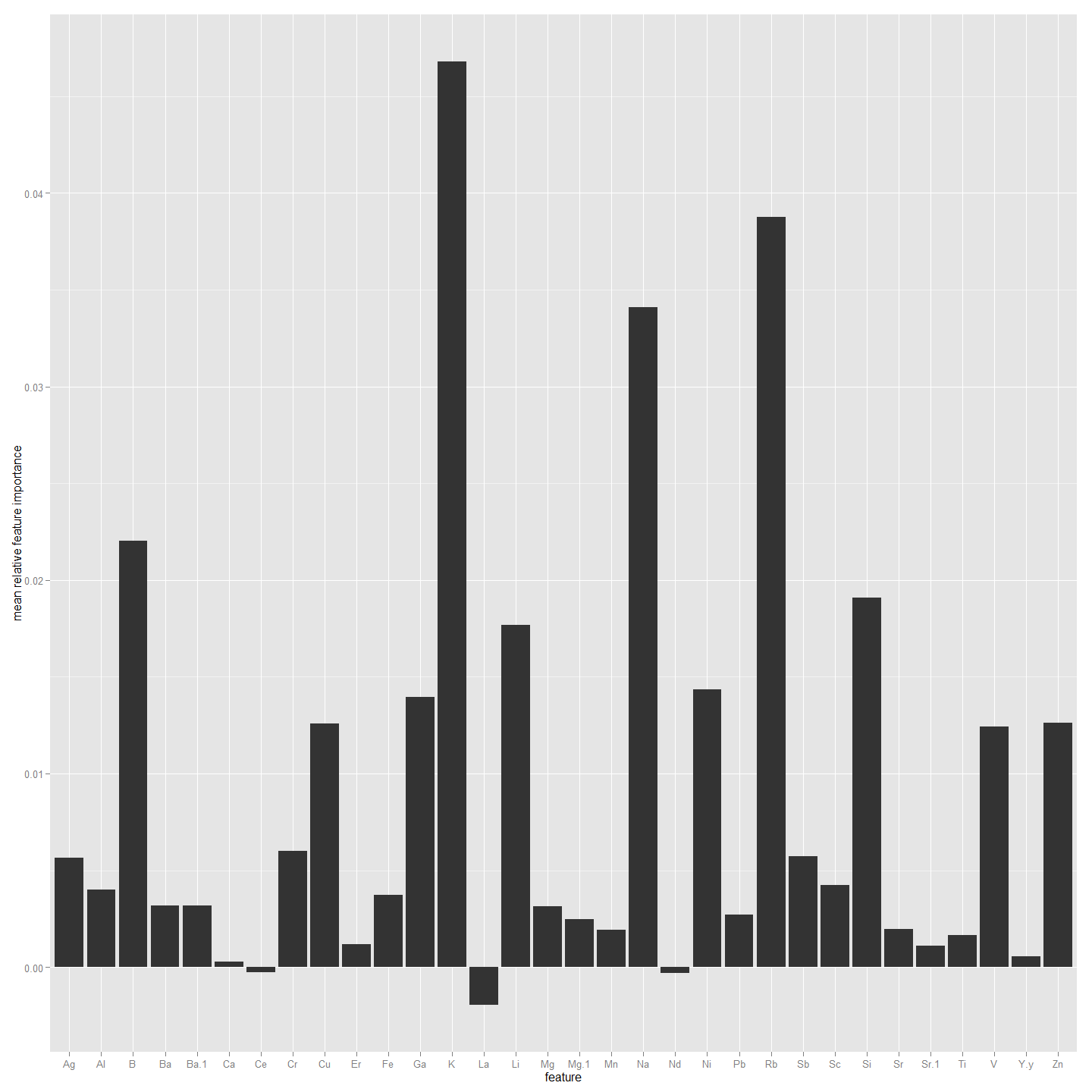
library(ggplot2)  
forest.importance = as.data.frame(importance(forest, scale=FALSE))  
forest.importance = forest.importance[,1:(ncol(forest.importance)-2)]  
forest.importance$mean = rowMeans(forest.importance)  
  
#forest.importance  
  
ggplot(forest.importance, aes(x=row.names(forest.importance), y=mean)) +  
 ylab('mean relative feature importance') +  
 xlab('feature') +  
 geom\_bar(stat='identity')



##################################################################################################  
#Now to do the same but using All4 values. So need to combine first  
ImportantVars<-ChemAnalAll4  
  
ImportantVars<- select(ImportantVars, SiteName, Li:Pb)#ImportantVars<-ImportantVars[c(84,120:151)] #  
ImportantVars<-ImportantVars[complete.cases(ImportantVars),] #remove any nulls  
colnames(ImportantVars)[1]<-"SiteName"  
ImportantVars$SiteName<-as.factor(ImportantVars$SiteName)  
  
ImportantVars <- droplevels(ImportantVars)#Not sure why get error without this line.  
  
library(randomForest)  
forest <- randomForest(SiteName ~.,data=ImportantVars, importance=TRUE)  
  
forest

##   
## Call:  
## randomForest(formula = SiteName ~ ., data = ImportantVars, importance = TRUE)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 5  
##   
## OOB estimate of error rate: 28.2%  
## Confusion matrix:  
## Bullen Range Kambah Pool Lanyon Nerreman Tharwa Sandwash class.error  
## Bullen Range 9 19 0 2 0 0.70000000  
## Kambah Pool 1 157 0 7 0 0.04848485  
## Lanyon 0 7 1 5 0 0.92307692  
## Nerreman 0 23 0 52 0 0.30666667  
## Tharwa Sandwash 0 18 0 4 0 1.00000000

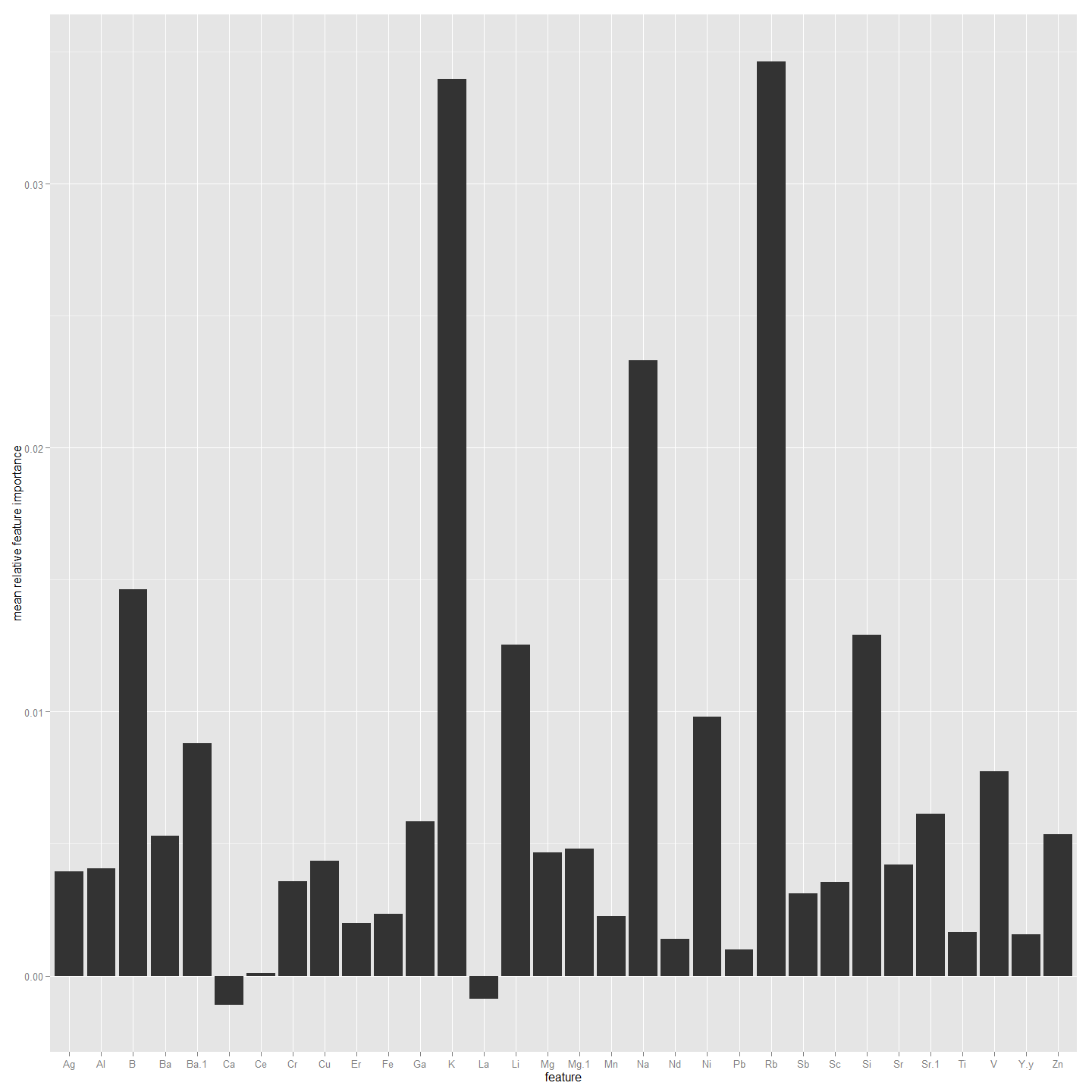
library(ggplot2)  
forest.importance = as.data.frame(importance(forest, scale=FALSE))  
forest.importance = forest.importance[,1:(ncol(forest.importance)-2)]  
forest.importance$mean = rowMeans(forest.importance)  
  
#forest.importance  
  
ggplot(forest.importance, aes(x=row.names(forest.importance), y=mean)) +  
 ylab('mean relative feature importance') +  
 xlab('feature') +  
 geom\_bar(stat='identity')



##################################################################################################  
#Now to do the same but using Non-Core values. So need to combine first  
ImportantVars<-ChemAnalNonCore  
  
ImportantVars<- select(ImportantVars, SiteName, Li:Pb)#ImportantVars<-ImportantVars[c(84,120:151)] #  
ImportantVars<-ImportantVars[complete.cases(ImportantVars),] #remove any nulls  
colnames(ImportantVars)[1]<-"SiteName"  
ImportantVars$SiteName<-as.factor(ImportantVars$SiteName)  
  
ImportantVars <- droplevels(ImportantVars)#Not sure why get error without this line.  
  
library(randomForest)  
forest <- randomForest(SiteName ~.,data=ImportantVars, importance=TRUE)  
  
forest

##   
## Call:  
## randomForest(formula = SiteName ~ ., data = ImportantVars, importance = TRUE)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 5  
##   
## OOB estimate of error rate: 31.56%  
## Confusion matrix:  
## Bullen Range Kambah Pool Lanyon Nerreman Tharwa Sandwash class.error  
## Bullen Range 6 15 0 1 0 0.72727273  
## Kambah Pool 1 111 0 10 0 0.09016393  
## Lanyon 0 6 0 4 0 1.00000000  
## Nerreman 0 19 0 37 0 0.33928571  
## Tharwa Sandwash 0 13 0 2 0 1.00000000

library(ggplot2)  
forest.importance = as.data.frame(importance(forest, scale=FALSE))  
forest.importance = forest.importance[,1:(ncol(forest.importance)-2)]  
forest.importance$mean = rowMeans(forest.importance)  
  
#forest.importance  
  
ggplot(forest.importance, aes(x=row.names(forest.importance), y=mean)) +  
 ylab('mean relative feature importance') +  
 xlab('feature') +  
 geom\_bar(stat='identity')

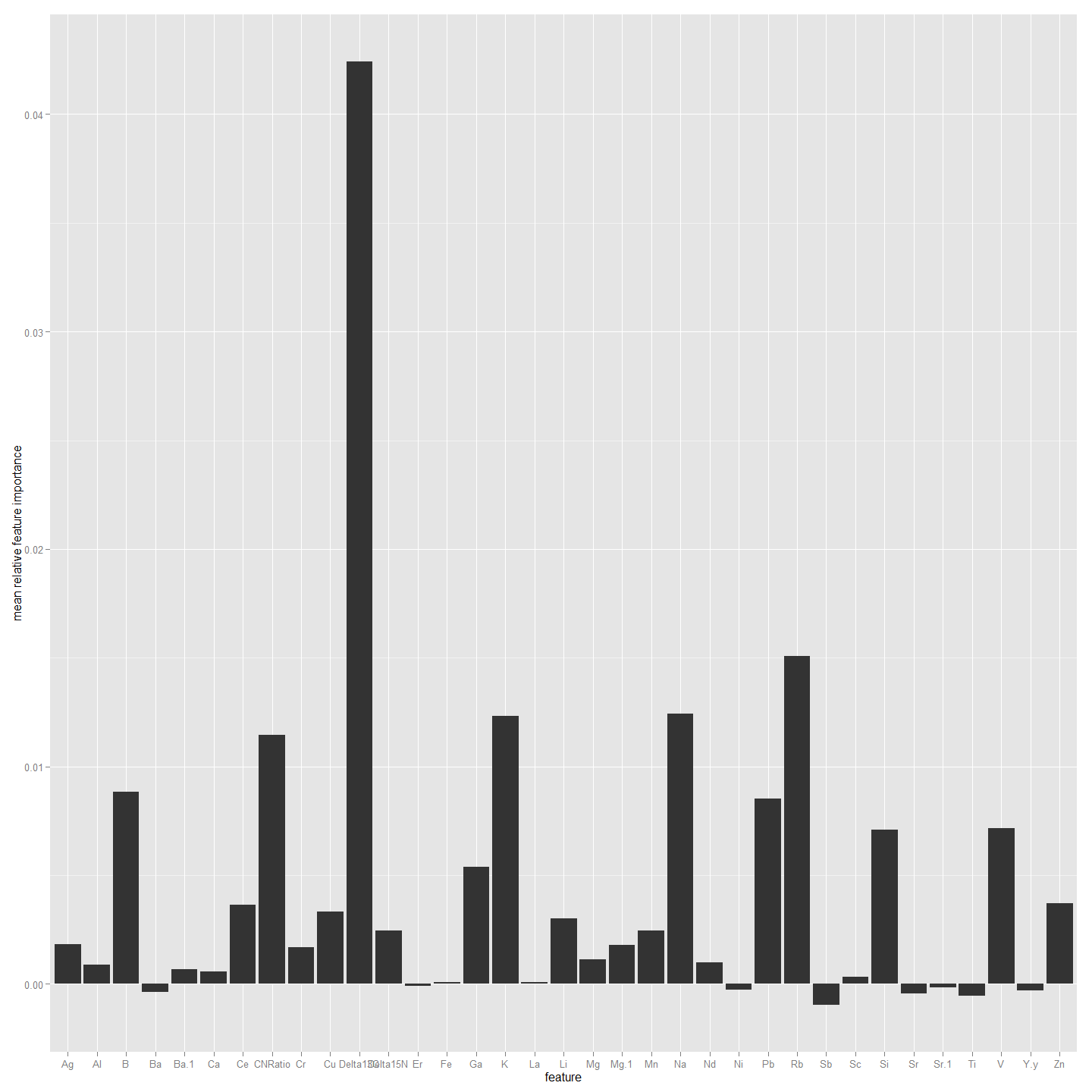


#### Then with Elements and Isotopes (Six OtoParts but reduced number of larvae)

require(dplyr)  
#This identifies the important Chemistry factors that could be used to predict site the larva are from.  
#First using Core as the values. So need to combine first  
ImportantVars<-ChemAnalCore  
  
ImportantVars<- select(ImportantVars, SiteName,Delta13C,Delta15N,CNRatio,Li:Pb)#ImportantVars<-ImportantVars[c(84,13,14,15,120:151)] #  
ImportantVars<-ImportantVars[complete.cases(ImportantVars),] #remove any nulls  
colnames(ImportantVars)[1]<-"SiteName"  
ImportantVars$SiteName<-as.factor(ImportantVars$SiteName)  
  
ImportantVars <- droplevels(ImportantVars)#Not sure why get error without this line.  
  
library(randomForest)  
forest <- randomForest(SiteName ~.,data=ImportantVars, importance=TRUE)  
  
forest

##   
## Call:  
## randomForest(formula = SiteName ~ ., data = ImportantVars, importance = TRUE)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 5  
##   
## OOB estimate of error rate: 25%  
## Confusion matrix:  
## Bullen Range Kambah Pool Lanyon Nerreman Tharwa Sandwash class.error  
## Bullen Range 1 5 0 2 0 0.8750000  
## Kambah Pool 0 43 0 0 0 0.0000000  
## Lanyon 0 2 0 1 0 1.0000000  
## Nerreman 0 6 0 13 0 0.3157895  
## Tharwa Sandwash 0 2 0 1 0 1.0000000

