

# NCI Analysis for MAC (IDENT)

*Aitor Ametztegui (CREAF - CTFC)*

*November 2016*

This is an R Markdown Notebook showing the descriptive analyses and first results of the NCI analyses carried out in MAC. The final aim is to determine the role of functional diversity on growth, through the classic NCI growth equations. For this, we first need to determine the target trees, identify which are the neighbours for each target, and define several indicators of functional diversity in the neighborhood of each tree.

## Target Trees

First thing we need is to describe our dataset, which contains 19 species (12 native, 7 exotic), that have been measured between 2009 and 2014. The total number of trees planted was 13,824, but after deleting the missing and dead trees, we kept a total of 12466 as target trees. These are the main descriptive statistics of the target trees per species

##										
##										
		Diam14		Height14		D Growth		H Growth		
##	CodeSp n	Mean	Sd	Mean	Sd	Mean	Sd	Mean	Sd	
##	ABBA	678	31.8	10.2	196.3	66.0	26.1	10.0	169.8	64.8
##	ACPL	429	40.1	18.7	415.4	157.1	33.7	18.7	392.0	157.0
##	ACRU	822	41.1	17.4	388.6	132.6	35.9	17.3	352.1	132.9
##	ACSA	841	35.4	12.5	396.8	123.0	31.1	12.4	348.6	121.7
##	BEAL	726	40.8	18.6	412.7	141.3	37.4	18.5	364.6	139.9
##	BEPa	893	60.4	21.8	627.7	126.7	56.7	21.8	582.5	126.5
##	LADE	258	41.6	22.4	377.8	176.1	38.6	22.3	353.0	174.8
##	LALA	842	51.7	16.8	491.1	112.4	47.3	16.6	453.5	111.9
##	PIAB	461	35.4	9.5	252.0	75.9	29.8	9.2	208.4	74.6
##	PIGL	980	32.9	8.7	192.4	57.7	27.3	8.5	154.5	57.1
##	PIOM	235	20.7	8.8	83.3	42.8	16.6	8.7	64.3	42.7
##	PIRE	742	41.2	11.3	233.5	60.1	36.3	11.2	212.6	59.3
##	PIRU	725	28.6	8.5	182.8	61.7	23.8	8.4	140.0	60.9
##	PIST	1136	37.2	12.4	249.4	91.6	32.0	12.3	226.4	91.4
##	PISY	497	41.9	14.9	228.1	80.4	36.9	14.9	192.9	79.3
##	QURO	351	39.4	16.0	390.9	108.3	36.8	16.0	375.7	108.3
##	QURU	706	48.2	20.5	459.9	116.3	39.2	19.4	437.9	116.2
##	THOC	812	37.4	14.0	205.9	67.2	33.9	13.9	177.2	66.7
##	TICO	332	45.2	25.1	404.9	142.5	42.3	24.9	380.3	141.0
##	All	12466	40.2	17.5	331.1	170.7	35.4	17.5	297.7	168.2

We can see that the exotic species are less abundant (LADE: 258; PIOM: 235) than the native ones, for which there are always > 700 individuals in the dataset (this contrasts with previous database, where some species were much rarer). The mean diameter growth for all the dataset is 7.07 mm per year in diameter, and 59.53 cm per year in height. We can also see that diameter growth is more consistent across species than height growth, for which the fastest growing species (BEPa: 5.82 m) grew almost 10 times faster than the slowest growing one (PIOM = 0.6m)