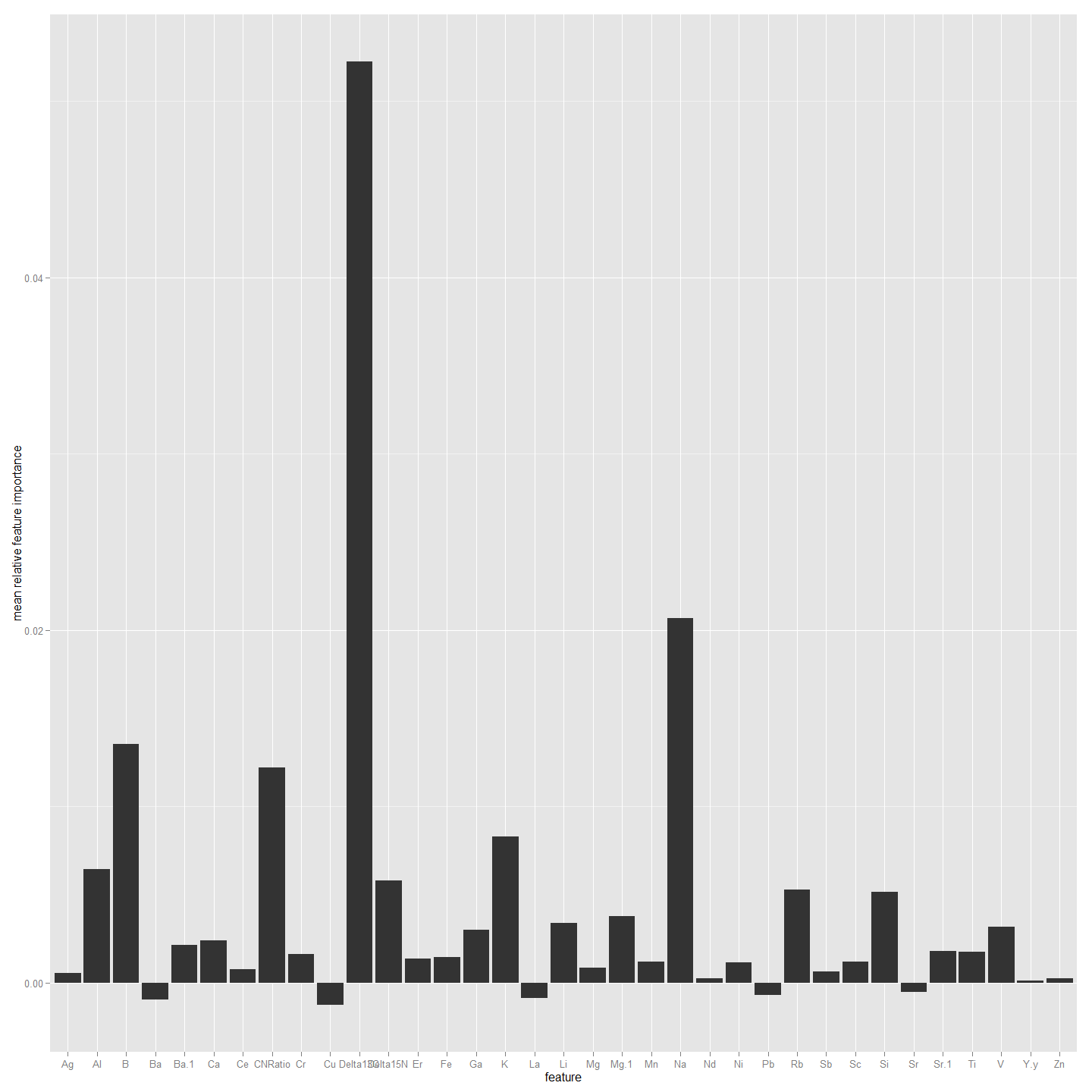
library(ggplot2)  
forest.importance = as.data.frame(importance(forest, scale=FALSE))  
forest.importance = forest.importance[,1:(ncol(forest.importance)-2)]  
forest.importance$mean = rowMeans(forest.importance)  
  
#forest.importance  
  
ggplot(forest.importance, aes(x=row.names(forest.importance), y=mean)) +  
 ylab('mean relative feature importance') +  
 xlab('feature') +  
 geom\_bar(stat='identity')



##################################################################################  
#Now to do the same but using Edge1 as the values. So need to combine first  
ImportantVars<-ChemAnalE1  
  
ImportantVars<- select(ImportantVars, SiteName,Delta13C,Delta15N,CNRatio,Li:Pb)#ImportantVars<-ImportantVars[c(97,13,14,15,120:151)] #  
ImportantVars<-ImportantVars[complete.cases(ImportantVars),] #remove any nulls  
colnames(ImportantVars)[1]<-"SiteName"  
ImportantVars$SiteName<-as.factor(ImportantVars$SiteName)  
  
ImportantVars <- droplevels(ImportantVars)#Not sure why get error without this line.  
  
library(randomForest)  
forest <- randomForest(SiteName ~.,data=ImportantVars, importance=TRUE)  
  
forest

##   
## Call:  
## randomForest(formula = SiteName ~ ., data = ImportantVars, importance = TRUE)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 5  
##   
## OOB estimate of error rate: 25.32%  
## Confusion matrix:  
## Bullen Range Kambah Pool Lanyon Nerreman Tharwa Sandwash class.error  
## Bullen Range 3 5 0 1 0 0.66666667  
## Kambah Pool 0 42 0 1 0 0.02325581  
## Lanyon 0 3 0 1 0 1.00000000  
## Nerreman 1 5 0 14 0 0.30000000  
## Tharwa Sandwash 0 2 0 1 0 1.00000000

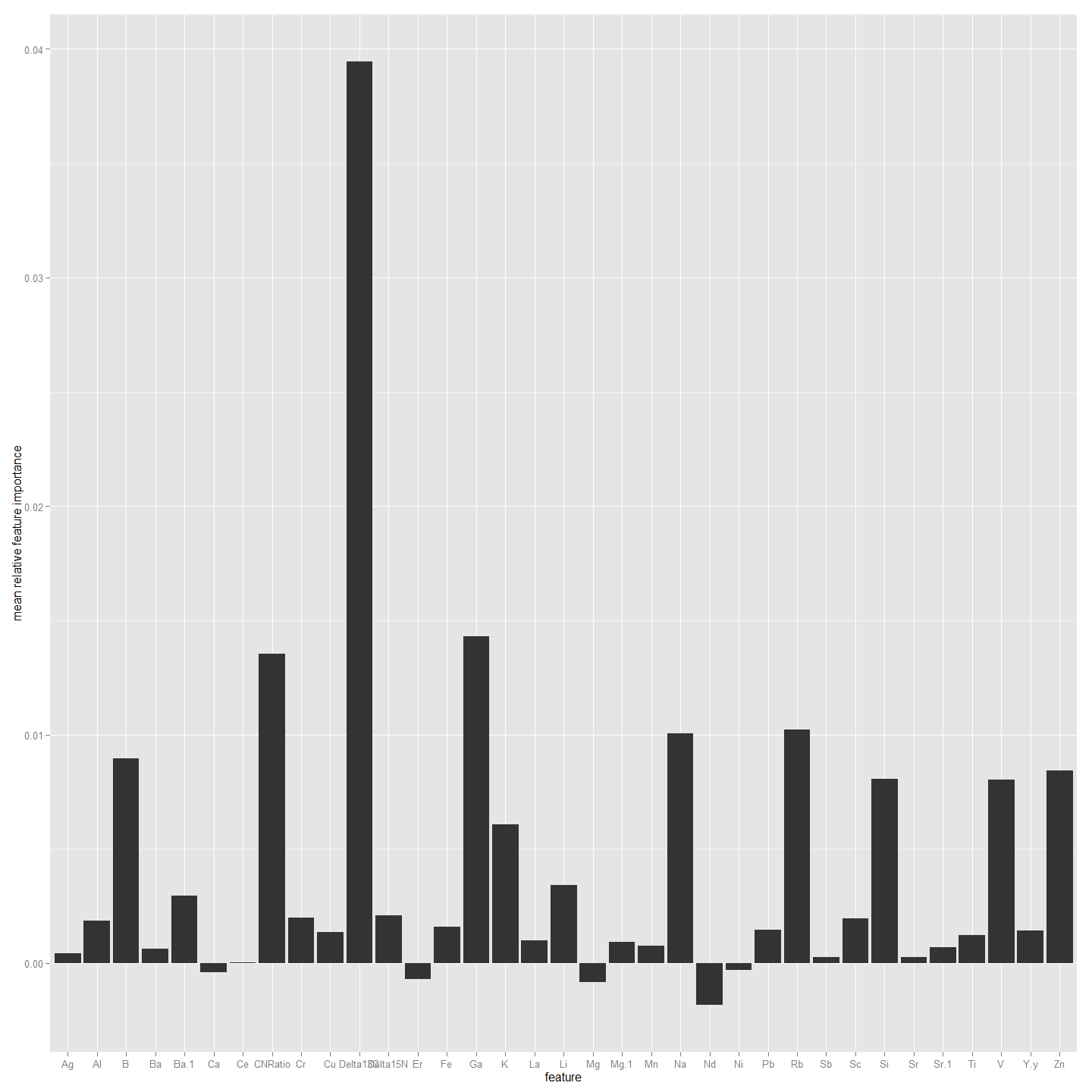
library(ggplot2)  
forest.importance = as.data.frame(importance(forest, scale=FALSE))  
forest.importance = forest.importance[,1:(ncol(forest.importance)-2)]  
forest.importance$mean = rowMeans(forest.importance)  
  
#forest.importance  
  
ggplot(forest.importance, aes(x=row.names(forest.importance), y=mean)) +  
 ylab('mean relative feature importance') +  
 xlab('feature') +  
 geom\_bar(stat='identity')



################################################################################################  
#Now to do the same but using Edge2 as the factor rather than SiteName. So need to combine first  
ImportantVars<-ChemAnalE2  
  
ImportantVars<- select(ImportantVars, SiteName,Delta13C,Delta15N,CNRatio,Li:Pb)#ImportantVars<-ImportantVars[c(97,13,14,15,120:151)] #  
ImportantVars<-ImportantVars[complete.cases(ImportantVars),] #remove any nulls  
colnames(ImportantVars)[1]<-"SiteName"  
ImportantVars$SiteName<-as.factor(ImportantVars$SiteName)  
  
ImportantVars <- droplevels(ImportantVars)#Not sure why get error without this line.  
  
library(randomForest)  
forest <- randomForest(SiteName ~.,data=ImportantVars, importance=TRUE)  
  
forest

##   
## Call:  
## randomForest(formula = SiteName ~ ., data = ImportantVars, importance = TRUE)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 5  
##   
## OOB estimate of error rate: 23.08%  
## Confusion matrix:  
## Bullen Range Kambah Pool Lanyon Nerreman Tharwa Sandwash class.error  
## Bullen Range 3 5 0 1 0 0.6666667  
## Kambah Pool 0 43 0 0 0 0.0000000  
## Lanyon 0 3 0 1 0 1.0000000  
## Nerreman 0 5 0 14 0 0.2631579  
## Tharwa Sandwash 0 2 0 1 0 1.0000000

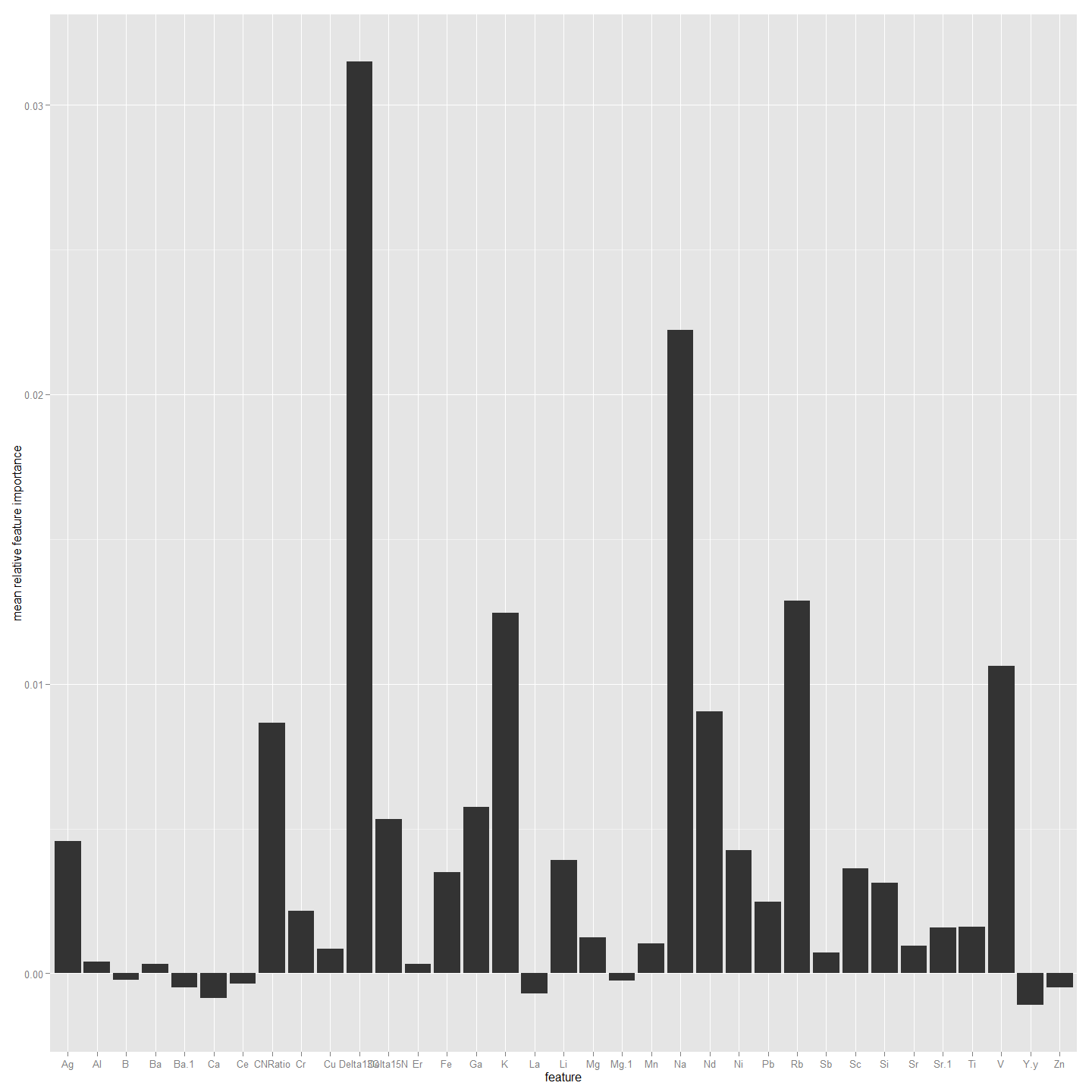
library(ggplot2)  
forest.importance = as.data.frame(importance(forest, scale=FALSE))  
forest.importance = forest.importance[,1:(ncol(forest.importance)-2)]  
forest.importance$mean = rowMeans(forest.importance)  
  
#forest.importance  
  
ggplot(forest.importance, aes(x=row.names(forest.importance), y=mean)) +  
 ylab('mean relative feature importance') +  
 xlab('feature') +  
 geom\_bar(stat='identity')



##################################################################################################  
#Now to do the same but using Edge0 values. So need to combine first  
ImportantVars<-ChemAnal0  
  
ImportantVars<- select(ImportantVars, SiteName,Delta13C,Delta15N,CNRatio,Li:Pb)#ImportantVars<-ImportantVars[c(97,13,14,15,120:151)] #  
ImportantVars<-ImportantVars[complete.cases(ImportantVars),] #remove any nulls  
colnames(ImportantVars)[1]<-"SiteName"  
ImportantVars$SiteName<-as.factor(ImportantVars$SiteName)  
  
ImportantVars <- droplevels(ImportantVars)#Not sure why get error without this line.  
  
library(randomForest)  
forest <- randomForest(SiteName ~.,data=ImportantVars, importance=TRUE)  
  
forest

##   
## Call:  
## randomForest(formula = SiteName ~ ., data = ImportantVars, importance = TRUE)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 5  
##   
## OOB estimate of error rate: 23.33%  
## Confusion matrix:  
## Bullen Range Kambah Pool Lanyon Nerreman Tharwa Sandwash class.error  
## Bullen Range 0 4 0 0 0 1.0000000  
## Kambah Pool 1 32 0 3 0 0.1111111  
## Lanyon 0 0 0 2 0 1.0000000  
## Nerreman 0 3 0 14 0 0.1764706  
## Tharwa Sandwash 0 1 0 0 0 1.0000000

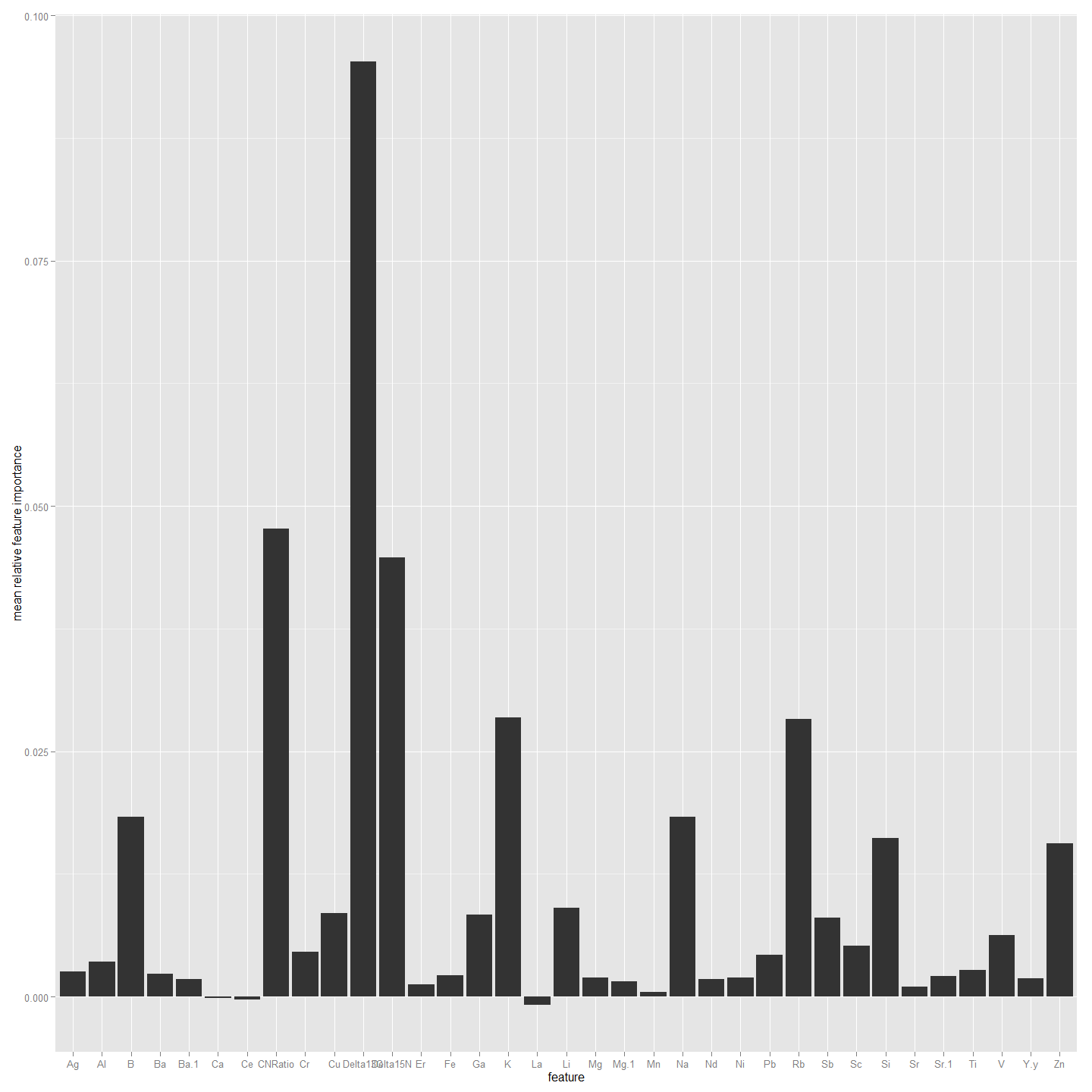
library(ggplot2)  
forest.importance = as.data.frame(importance(forest, scale=FALSE))  
forest.importance = forest.importance[,1:(ncol(forest.importance)-2)]  
forest.importance$mean = rowMeans(forest.importance)  
  
#forest.importance  
  
ggplot(forest.importance, aes(x=row.names(forest.importance), y=mean)) +  
 ylab('mean relative feature importance') +  
 xlab('feature') +  
 geom\_bar(stat='identity')



##################################################################################################  
#Now to do the same but using All4 values. So need to combine first  
ImportantVars<-ChemAnalAll4  
  
ImportantVars<- select(ImportantVars, SiteName,Delta13C,Delta15N,CNRatio,Li:Pb)#ImportantVars<-ImportantVars[c(97,13,14,15,120:151)] #  
ImportantVars<-ImportantVars[complete.cases(ImportantVars),] #remove any nulls  
colnames(ImportantVars)[1]<-"SiteName"  
ImportantVars$SiteName<-as.factor(ImportantVars$SiteName)  
  
ImportantVars <- droplevels(ImportantVars)#Not sure why get error without this line.  
  
library(randomForest)  
forest <- randomForest(SiteName ~.,data=ImportantVars, importance=TRUE)  
  
forest

##   
## Call:  
## randomForest(formula = SiteName ~ ., data = ImportantVars, importance = TRUE)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 5  
##   
## OOB estimate of error rate: 16.04%  
## Confusion matrix:  
## Bullen Range Kambah Pool Lanyon Nerreman Tharwa Sandwash class.error  
## Bullen Range 12 15 0 3 0 0.600000000  
## Kambah Pool 1 164 0 0 0 0.006060606  
## Lanyon 0 8 2 3 0 0.846153846  
## Nerreman 0 7 0 68 0 0.093333333  
## Tharwa Sandwash 0 7 0 3 0 1.000000000

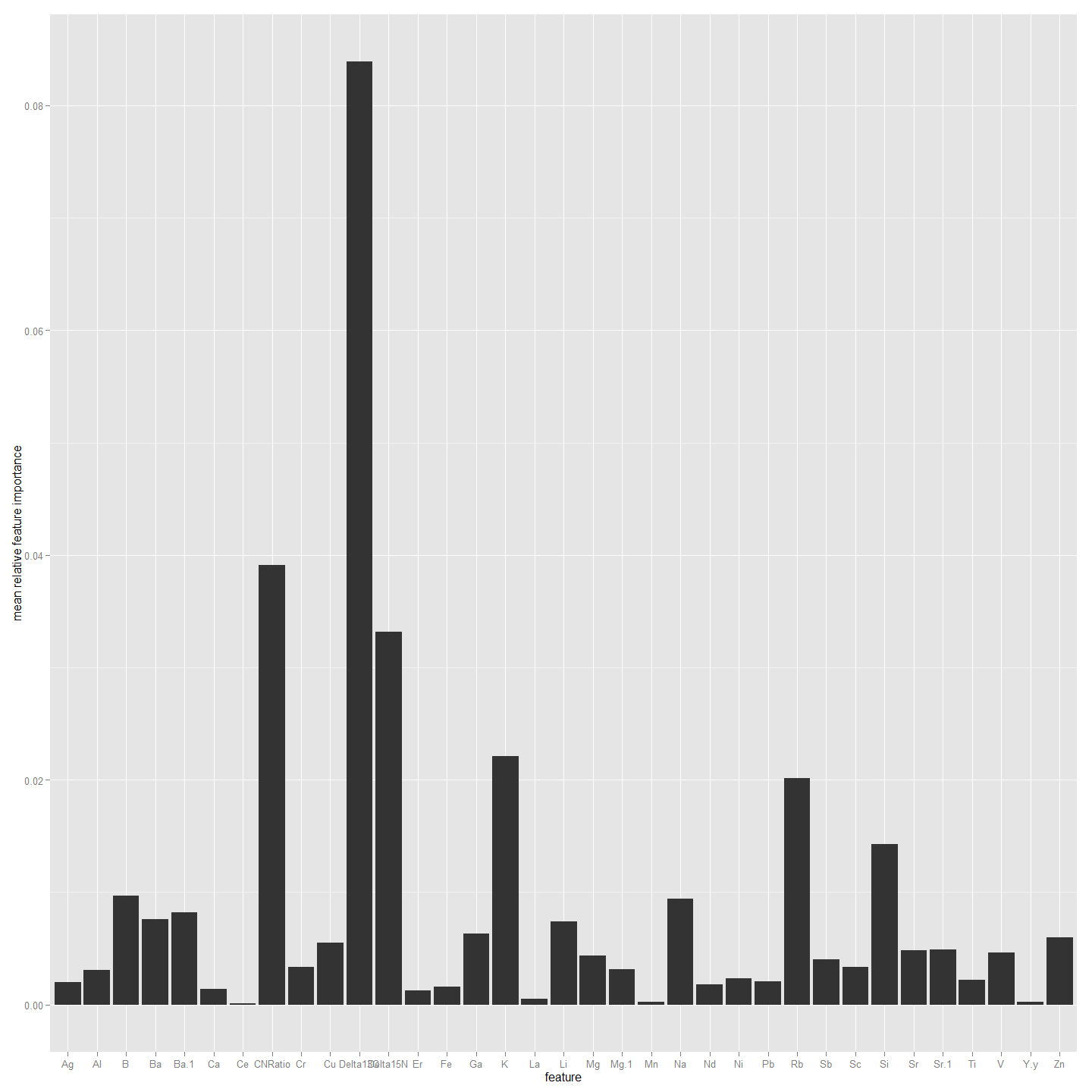
library(ggplot2)  
forest.importance = as.data.frame(importance(forest, scale=FALSE))  
forest.importance = forest.importance[,1:(ncol(forest.importance)-2)]  
forest.importance$mean = rowMeans(forest.importance)  
  
#forest.importance  
  
ggplot(forest.importance, aes(x=row.names(forest.importance), y=mean)) +  
 ylab('mean relative feature importance') +  
 xlab('feature') +  
 geom\_bar(stat='identity')



##################################################################################################  
#Now to do the same but using Non-Core values. So need to combine first  
ImportantVars<-ChemAnalNonCore  
  
ImportantVars<- select(ImportantVars, SiteName,Delta13C,Delta15N,CNRatio,Li:Pb)#ImportantVars<-ImportantVars[c(97,13,14,15,120:151)] #  
ImportantVars<-ImportantVars[complete.cases(ImportantVars),] #remove any nulls  
colnames(ImportantVars)[1]<-"SiteName"  
ImportantVars$SiteName<-as.factor(ImportantVars$SiteName)  
  
ImportantVars <- droplevels(ImportantVars)#Not sure why get error without this line.  
  
library(randomForest)  
forest <- randomForest(SiteName ~.,data=ImportantVars, importance=TRUE)  
  
forest

##   
## Call:  
## randomForest(formula = SiteName ~ ., data = ImportantVars, importance = TRUE)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 5  
##   
## OOB estimate of error rate: 16.59%  
## Confusion matrix:  
## Bullen Range Kambah Pool Lanyon Nerreman Tharwa Sandwash class.error  
## Bullen Range 10 10 0 2 0 0.545454545  
## Kambah Pool 1 121 0 0 0 0.008196721  
## Lanyon 0 8 0 2 0 1.000000000  
## Nerreman 0 6 0 50 0 0.107142857  
## Tharwa Sandwash 0 5 0 2 0 1.000000000

library(ggplot2)  
forest.importance = as.data.frame(importance(forest, scale=FALSE))  
forest.importance = forest.importance[,1:(ncol(forest.importance)-2)]  
forest.importance$mean = rowMeans(forest.importance)  
  
#forest.importance  
  
ggplot(forest.importance, aes(x=row.names(forest.importance), y=mean)) +  
 ylab('mean relative feature importance') +  
 xlab('feature') +  
 geom\_bar(stat='identity')



The highest accuracy prediction (15.4% error rate) is the one using all four otolith zones. This may be due to variable age of larvae, some local effect, exogenous feeding etc. as well as the femals migreation and barriers etc. Given that each part of the otolith can provide an out-of-bag classification rate better than 75% it suggests that otolith microchemistry with delta 13 C and CN ratio, over this small spatial scale, is sufficiently consistently variable, at least within a year, to make a reasonable predictor of site of origin for the Larvae. It is worth noting that the otolith core provided the most accurate classification. Given the accuracy of the core chemistry is sub-optimal because of the inability of the operator to accuratel identify and sample the centre of the otolith in all cases with laser ablation the possibilty remains that cleaning the data in that dataset might be improved by deleting inaccurate ablates. This is possible because each otolith was assigned an accuracy score.

### A Chemotype Dendrogram and a Mantel Test.

Now that we have identified the best variables to use from the data we can create a dendrogram and use Mantel test to look for any relationship between chemotype and geographic distances. This first creates distance matrices for chemotype and geographic distance and then uses a mantel test to see correlation. The latter chemotypes are scaled.

### What about LDA and Jacknifed Prediction

# Linear Discriminant Analysis with Jacknifed Prediction   
library(MASS)  
fit <- lda(SiteName ~ Delta13C + CNRatio+B+K+V+Rb, data=ImportantVars, na.action="na.omit", CV=TRUE)  
#fit # show results  
# Assess the accuracy of the prediction  
# percent correct for each category of G  
ct <- table(ImportantVars$SiteName, fit$class)  
diag(prop.table(ct, 1))

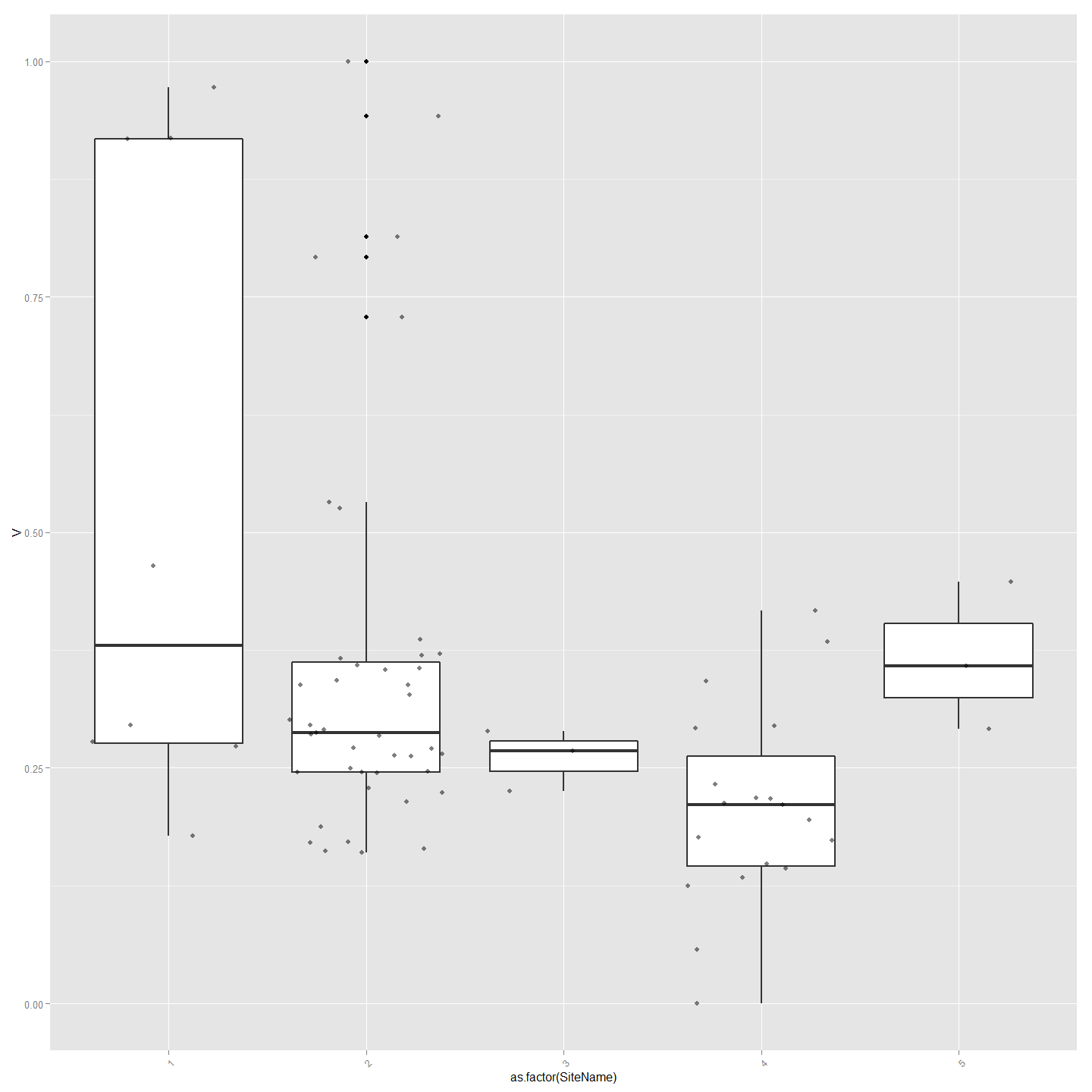
## Bullen Range Kambah Pool Lanyon Nerreman Tharwa Sandwash   
## 0.5000000 0.9016393 0.0000000 0.7678571 0.0000000

# total percent correct  
sum(diag(prop.table(ct)))

## [1] 0.7557604

### What about Ordinal Logistic Regression

require(dplyr)  
  
ImportantVars<-ChemAnalCore  
ImportantVars<- dplyr::select(ImportantVars, SiteName,Delta13C,Delta15N,CNRatio,Li:Pb)#ImportantVars<-ImportantVars[c(84,13,14,15,121,122,127,131,139)] Not all are in the new bit....  
ImportantVars<-ImportantVars[complete.cases(ImportantVars),] #remove any nulls  
ImportantVars <- droplevels(ImportantVars)#Not sure why get error without this line  
require(clusterSim)  
# Normalise the Data  
ImpVarScaled<-data.Normalization (ImportantVars,type="n4",normalization="column")  
#Check it looks OK  
#print(ImpVarScaled)[10:20,]  
  
ggplot(ImpVarScaled, aes(x = as.factor(SiteName), y = V)) +  
 geom\_boxplot(size = .75) +  
 geom\_jitter(alpha = .5) +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1, vjust = 1))



#Ordinal Logistic Regression  
## fit ordered logit model and store results 'model'  
model <- polr(as.factor(SiteName) ~ B + Na + V + K+ Delta13C + CNRatio, data = ImpVarScaled, Hess=TRUE)  
  
## view a summary of the model  
summary(model)

## Call:  
## polr(formula = as.factor(SiteName) ~ B + Na + V + K + Delta13C +   
## CNRatio, data = ImpVarScaled, Hess = TRUE)  
##   
## Coefficients:  
## Value Std. Error t value  
## B -4.526 1.579 -2.8664  
## Na 3.084 1.405 2.1941  
## V -4.141 1.490 -2.7784  
## K -1.669 2.302 -0.7251  
## Delta13C 2.910 1.628 1.7875  
## CNRatio -1.890 2.388 -0.7918  
##   
## Intercepts:  
## Value Std. Error t value  
## 1|2 -3.7051 1.9595 -1.8908  
## 2|3 0.7724 1.7758 0.4349  
## 3|4 1.1004 1.7973 0.6122  
## 4|5 3.9390 1.9404 2.0300  
##   
## Residual Deviance: 136.7695   
## AIC: 156.7695

## store table  
(ctable <- coef(summary(model)))

## Value Std. Error t value  
## B -4.5257972 1.578888 -2.8664458  
## Na 3.0837078 1.405462 2.1940886  
## V -4.1408029 1.490366 -2.7783797  
## K -1.6691848 2.302104 -0.7250692  
## Delta13C 2.9097836 1.627825 1.7875280  
## CNRatio -1.8904731 2.387574 -0.7917967  
## 1|2 -3.7050744 1.959491 -1.8908348  
## 2|3 0.7723822 1.775806 0.4349474  
## 3|4 1.1003916 1.797305 0.6122452  
## 4|5 3.9390455 1.940393 2.0300250