## Determination of neighbours

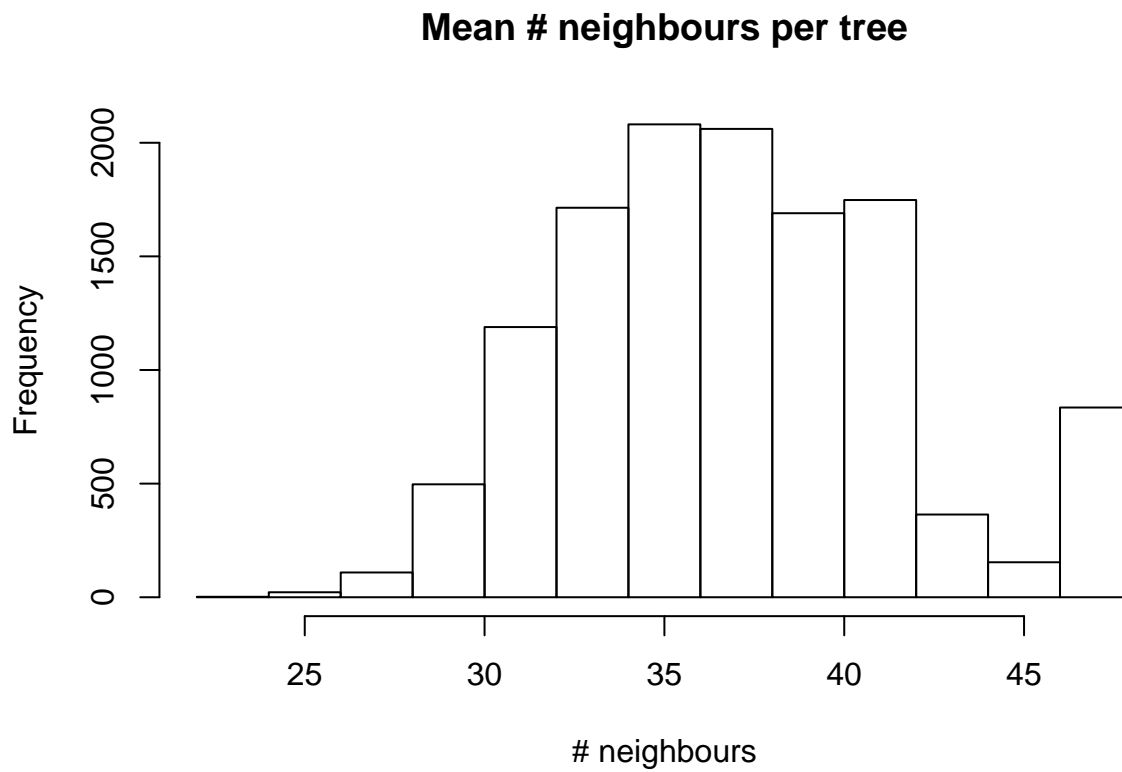
As we will work at an individual level, not at the plot level, we must identify the neighbours of each target tree. We do this externally, using ArcGIS and the Generate Near Table utility. We must note that every target tree has a different and unique neighborhood, and as a consequence, a given species may not exist in the neighborhood of another given species. This can have computational consequences, so first we need to know which species are neighbours of which ones.

```
mytable <- table(neighbours$CodeSp_IN, neighbours$CodeSp_NEAR)
kable(addmargins(mytable,2), format = 'markdown' )
```

	ABBA	ACPA	ACRU	ACSA	BEAL	BEPAL	LADE	ALAPI	PIGL	PION	PIRE	PIRU	PIST	PISY	QURQ	QURU	THO	TICQ	Sum	
ABBA	2470	293	2390	2541	312	742	141	776	76	1560	337	966	1299	351	99	222	334	270	225	25404
ACPA	296	8523	722	646	629	1068	81	167	562	207	545	175	202	205	315	240	198	511	604	15896
ACRU	227	738	14556	528	3236	834	74	519	225	388	120	490	635	1008	352	186	1238	2588	678	30720
ACSA	468	634	513	14837	794	1207	109	2160	166	1851	778	417	750	1019	233	201	285	2662	359	31443
BEAL	297	632	3121	795	12831	773	30	464	405	844	262	713	956	937	272	224	2136	891	614	27197
BEPAL	40	1104	833	1247	807	14848	210	1104	657	1540	598	560	1155	2937	649	761	2684	825	174	33433
LADE	104	68	60	102	25	185	6471	116	132	149	44	632	132	307	485	190	428	96	59	9785
LALAPI	89	170	519	2213	478	1101	150	15229	654	2868	321	198	808	2797	656	194	941	1050	267	31403
PIAB	77	589	249	173	411	655	156	661	9103	642	119	275	685	449	1192	207	202	1175	148	17168
PIGL	1578	201	374	1918	863	1524	179	2863	643	15361	191	1288	1616	3813	933	522	509	1371	813	36560
PION	60	352	79	481	177	393	42	255	73	135	5624	165	52	301	78	43	173	90	112	8885
PIRE	228	165	481	421	656	518	594	189	242	1242	237	13779	3544	3023	441	142	665	420	166	27853
PIRU	245	199	611	724	900	1086	148	735	648	1561	102	3553	12188	1188	1048	699	377	277	118	27407
PIST	357	201	993	1105	964	2894	333	2757	443	3835	361	3162	1295	18802	300	719	1481	1468	866	42336
PISY	88	309	351	245	272	584	516	591	1119	937	120	535	1005	300	9426	563	540	915	201	18617
QURQ	269	218	189	167	151	738	203	169	191	561	74	145	758	724	619	7268	112	92	503	13151
QURU	329	191	1243	282	2356	2664	481	934	200	490	239	815	453	1501	565	112	11971	1462	140	26428
THO	273	516	2494	2979	939	801	109	1057	1122	1396	102	438	298	1488	919	107	1457	13821	261	30577
TICQ	60	593	597	219	605	148	64	238	131	715	156	148	114	774	175	449	118	188	6638	12230

Although some combinations are rarer than others, with the new dataset there are many more neighbours for all combinations of species. The minimum is the 25 BEAL (*Betula alleghaniensis*) in the neighbourhood of LADE (*Larix decidua*).