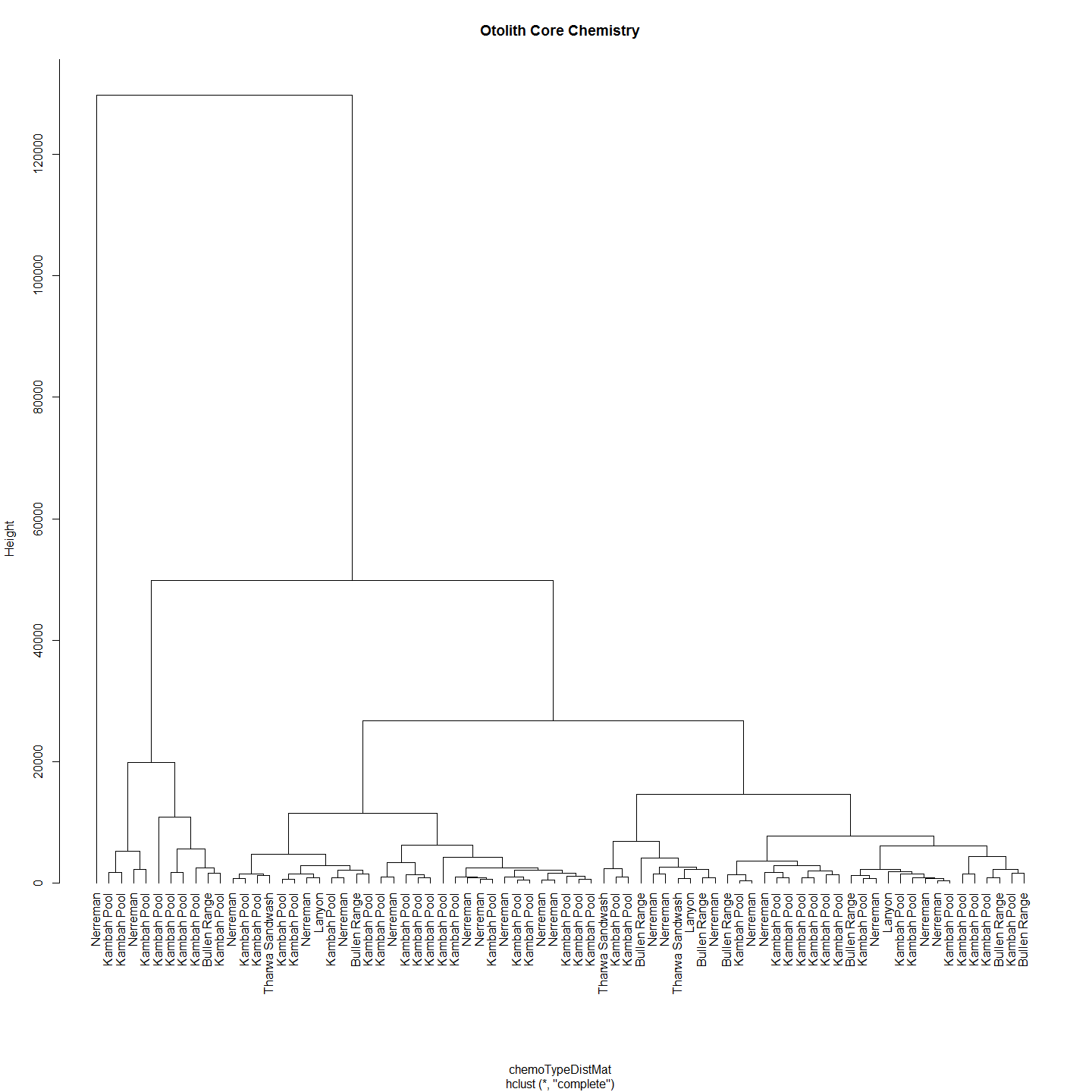
## calculate and store p values  
p <- pnorm(abs(ctable[, "t value"]), lower.tail = FALSE) \* 2  
## combined table  
(ctable <- cbind(ctable, "p value" = p))

## Value Std. Error t value p value  
## B -4.5257972 1.578888 -2.8664458 0.004151092  
## Na 3.0837078 1.405462 2.1940886 0.028229041  
## V -4.1408029 1.490366 -2.7783797 0.005463074  
## K -1.6691848 2.302104 -0.7250692 0.468409554  
## Delta13C 2.9097836 1.627825 1.7875280 0.073852185  
## CNRatio -1.8904731 2.387574 -0.7917967 0.428479199  
## 1|2 -3.7050744 1.959491 -1.8908348 0.058646407  
## 2|3 0.7723822 1.775806 0.4349474 0.663600619  
## 3|4 1.1003916 1.797305 0.6122452 0.540375557  
## 4|5 3.9390455 1.940393 2.0300250 0.042353996

#or just do confidence intervals  
ci<-confint(model)

### Calculate Chemotype and Distance Correlations for Otolith Parts

library(clusterSim)  
require(dplyr)  
#This creates distance matrices for chemotype and geographic distance and then use a mantel test to see correlation  
#First using Otolith Core chemistry.  
  
allVars<-ChemAnalCore  
  
allVars<-dplyr::select(allVars, Label, SiteName,Delta13C,Delta15N,CNRatio,Li:Pb)#[c(2,84,13,14,15,120:151)] #allVars<-allVars[c(2,84,13,14,15,120:151)]  
allVars<-allVars[complete.cases(allVars),] #remove any nulls  
row.names(allVars)<-allVars[,1]  
allVars$Label<-NULL  
colnames(allVars)[1]<-"SiteName"  
  
allVars$SiteName<-as.factor(allVars$SiteName)  
  
df<-allVars  
chemoTypeDistMat <- dist(df)  
hClusters <- hclust(chemoTypeDistMat)  
plot(hClusters,labels=(df$SiteName), hang = -1, main="Otolith Core Chemistry")



#Make distance matrices for geographic distance as well  
allVars<-ChemAnalCore  
allVars<-dplyr::select(allVars, Label, SiteName,Delta13C,Delta15N,CNRatio,Li:Pb)#allVars<-allVars[c(2,84,13,14,15,107,120:151)]  
allVars<-allVars[complete.cases(allVars),] #remove any nulls  
allVars<-allVars[c(1,6)] # Distance from Angle Crossing changes from 107 above to 6  
row.names(allVars)<-allVars[,1]  
allVars$Label<-NULL  
  
geoDist<-allVars  
geoDist<-na.omit(geoDist)  
#geoDistColl1000<-geodist #save this estimate for haplogroups distance plot (after the Iterated Mantel has changed it)  
  
geoDistMat<-dist(geoDist)  
  
  
#make sure both matrices are in correct order - rows and cols  
#Check all is in order  
as.matrix(geoDistMat)[1:5, 1:5] # zero distances in the first 5

## K2013117-185 K2013117-186 K2013117-187 K2013117-188 K2013117-189  
## K2013117-185 0.0000000000 0.0153280664 0.0147953405 0.012204794 0.0005952875  
## K2013117-186 0.0153280664 0.0000000000 0.0005327259 0.003123273 0.0159233539  
## K2013117-187 0.0147953405 0.0005327259 0.0000000000 0.002590547 0.0153906280  
## K2013117-188 0.0122047936 0.0031232728 0.0025905469 0.000000000 0.0128000811  
## K2013117-189 0.0005952875 0.0159233539 0.0153906280 0.012800081 0.0000000000

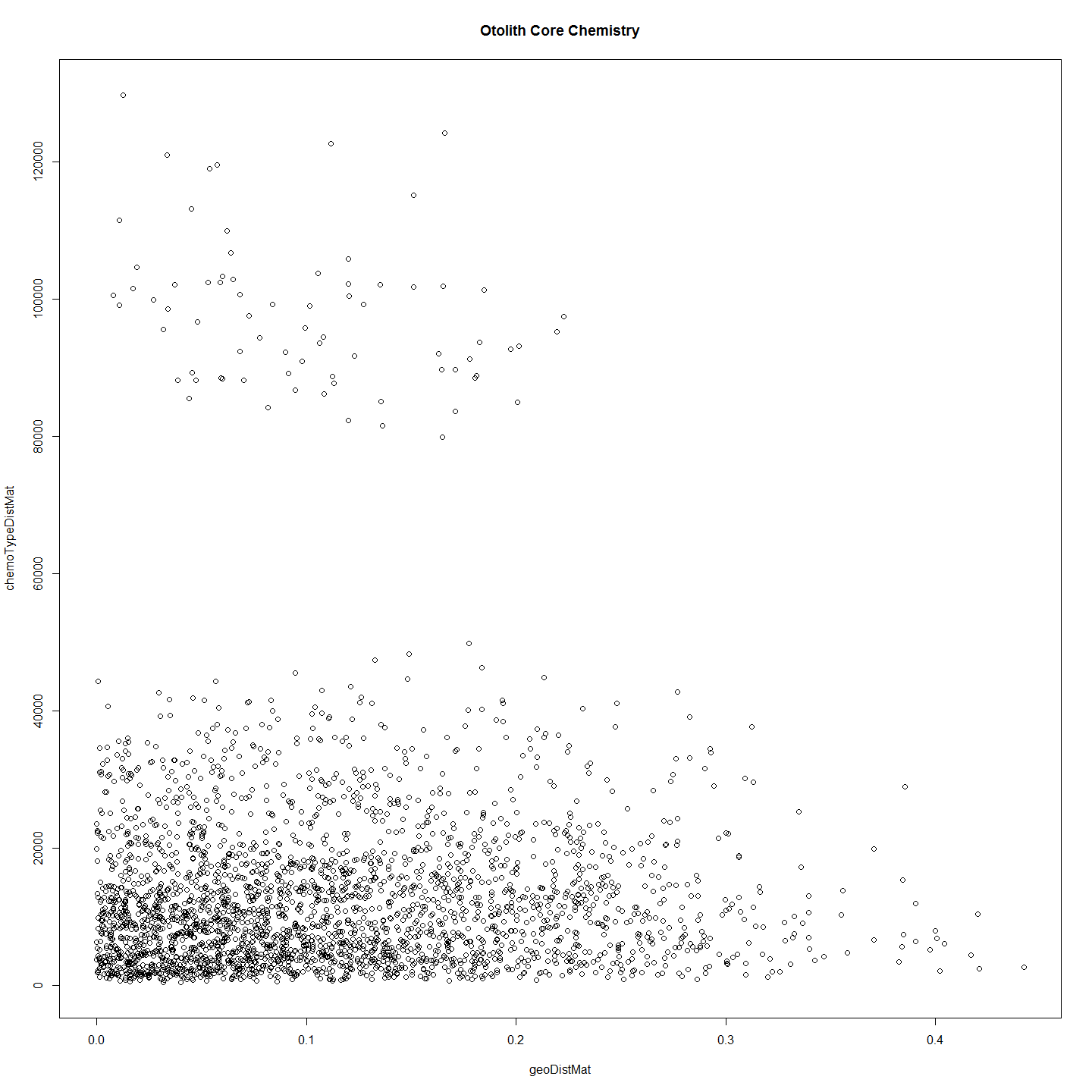
as.matrix(chemoTypeDistMat)[1:5, 1:5]

## K2013117-185 K2013117-186 K2013117-187 K2013117-188 K2013117-189  
## K2013117-185 0.00 35447.596 35951.091 33050.323 22444.82  
## K2013117-186 35447.60 0.000 1936.981 2662.856 13139.70  
## K2013117-187 35951.09 1936.981 0.000 3006.428 13514.08  
## K2013117-188 33050.32 2662.856 3006.428 0.000 10657.86  
## K2013117-189 22444.82 13139.698 13514.079 10657.864 0.00

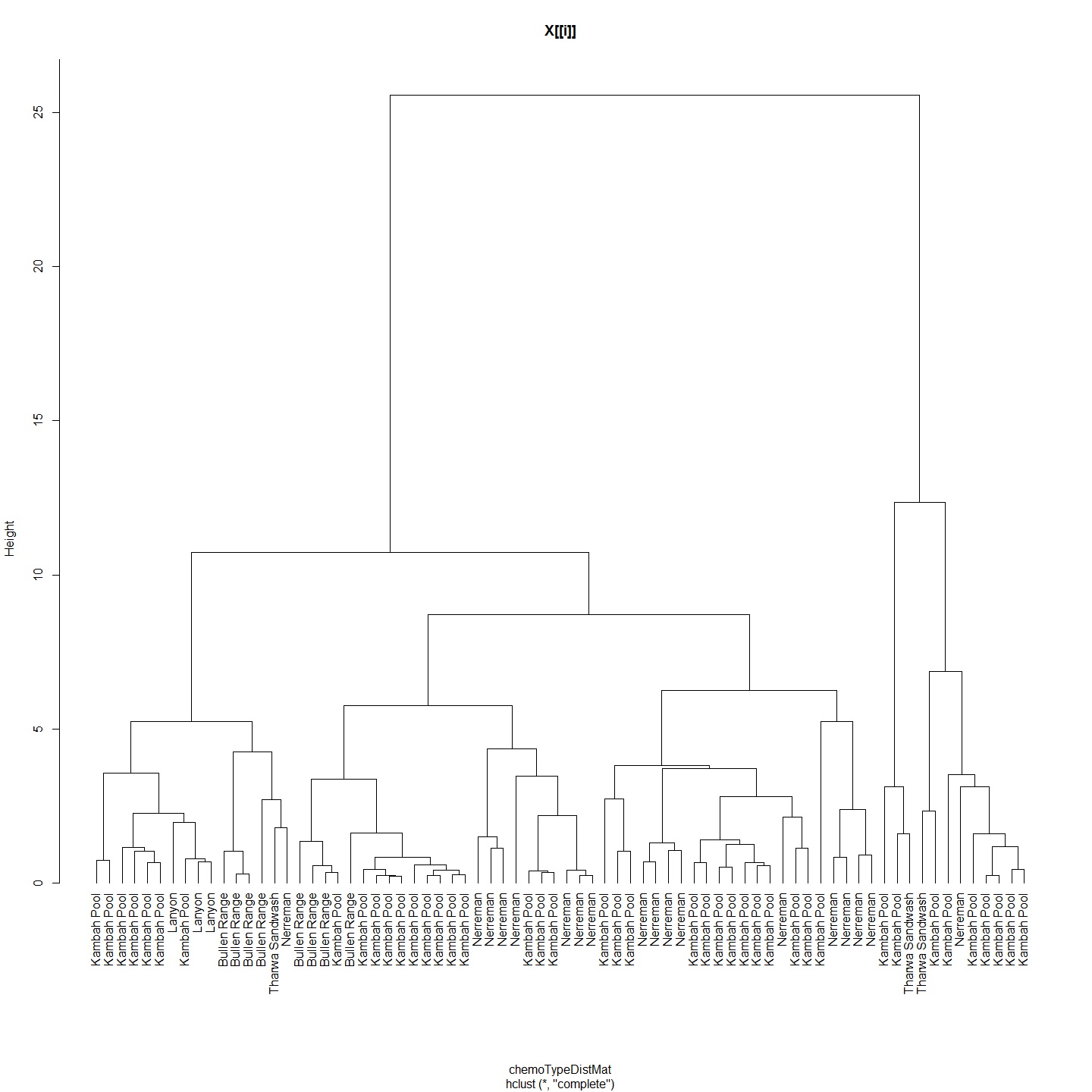
#Conduct Mantel Test on Matrices  
mant<-mantel.rtest(as.dist(geoDistMat), as.dist(chemoTypeDistMat), nrepet = 9999)  
mant

## Monte-Carlo test  
## Observation: -0.004127413   
## Call: mantel.rtest(m1 = as.dist(geoDistMat), m2 = as.dist(chemoTypeDistMat),   
## nrepet = 9999)  
## Based on 9999 replicates  
## Simulated p-value: 0.4494

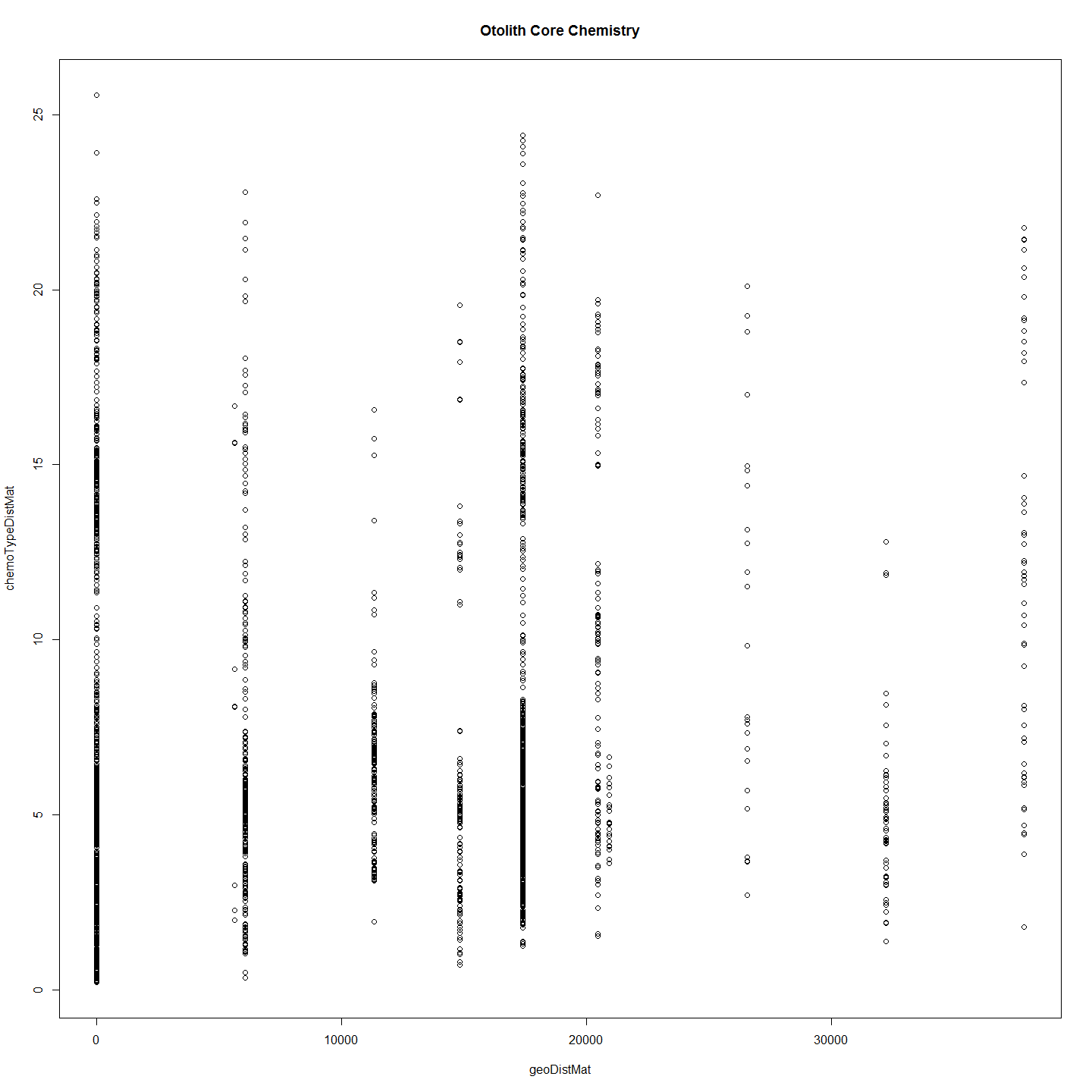
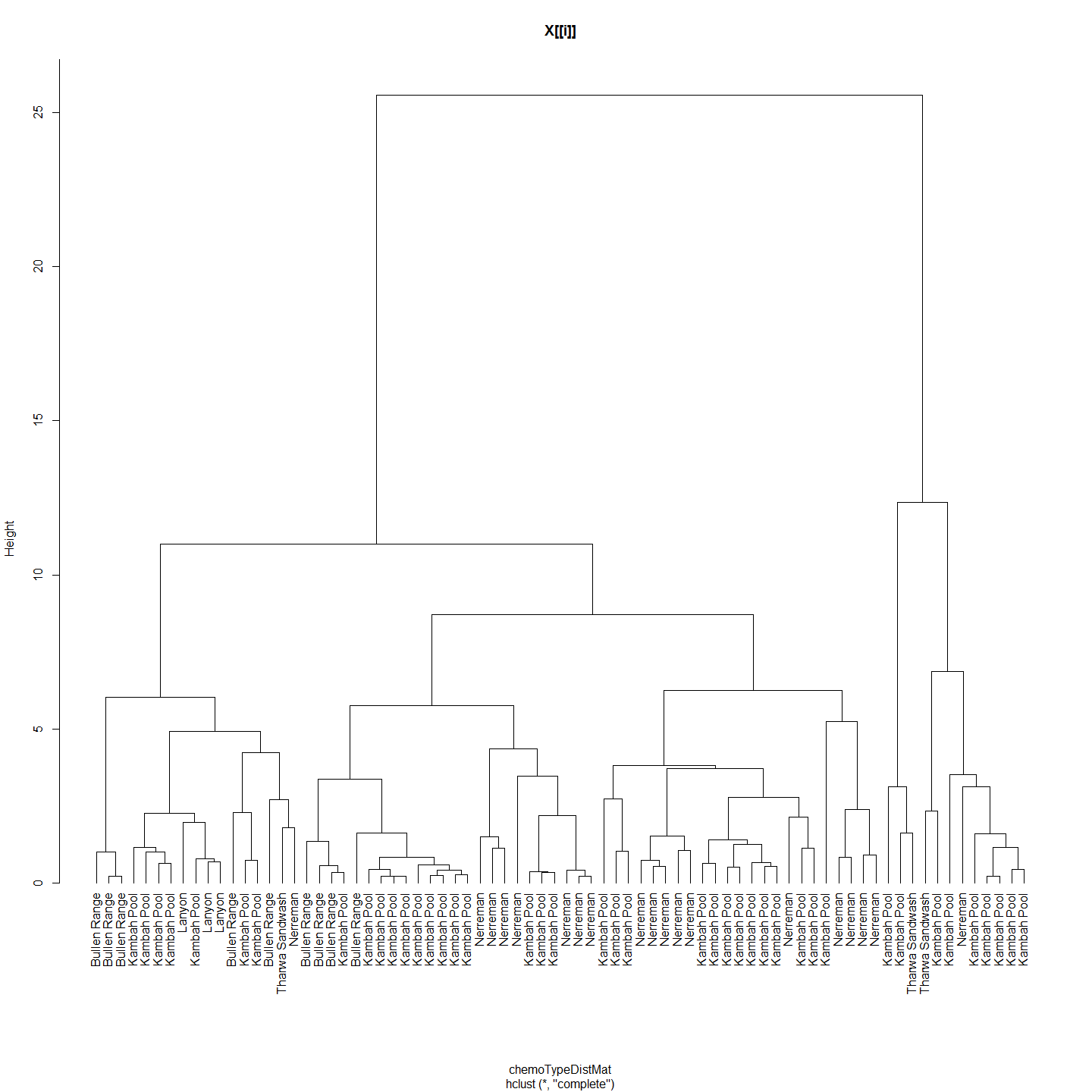
plot(geoDistMat,chemoTypeDistMat, main="Otolith Core Chemistry")



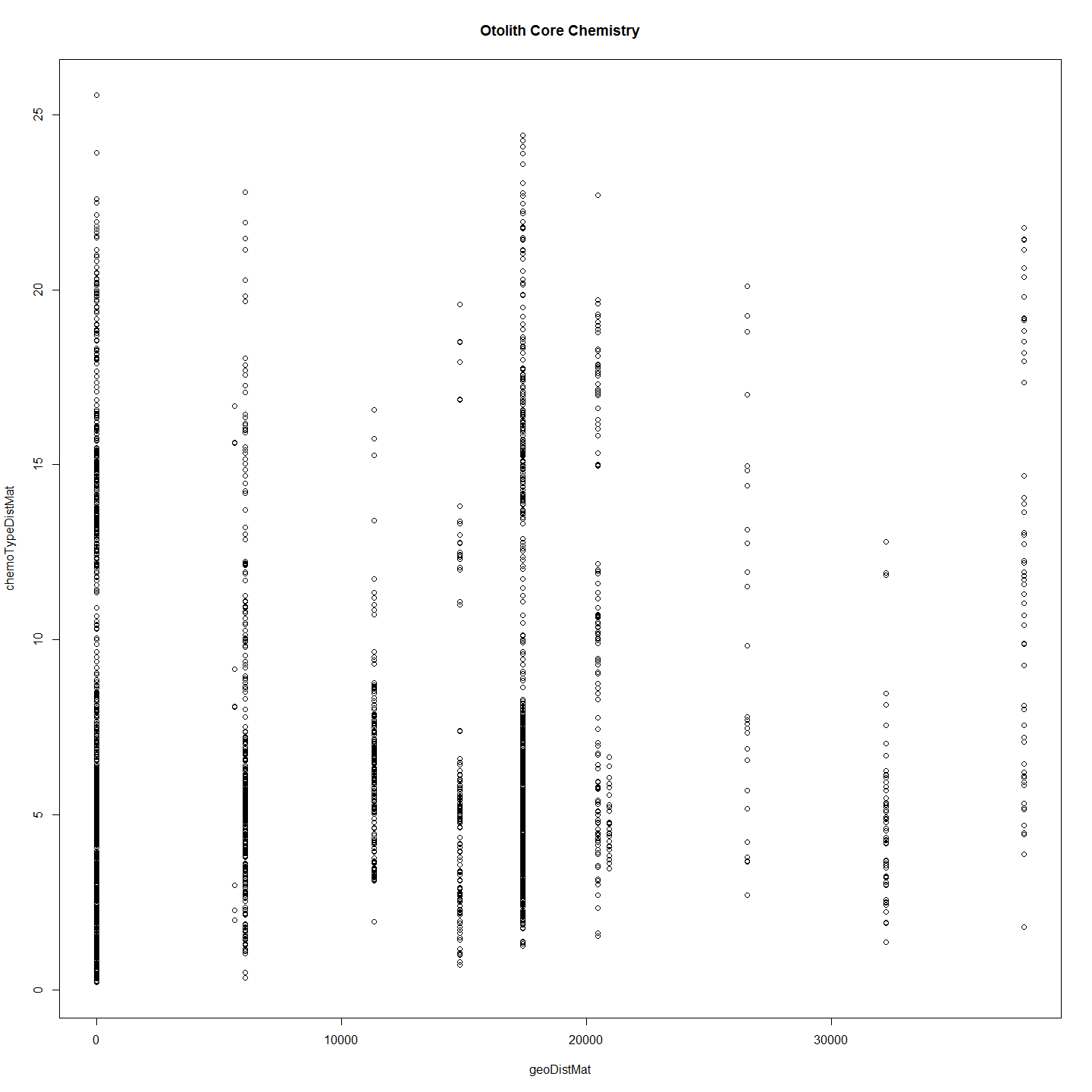
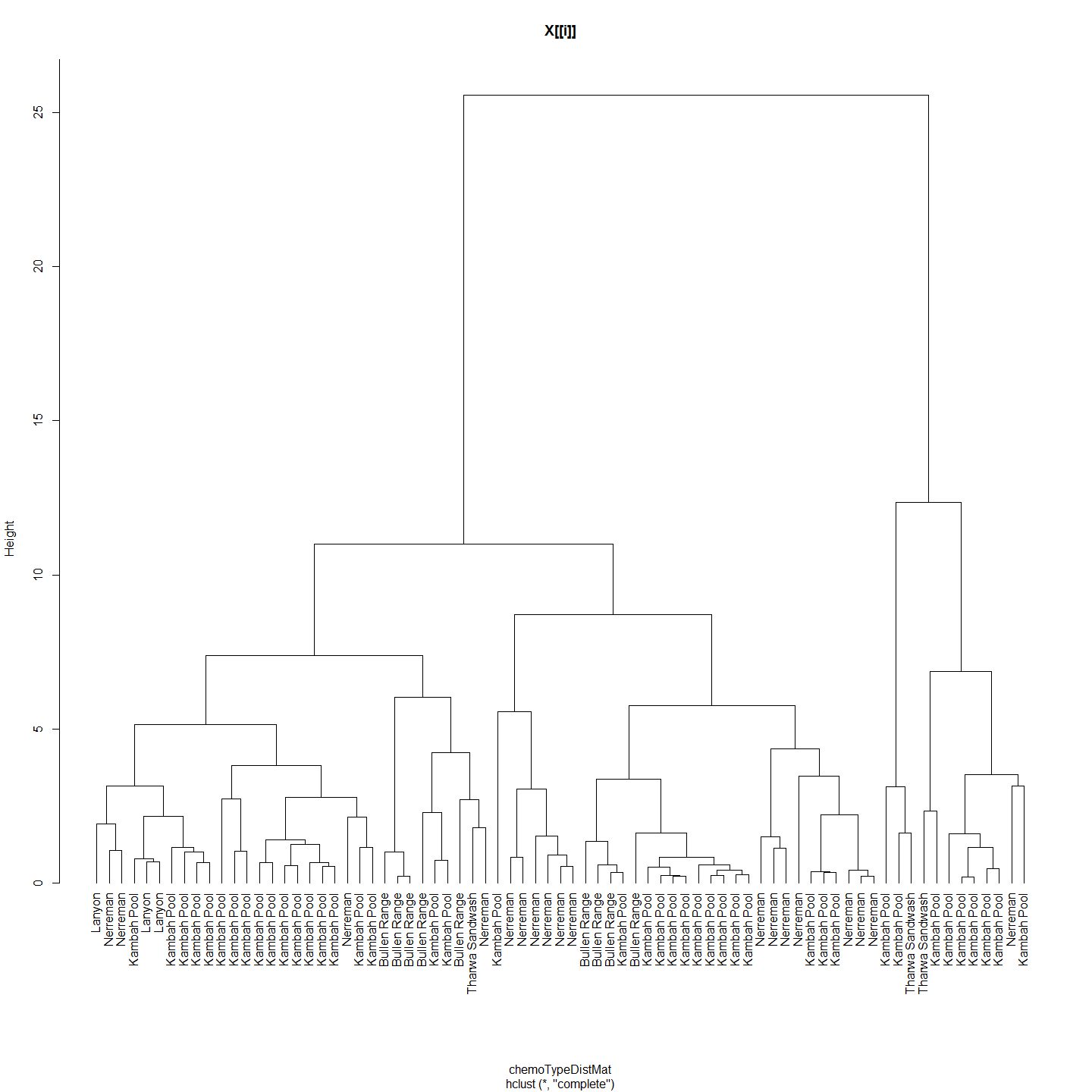
#Even if the data is scaled there is no correlation to speak of.  
#########################################  
#allVars$SiteName<-NULL  
#allVars <- droplevels(allVars)#Not sure why get error without this line  
#allVarScaled<-data.Normalization (allVars,type="n7",normalization="column")  
# allVarScaled$Label<-row.names(allVarScaled)  
# sChemAnalCore<-subset(ChemAnalCore, select=c(Label,SiteName))  
# allVars<-merge(allVarScaled, sChemAnalCore,by = "Label")  
# allVars$Label<-NULL  
# df<-allVarScaled  
# distxy <- dist(df)  
# hClusters <- hclust(distxy)  
# plot(hClusters,labels=(df$Site.Name), hang = -1)  
# df<-allVarScaled  
###################################  
  
#But now with important variables only. These include Delta13C, Delta15N, CN ratio, B, K, V, Na, Rb and were settled on by using Random Forest.  
  
otoParts<-list(ChemAnalCore,ChemAnalE1,ChemAnalE2,ChemAnal0,ChemAnalAll4,ChemAnalNonCore)  
lapply(otoParts, otoPartChem)



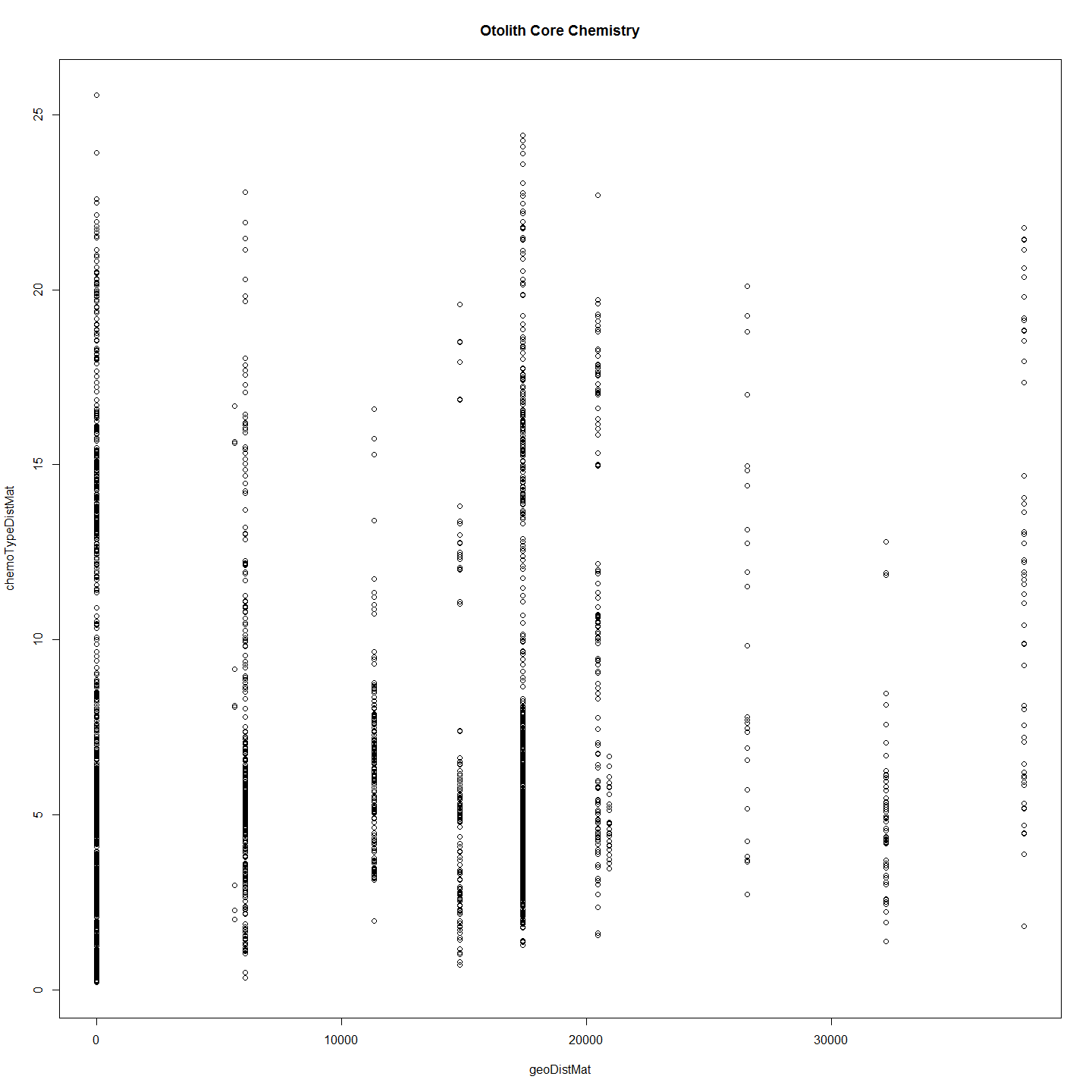
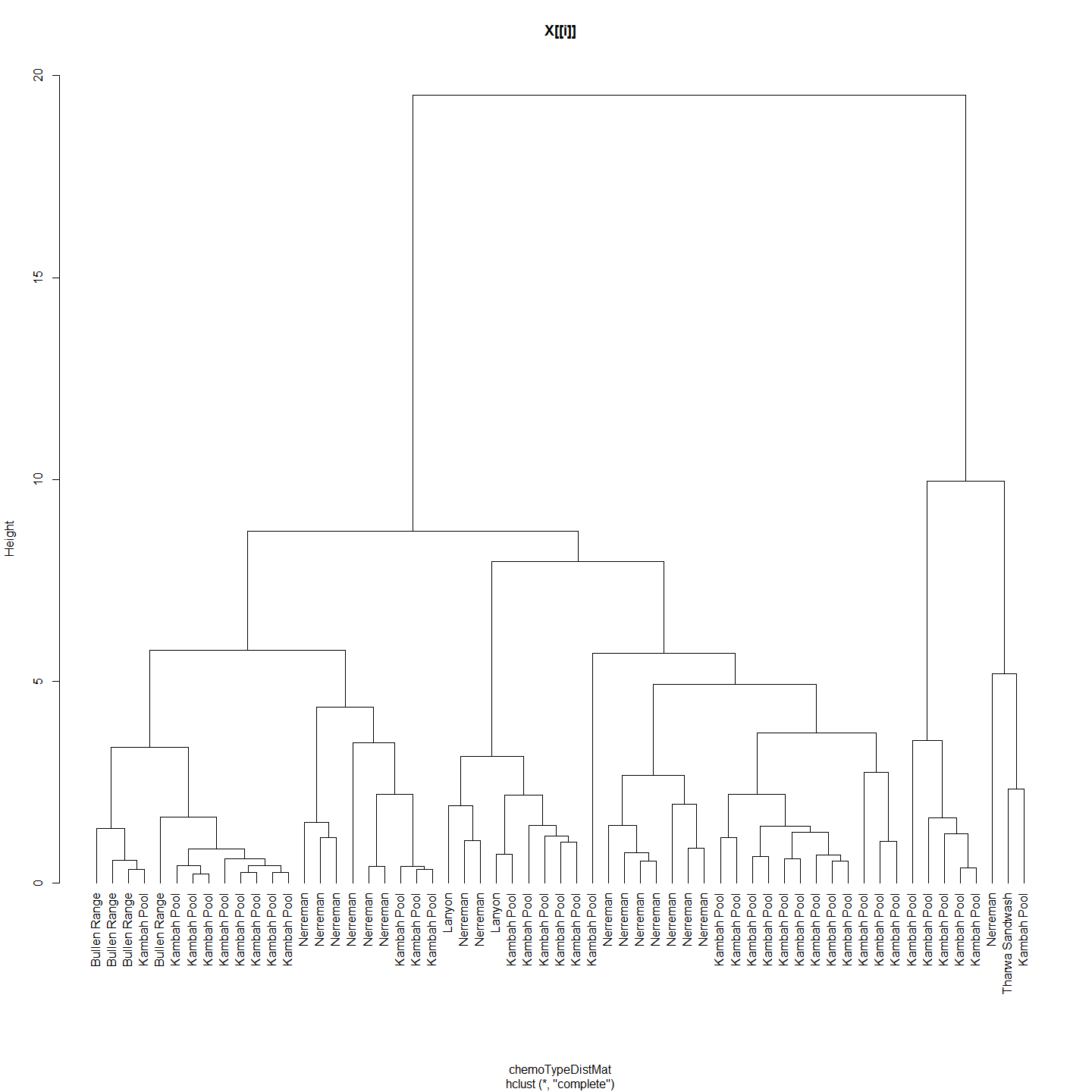
## 1 2 3 4 5  
## 1 0 0 0 0 0  
## 2 0 0 0 0 0  
## 3 0 0 0 0 0  
## 4 0 0 0 0 0  
## 5 0 0 0 0 0  
## 1 2 3 4 5  
## 1 0.0000000 0.4466052 0.7249025 0.6552282 0.6952819  
## 2 0.4466052 0.0000000 0.4320683 0.6014812 0.3556002  
## 3 0.7249025 0.4320683 0.0000000 0.3608581 0.2594037  
## 4 0.6552282 0.6014812 0.3608581 0.0000000 0.5500707  
## 5 0.6952819 0.3556002 0.2594037 0.5500707 0.0000000  
## Monte-Carlo test  
## Observation: 0.07902223   
## Call: mantel.rtest(m1 = as.dist(geoDistMat), m2 = as.dist(chemoTypeDistMat),   
## nrepet = 9999)  
## Based on 9999 replicates  
## Simulated p-value: 0.1207

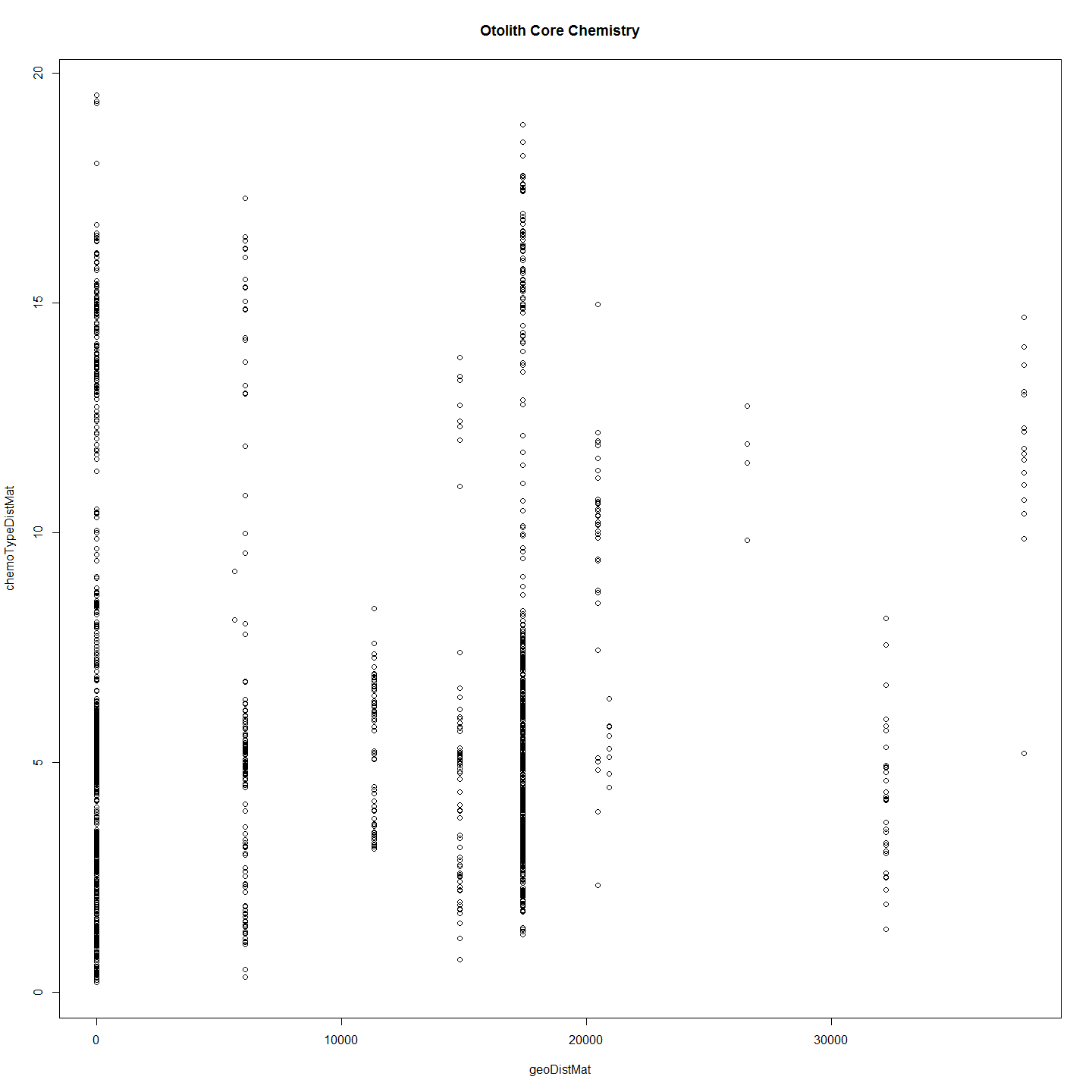
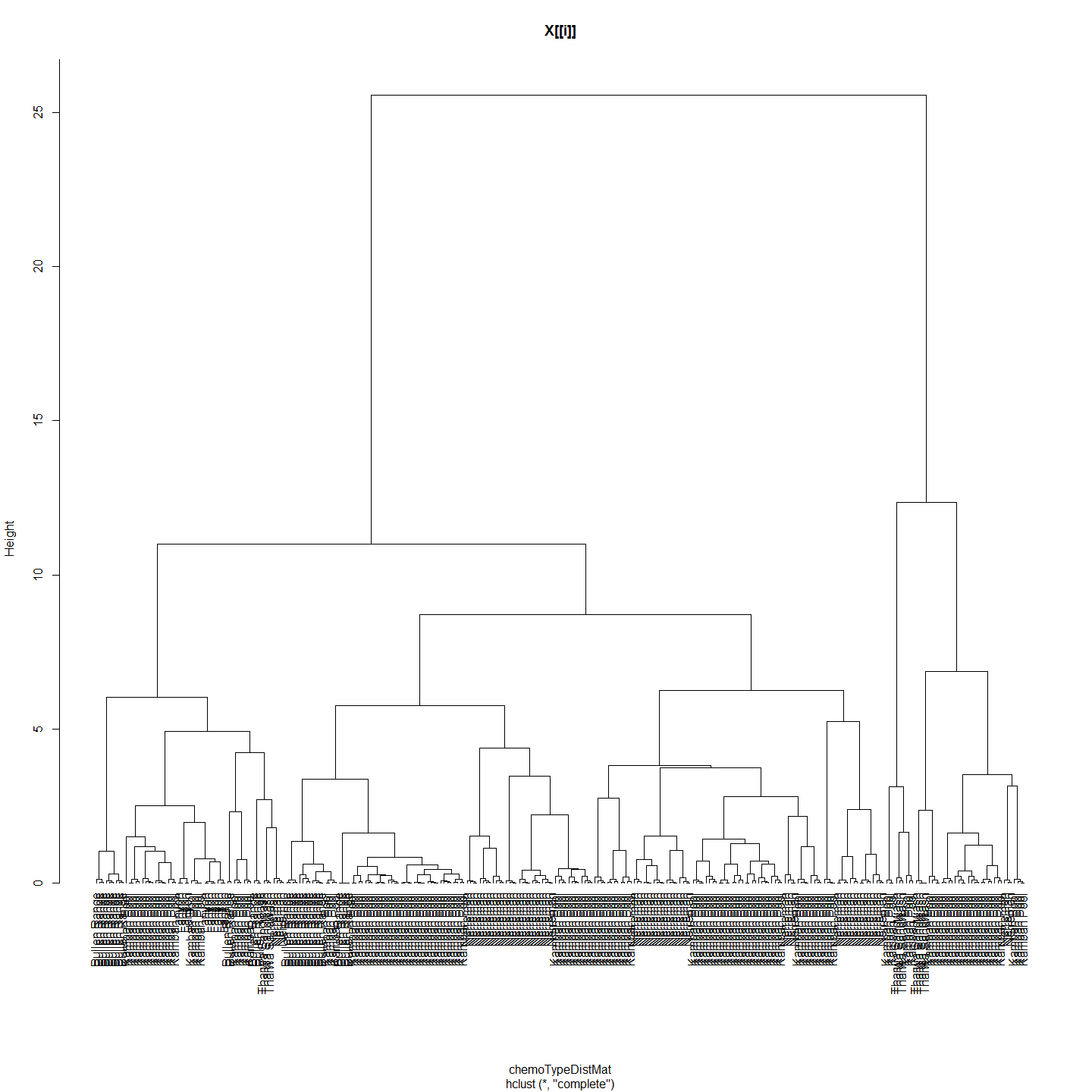
## 1 2 3 4 5  
## 1 0 0 0 0 0  
## 2 0 0 0 0 0  
## 3 0 0 0 0 0  
## 4 0 0 0 0 0  
## 5 0 0 0 0 0  
## 1 2 3 4 5  
## 1 0.0000000 0.4462723 0.7247957 0.6551568 0.6964957  
## 2 0.4462723 0.0000000 0.4321628 0.6015201 0.3572664  
## 3 0.7247957 0.4321628 0.0000000 0.3608487 0.2605318  
## 4 0.6551568 0.6015201 0.3608487 0.0000000 0.5507640  
## 5 0.6964957 0.3572664 0.2605318 0.5507640 0.0000000  
## Monte-Carlo test  
## Observation: 0.08057341   
## Call: mantel.rtest(m1 = as.dist(geoDistMat), m2 = as.dist(chemoTypeDistMat),   
## nrepet = 9999)  
## Based on 9999 replicates  
## Simulated p-value: 0.1174

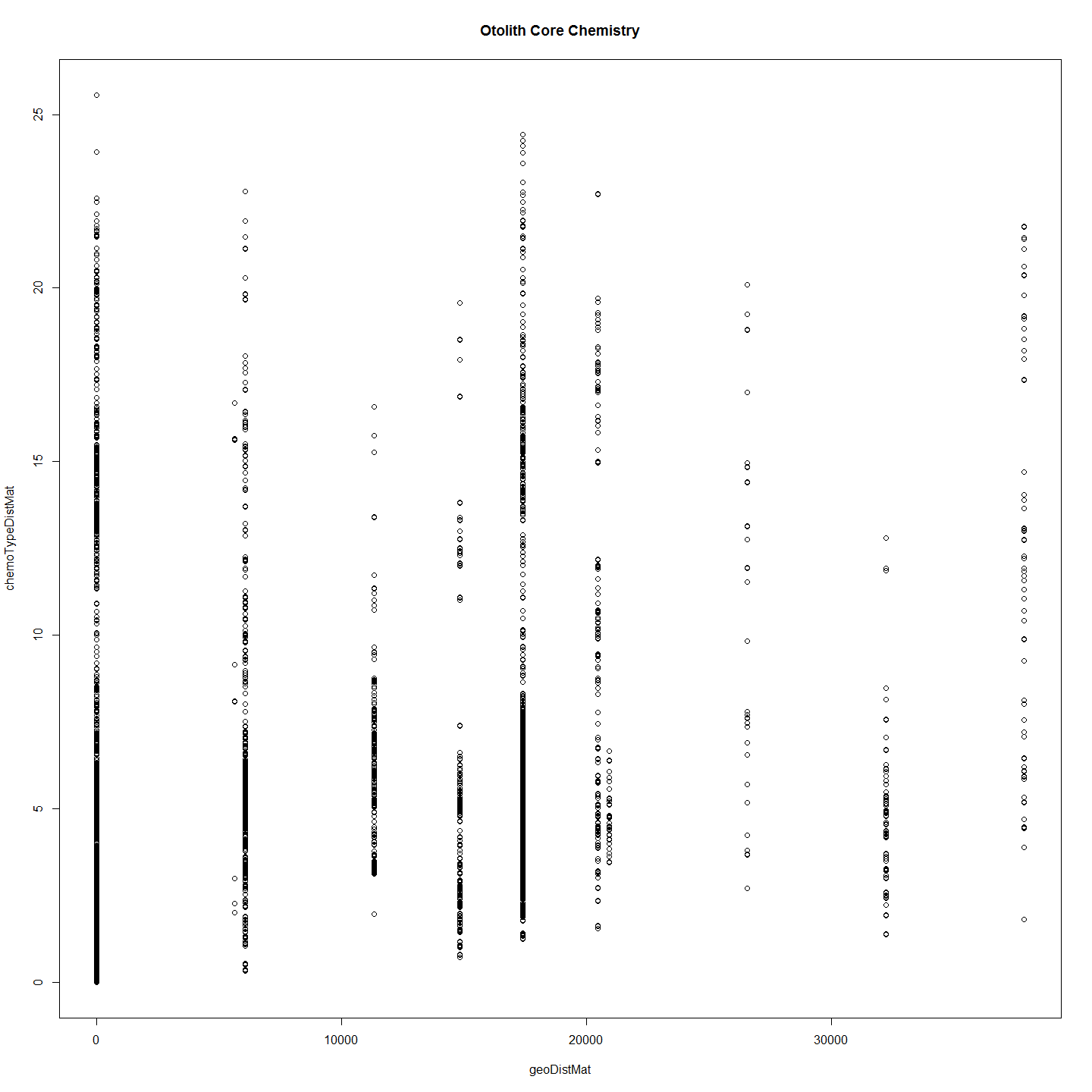
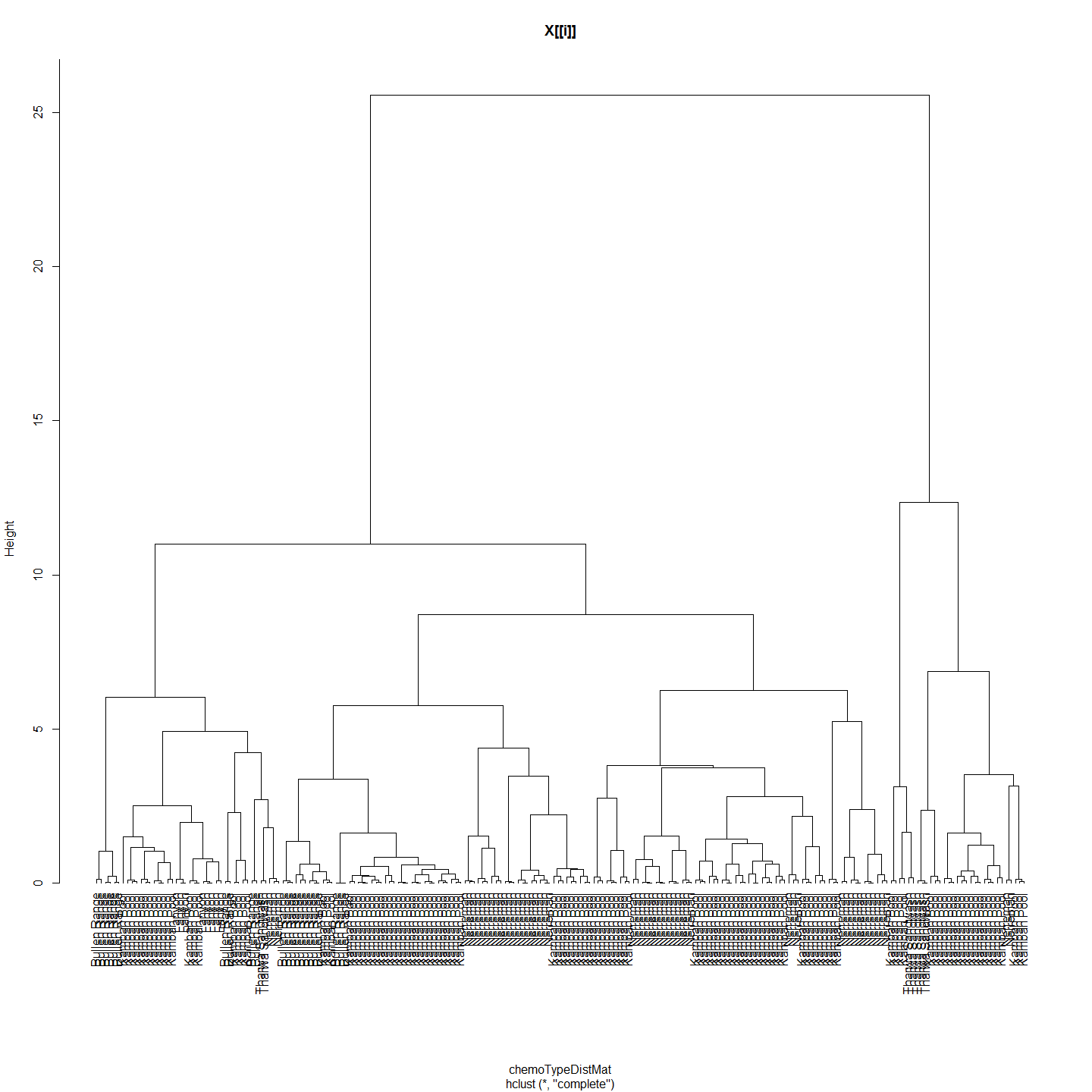
## 1 2 3 4 5  
## 1 0 0 0 0 0  
## 2 0 0 0 0 0  
## 3 0 0 0 0 0  
## 4 0 0 0 0 0  
## 5 0 0 0 0 0  
## 1 2 3 4 5  
## 1 0.0000000 0.4465242 0.7257491 0.6603559 0.6972760  
## 2 0.4465242 0.0000000 0.4327172 0.6053176 0.3571205  
## 3 0.7257491 0.4327172 0.0000000 0.3635721 0.2591861  
## 4 0.6603559 0.6053176 0.3635721 0.0000000 0.5507294  
## 5 0.6972760 0.3571205 0.2591861 0.5507294 0.0000000  
## Monte-Carlo test  
## Observation: 0.07983363   
## Call: mantel.rtest(m1 = as.dist(geoDistMat), m2 = as.dist(chemoTypeDistMat),   
## nrepet = 9999)  
## Based on 9999 replicates  
## Simulated p-value: 0.1251

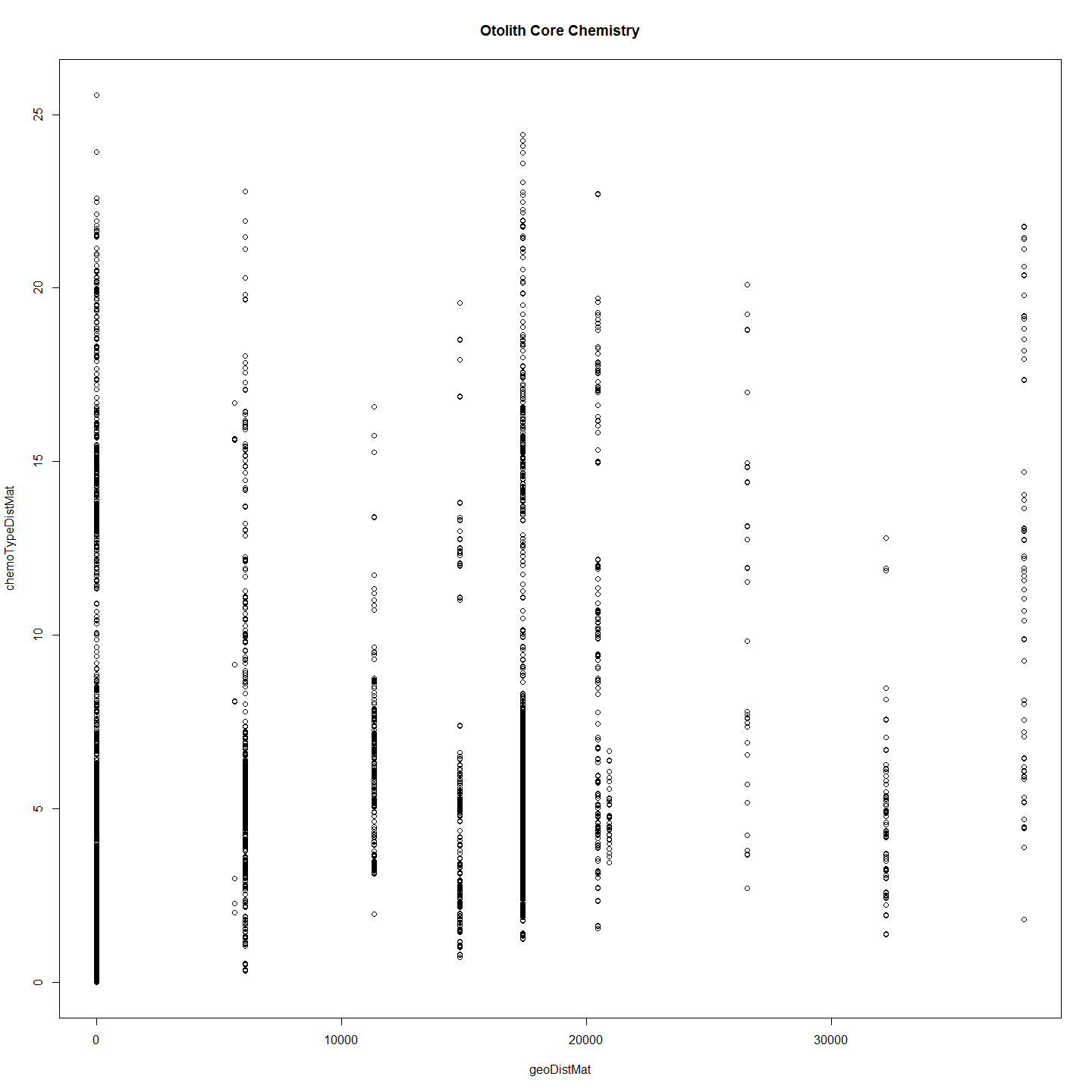
## 1 2 3 4 5  
## 1 0 0 0 0 0  
## 2 0 0 0 0 0  
## 3 0 0 0 0 0  
## 4 0 0 0 0 0  
## 5 0 0 0 0 0  
## 1 2 3 4 5  
## 1 0.0000000 0.4467655 0.7248863 0.6550822 0.6956475  
## 2 0.4467655 0.0000000 0.4336633 0.6018223 0.3578156  
## 3 0.7248863 0.4336633 0.0000000 0.3612277 0.2588976  
## 4 0.6550822 0.6018223 0.3612277 0.0000000 0.5503641  
## 5 0.6956475 0.3578156 0.2588976 0.5503641 0.0000000  
## Monte-Carlo test  
## Observation: 0.05066918   
## Call: mantel.rtest(m1 = as.dist(geoDistMat), m2 = as.dist(chemoTypeDistMat),   
## nrepet = 9999)  
## Based on 9999 replicates  
## Simulated p-value: 0.2347

## 1 2 3 4 5  
## 1 0 0 0 0 0  
## 2 0 0 0 0 0  
## 3 0 0 0 0 0  
## 4 0 0 0 0 0  
## 5 0 0 0 0 0  
## 1 2 3 4 5  
## 1 0.0000000 0.4467655 0.7248863 0.6550822 0.6956475  
## 2 0.4467655 0.0000000 0.4336633 0.6018223 0.3578156  
## 3 0.7248863 0.4336633 0.0000000 0.3612277 0.2588976  
## 4 0.6550822 0.6018223 0.3612277 0.0000000 0.5503641  
## 5 0.6956475 0.3578156 0.2588976 0.5503641 0.0000000  
## Monte-Carlo test  
## Observation: 0.08987449   
## Call: mantel.rtest(m1 = as.dist(geoDistMat), m2 = as.dist(chemoTypeDistMat),   
## nrepet = 9999)  
## Based on 9999 replicates  
## Simulated p-value: 0.006

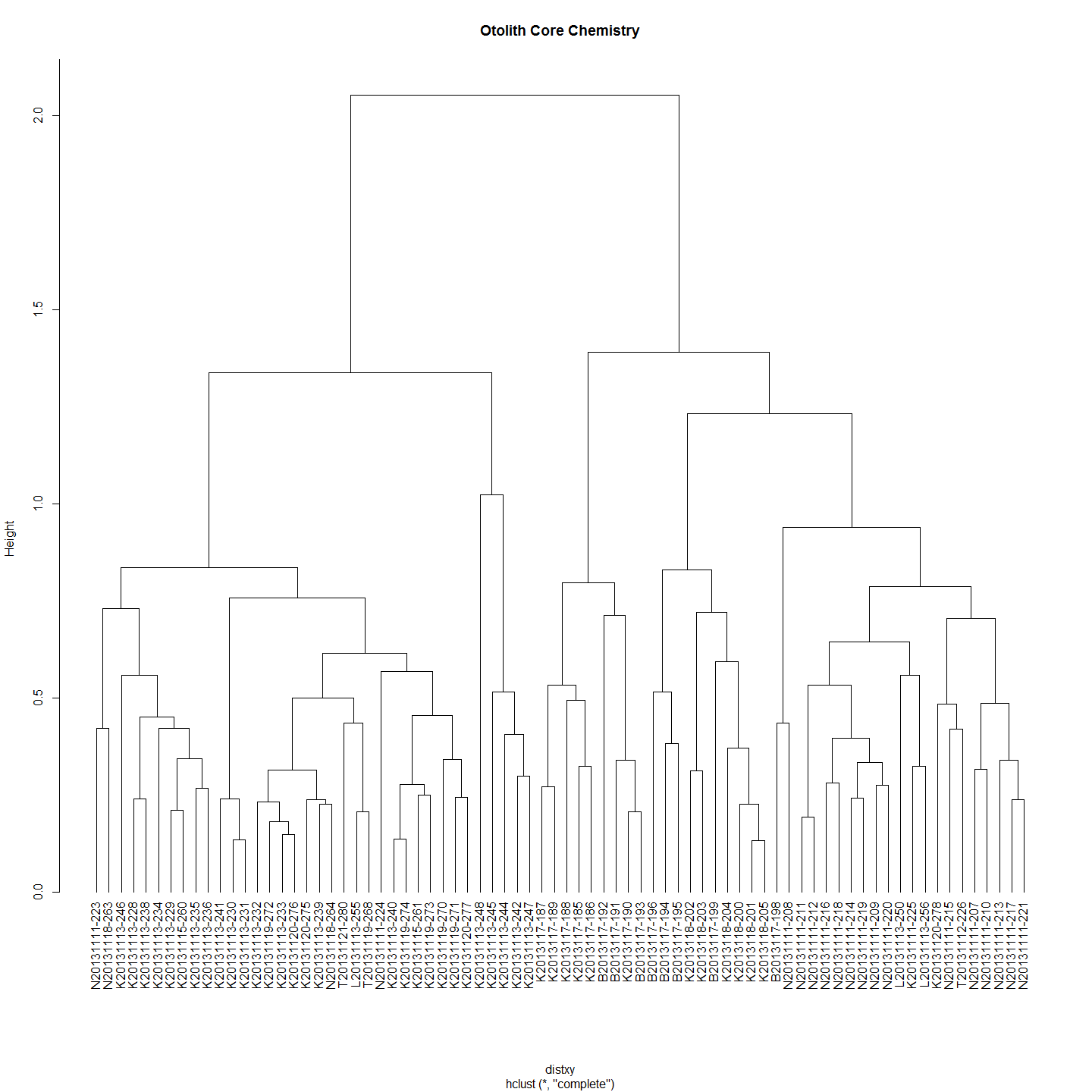
 

## 1 2 3 4 5  
## 1 0 0 0 0 0  
## 2 0 0 0 0 0  
## 3 0 0 0 0 0  
## 4 0 0 0 0 0  
## 5 0 0 0 0 0  
## 1 2 3 4 5  
## 1 0.0000000 0.4467655 0.7248863 0.6550822 0.6956475  
## 2 0.4467655 0.0000000 0.4336633 0.6018223 0.3578156  
## 3 0.7248863 0.4336633 0.0000000 0.3612277 0.2588976  
## 4 0.6550822 0.6018223 0.3612277 0.0000000 0.5503641  
## 5 0.6956475 0.3578156 0.2588976 0.5503641 0.0000000  
## Monte-Carlo test  
## Observation: 0.08732716   
## Call: mantel.rtest(m1 = as.dist(geoDistMat), m2 = as.dist(chemoTypeDistMat),   
## nrepet = 9999)  
## Based on 9999 replicates  
## Simulated p-value: 0.0167

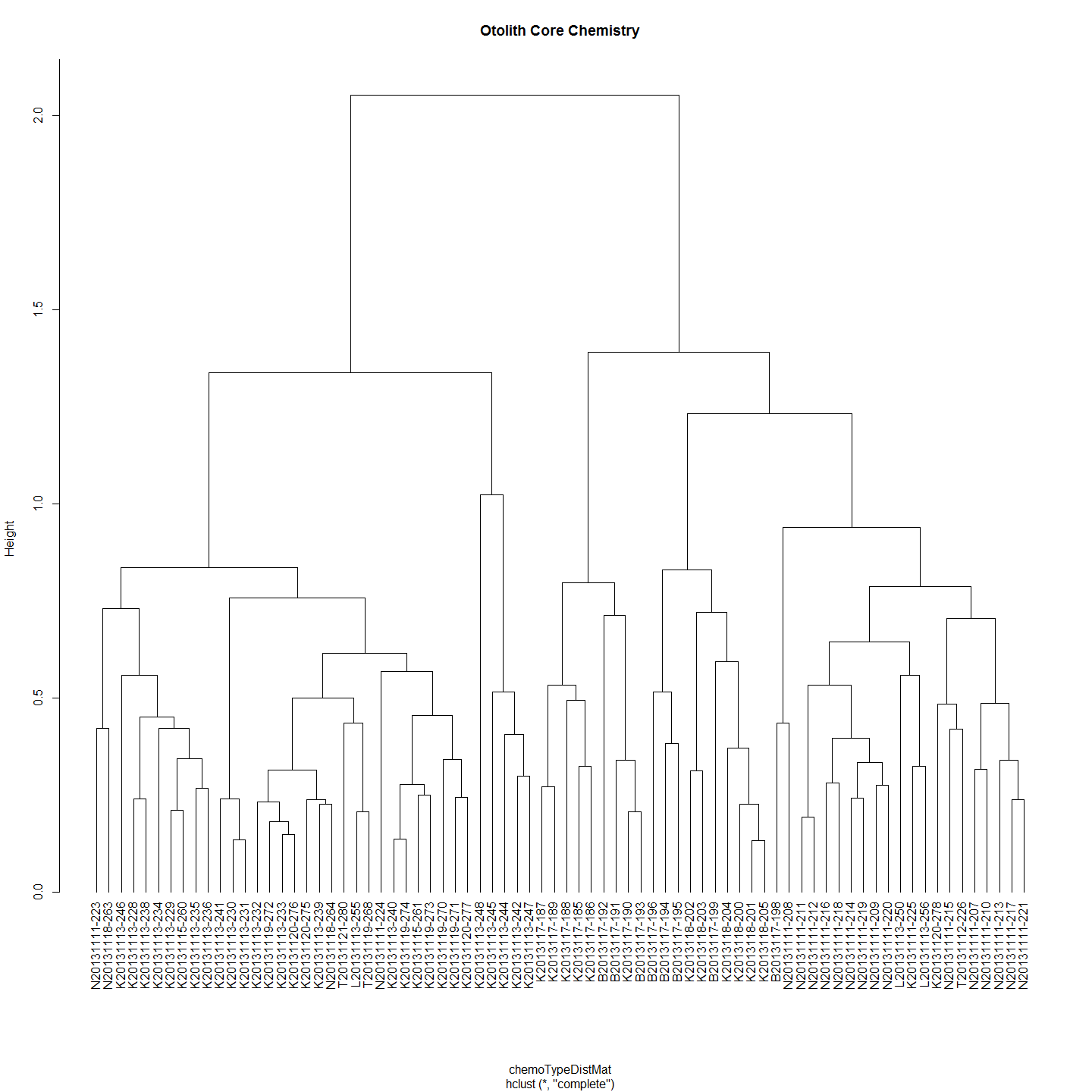


## [[1]]  
## NULL  
##   
## [[2]]  
## NULL  
##   
## [[3]]  
## NULL  
##   
## [[4]]  
## NULL  
##   
## [[5]]  
## NULL  
##   
## [[6]]  
## NULL

# otoPartChem <- function(otoParts){  
#   
# ImportantVars<-otoParts  
#   
# ImportantVars<-ImportantVars[c(2,84,13,14,15,121,122,127,131,139)] #  
# ImportantVars<-ImportantVars[complete.cases(ImportantVars),] #remove any nulls  
# #row.names(ImportantVars)<-ImportantVars[,1]  
# ImportantVars$Label<-NULL  
# colnames(ImportantVars)[1]<-"SiteName"  
#   
# ImportantVars$SiteName<-as.factor(ImportantVars$SiteName)  
#   
# df<-ImportantVars  
# print(df)  
# chemoTypeDistMat <- dist(df)  
# hClusters <- hclust(chemoTypeDistMat)  
# plot(hClusters,labels=(df$SiteName), hang = -1, main=deparse(substitute(otoParts)))  
#   
# #Make distance matrices for geographic distance as well  
# ImportantVars<-otoParts  
# ImportantVars<-ImportantVars[c(2,84,13,14,15,107,121,122,127,131,139)]  
# ImportantVars<-ImportantVars[complete.cases(ImportantVars),] #remove any nulls  
# ImportantVars<-ImportantVars[c(1,6)] # Distance from Angle Crossing changes from 107 above to 6  
# #row.names(ImportantVars)<-ImportantVars[,1]  
# ImportantVars$Label<-NULL  
#   
# geoDist<-ImportantVars  
# geoDist<-na.omit(geoDist)  
# #geoDistColl1000<-geodist #save this estimate for haplogroups distance plot (after the Iterated Mantel has changed it)  
#   
# geoDistMat<-dist(geoDist)  
#   
# #make sure both matrices are in correct order - rows and cols  
# #Check all is in order  
# print(as.matrix(geoDistMat)[1:5, 1:5]) # zero distances in the first 5  
# print(as.matrix(chemoTypeDistMat)[1:5, 1:5])  
#   
# #Conduct Mantel Test on Matrices  
# mantIV<-mantel.rtest(as.dist(geoDistMat), as.dist(chemoTypeDistMat), nrepet = 9999)  
# print(mantIV)  
# plot(geoDistMat,chemoTypeDistMat, main="Otolith Core Chemistry")  
# }  
  
##############################  
  
#But now with important variables only AFTER SCALING. These include C13, N15, CN ratio, B, K, V, Na, Rb.  
ImportantVars<-ChemAnalCore  
  
ImportantVars<-dplyr::select(ImportantVars, Label, SiteName,Delta13C,Delta15N,CNRatio,B,K,V,Na,Rb)#ImportantVars<-ImportantVars[c(2,84,13,14,15,121,122,127,131,139)] #  
ImportantVars<-ImportantVars[complete.cases(ImportantVars),] #remove any nulls  
row.names(ImportantVars)<-ImportantVars[,1]  
ImportantVars$Label<-NULL  
colnames(ImportantVars)[1]<-"SiteName"  
  
ImportantVars$SiteName<-as.factor(ImportantVars$SiteName)  
  
ImportantVars$SiteName<-NULL  
ImportantVars <- droplevels(ImportantVars)#Not sure why get error without this line  
ImpVarScaled<-data.Normalization (ImportantVars,type="n4",normalization="column")  
ImpVarScaled$Label<-row.names(ImpVarScaled)  
sChemAnalCore<-subset(ChemAnalCore, select=c(Label,SiteName))  
ImportantVars<-merge(ImpVarScaled, sChemAnalCore,by = "Label")  
ImportantVars$Label<-NULL  
df<-ImpVarScaled  
distxy <- dist(df)  
hClusters <- hclust(distxy)  
plot(hClusters,labels=(df$Site.Name), hang = -1, main="Otolith Core Chemistry")



df<-ImpVarScaled  
  
#df<-ImportantVars  
chemoTypeDistMat <- dist(df)  
hClusters <- hclust(chemoTypeDistMat)  
plot(hClusters,labels=(df$SiteName), hang = -1, main="Otolith Core Chemistry")



#Make distance matrices for geographic distance as well  
ImportantVars<-ChemAnalCore  
ImportantVars<-dplyr::select(ImportantVars, Label, SiteName,Delta13C,Delta15N,CNRatio,B,K,V,Na,Rb)#ImportantVars<-ImportantVars[c(2,84,13,14,15,107,121,122,127,131,139)]  
ImportantVars<-ImportantVars[complete.cases(ImportantVars),] #remove any nulls  
ImportantVars<-ImportantVars[c(1,6)] # Distance from Angle Crossing changes from 107 above to 6  
row.names(ImportantVars)<-ImportantVars[,1]  
ImportantVars$Label<-NULL  
  
geoDist<-ImportantVars  
geoDist<-na.omit(geoDist)  
#geoDistColl1000<-geodist #save this estimate for haplogroups distance plot (after the Iterated Mantel has changed it)  
  
geoDistMat<-dist(geoDist)  
  
#make sure both matrices are in correct order - rows and cols  
#Check all is in order  
as.matrix(geoDistMat)[1:5, 1:5] # zero distances in the first 5

## K2013117-185 K2013117-186 K2013117-187 K2013117-188 K2013117-189  
## K2013117-185 0.0000000 0.1040563 0.9801282 0.5760966 1.0036077  
## K2013117-186 0.1040563 0.0000000 1.0841845 0.4720403 1.1076640  
## K2013117-187 0.9801282 1.0841845 0.0000000 1.5562248 0.0234795  
## K2013117-188 0.5760966 0.4720403 1.5562248 0.0000000 1.5797043  
## K2013117-189 1.0036077 1.1076640 0.0234795 1.5797043 0.0000000

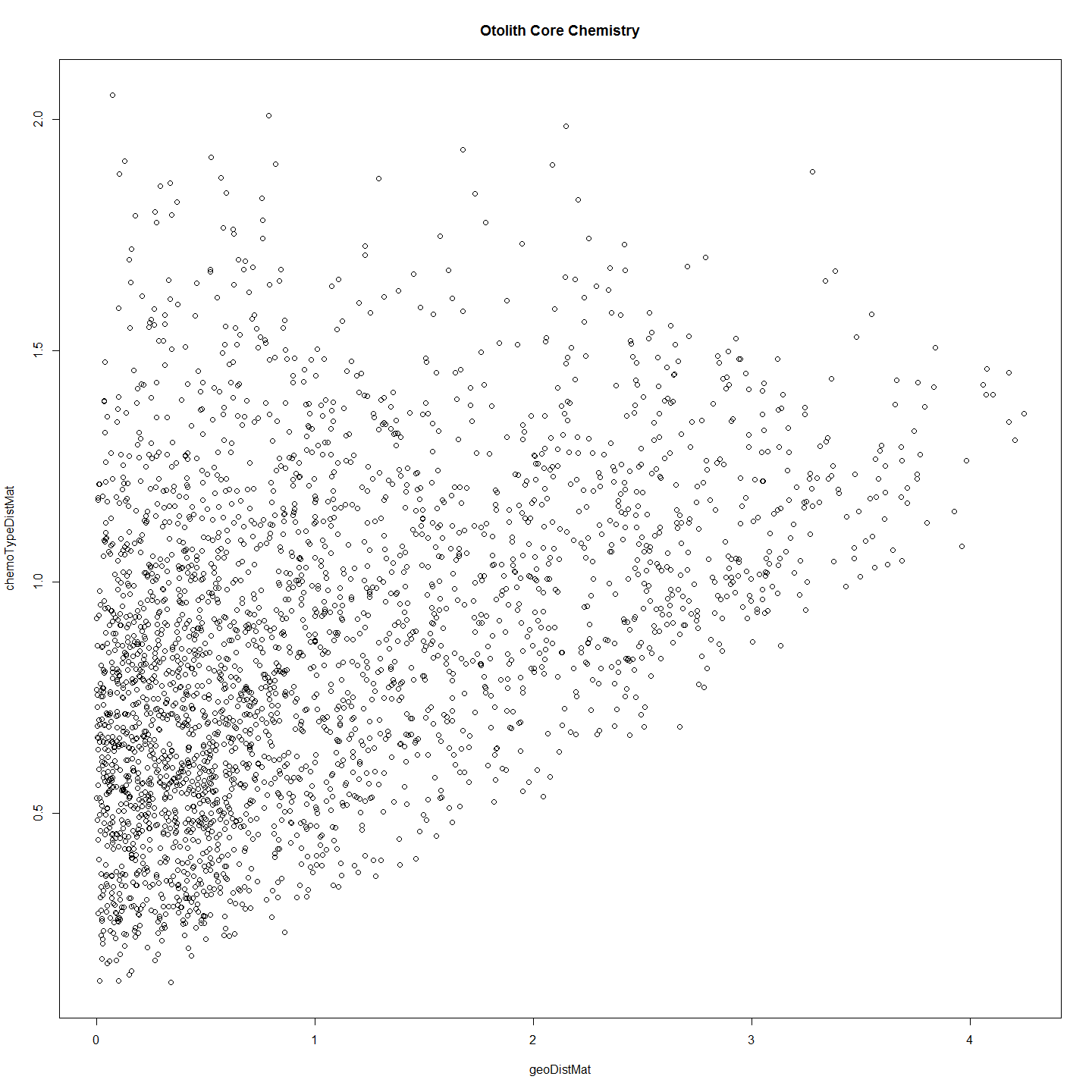
as.matrix(chemoTypeDistMat)[1:5, 1:5]

## K2013117-185 K2013117-186 K2013117-187 K2013117-188 K2013117-189  
## K2013117-185 0.0000000 0.3253653 0.3834778 0.4946802 0.5130590  
## K2013117-186 0.3253653 0.0000000 0.3437837 0.3967595 0.3405441  
## K2013117-187 0.3834778 0.3437837 0.0000000 0.4501161 0.2710697  
## K2013117-188 0.4946802 0.3967595 0.4501161 0.0000000 0.5329379  
## K2013117-189 0.5130590 0.3405441 0.2710697 0.5329379 0.0000000

#Conduct Mantel Test on Matrices  
mantIVScaled<-mantel.rtest(as.dist(geoDistMat), as.dist(chemoTypeDistMat), nrepet = 9999)  
mantIVScaled

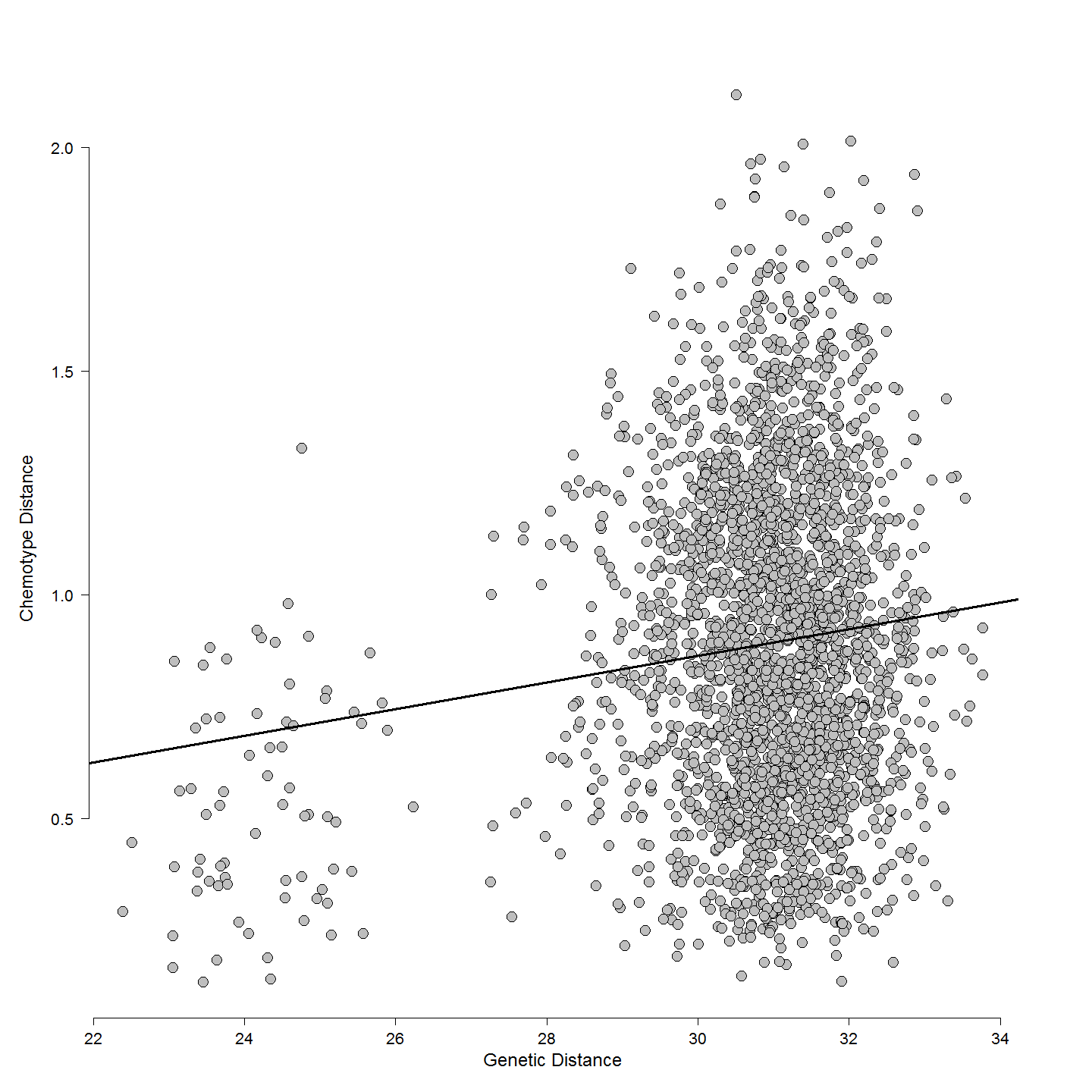
## Monte-Carlo test  
## Observation: 0.4249114   
## Call: mantel.rtest(m1 = as.dist(geoDistMat), m2 = as.dist(chemoTypeDistMat),   
## nrepet = 9999)  
## Based on 9999 replicates  
## Simulated p-value: 1e-04

plot(geoDistMat,chemoTypeDistMat, main="Otolith Core Chemistry")

 Based on these results, we cannot reject the null hypothesis that these two matrices, spatial distance and chemotype distance, are unrelated with alpha = 0.4494. The observed correlation, -0.0041274, suggests that the matrix entries are not positively associated.

### Is Chemotype Distance Correlated with Genotype Distance?

library(ade4)  
  
MCdmPrep<-MCsnps[-c(1:7),]  
  
test<-merge(MCdmPrep,ImpVarScaled,by = "row.names")  
genDist<-dist(test[,c(1:21076)])  
chemDist<-dist(test[,c(21077:21086)])  
  
require(plotrix)  
op <- par(cex.main = 1.5, mar = c(5, 6, 4, 5) + 0.1, mgp = c(3.5, 1, 0), cex.lab = 1.5 , font.lab = 2, cex.axis = 1.3, bty = "n", las=1)  
plot(genDist,chemDist, col="black", pch=21, bg = "grey", cex = 2,  
 ylab="", xlab="", axes=F)  
axis(1)  
axis(2)   
reg1 <- lm(chemDist~genDist)  
ablineclip(reg1, lwd=3)   
par(las=0)  
mtext("Genetic Distance", side=1, line=2.5, cex=1.5)  
mtext("Chemotype Distance", side=2, line=3.7, cex=1.5)



cor(genDist,chemDist)

## [1] 0.1202282

rcorr(genDist[lower.tri(genDist)],chemDist[lower.tri(chemDist)])#(x, type="pearson") # type can be pearson or spearman

## x y  
## x 1.0 0.1  
## y 0.1 1.0  
##   
## n= 1998   
##   
##   
## P  
## x y   
## x 0  
## y 0

mant<-mantel.rtest(genDist, chemDist, nrepet = 9999)  
mant

## Monte-Carlo test  
## Observation: 0.1202282   
## Call: mantelnoneuclid(m1 = m1, m2 = m2, nrepet = nrepet)  
## Based on 9999 replicates  
## Simulated p-value: 0.0035

#Check all is in order  
as.matrix(genDist)[1:5, 1:5]

## 1 2 3 4 5  
## 1 0.00000 30.07891 23.66292 30.12886 30.33396  
## 2 30.07891 0.00000 30.04523 28.73927 30.06496  
## 3 23.66292 30.04523 0.00000 30.12028 29.26381  
## 4 30.12886 28.73927 30.12028 0.00000 29.86714  
## 5 30.33396 30.06496 29.26381 29.86714 0.00000

as.matrix(chemDist)[1:5, 1:5]

## 1 2 3 4 5  
## 1 0.0000000 0.7524571 0.3492786 1.2691199 1.1964719  
## 2 0.7524571 0.0000000 0.5106939 1.0790292 0.8908128  
## 3 0.3492786 0.5106939 0.0000000 1.1545236 0.9933279  
## 4 1.2691199 1.0790292 1.1545236 0.0000000 0.3816798  
## 5 1.1964719 0.8908128 0.9933279 0.3816798 0.0000000

## Discussion Points (just dot points)

This is interesting given that there is a relationship between genotype and chemotype but it is not one due to covariance based on geographic distance. One not unreasonable explaination for this is that suggest that chemotype is afftected by the genotype directly. That is, element deposition in otolith is variable but under some genetic control. Another more intriguing possibility is that the geochemistry fails to predict the capture site of the larvae because it is not the ultimte source of that geochemistry. The ultimate source of the geochemistry is thewater and food environment of the female when the yolk which was laid down in the developing eggs. This happens earlier in the season and well before spawning so it does leave ope the possibiltiy as suggested by some authors that the female has migrated to the spawning site. Unfortunately in this study there is no measure that might be suitable as a proxy regarding the origin of the female during oogenesis.

## Code Chunks in this Document

## [1] "Project\_Template\_and\_Knitr" "Set\_Global\_Options" "LoadLibraries" "unnamed-chunk-1"   
## [5] "unnamed-chunk-2" "RandomForestElemGo" "unnamed-chunk-3" "RandomForestElemIsoGo"   
## [9] "unnamed-chunk-4" "ldaGo" "unnamed-chunk-5" "ordLogRegnGo"   
## [13] "unnamed-chunk-6" "chemotypeDistanceMatricesGo" "unnamed-chunk-7" "genChemDistPlotGo"   
## [17] "Include\_Chunk\_Labels\_and\_Session Information" "RandomForestElem" "RandomForestElemIso" "lda"   
## [21] "ordLogRegn" "chemotypeDistanceMatrices" "genChemPlot"

## R version 3.2.2 (2015-08-14)  
## Platform: x86\_64-w64-mingw32/x64 (64-bit)  
## Running under: Windows 7 x64 (build 7601) Service Pack 1  
##   
## locale:  
## [1] LC\_COLLATE=English\_Australia.1252 LC\_CTYPE=English\_Australia.1252 LC\_MONETARY=English\_Australia.1252 LC\_NUMERIC=C LC\_TIME=English\_Australia.1252   
##   
## attached base packages:  
## [1] grid stats graphics grDevices utils datasets methods base   
##   
## other attached packages:  
## [1] plotrix\_3.5-12 clusterSim\_0.44-2 cluster\_2.0.3 MASS\_7.3-44 randomForest\_4.6-10 dplyr\_0.4.3 ade4\_1.7-2 Hmisc\_3.17-0 Formula\_1.2-1   
## [10] survival\_2.38-3 lattice\_0.20-33 dendextend\_1.1.0 ape\_3.3 ggdendro\_0.1-17 ggplot2\_1.0.1 ProjectTemplate\_0.6 knitr\_1.11   
##   
## loaded via a namespace (and not attached):  
## [1] Rcpp\_0.12.1 formatR\_1.2.1 RColorBrewer\_1.1-2 plyr\_1.8.3 class\_7.3-14 tools\_3.2.2 rpart\_4.1-10 digest\_0.6.8 evaluate\_0.8   
## [10] nlme\_3.1-122 gtable\_0.1.2 DBI\_0.3.1 R2HTML\_2.3.1 parallel\_3.2.2 yaml\_2.1.13 proto\_0.3-10 e1071\_1.6-7 gridExtra\_2.0.0   
## [19] stringr\_1.0.0 nnet\_7.3-11 R6\_2.1.1 rgl\_0.95.1337 foreign\_0.8-66 rmarkdown\_0.8 latticeExtra\_0.6-26 reshape2\_1.4.1 magrittr\_1.5   
## [28] whisker\_0.3-2 scales\_0.3.0 htmltools\_0.2.6 splines\_3.2.2 assertthat\_0.1 colorspace\_1.2-6 labeling\_0.3 stringi\_0.5-5 acepack\_1.3-3.3   
## [37] lazyeval\_0.1.10 munsell\_0.4.2