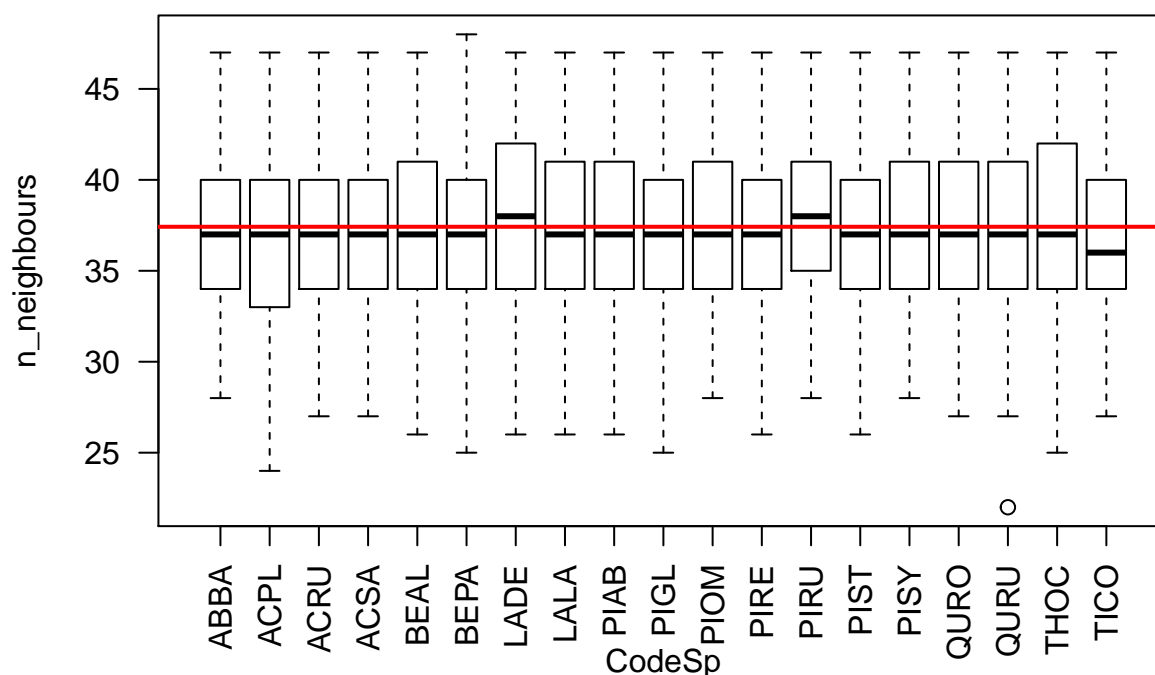
Let's see how many neighbours has a target tree in average



It is also important to know if this value (mean #neighbours) is consistent across species.

```
plot(n_neighbours~ CodeSp, data=targets, las =2, main = "# Neighbours per species")  
abline(mean(targets$n_neighbours, na.rm=T), 0, col="red", lwd=2, cex=0.8)
```

## # Neighbours per species



Another thing to consider is if the mean distances to neighbours of different species are similar for all target species, since competition coefficients might be influenced by spatial segregation of species. Thus, we determine the mean distances to neighbours of different species.

```
melt_distance <- neighbours %>%
  group_by(CodeSp_IN, CodeSp_NEAR) %>%
  summarise(mean_dist=mean(NEAR_DIST),
            sd_dist=sd(NEAR_DIST))

neig_sps_dist <- dcast(data=melt_distance, CodeSp_IN~CodeSp_NEAR,
  value.var= "mean_dist")
row.names(neig_sps_dist)= neig_sps_dist$CodeSp_IN
kable(round(neig_sps_dist[,-1],2),format = "markdown")
```

	ABBA	ACPL	ACRU	ACSA	BEAL	BEPA	LADE	LALA	PIAB	PIGL	PIOM	PIRE	PIRU	PIST	PISY	QURO	QURU	THOC	TICO
ABBA	1.20	1.66	1.32	1.30	1.55	1.41	1.55	1.37	1.51	1.29	1.61	1.45	1.37	1.65	1.54	1.58	1.63	1.56	1.57
ACPL	1.66	1.20	1.44	1.37	1.36	1.34	1.34	1.50	1.27	1.55	1.21	1.51	1.58	1.54	1.52	1.54	1.64	1.31	1.27
ACRU	1.31	1.43	1.20	1.56	1.31	1.42	1.72	1.61	1.61	1.63	1.51	1.66	1.57	1.48	1.67	1.63	1.33	1.29	1.36
ACSA	1.30	1.37	1.56	1.20	1.37	1.35	1.68	1.30	1.54	1.36	1.41	1.64	1.41	1.47	1.64	1.58	1.64	1.34	1.63
BEAL	1.55	1.37	1.31	1.37	1.20	1.42	1.78	1.62	1.66	1.45	1.63	1.40	1.39	1.46	1.62	1.59	1.28	1.44	1.36
BEPA	1.40	1.33	1.42	1.36	1.41	1.21	1.61	1.48	1.34	1.37	1.36	1.61	1.44	1.33	1.34	1.39	1.29	1.41	1.53
LADE	1.55	1.35	1.72	1.67	1.77	1.61	1.22	1.61	1.45	1.55	1.22	1.38	1.62	1.59	1.30	1.68	1.26	1.53	1.45
LALA	1.37	1.51	1.61	1.31	1.62	1.49	1.59	1.21	1.35	1.31	1.61	1.60	1.37	1.30	1.36	1.59	1.43	1.43	1.61
PIAB	1.51	1.27	1.61	1.55	1.66	1.35	1.48	1.36	1.20	1.33	1.37	1.65	1.36	1.67	1.30	1.49	1.60	1.34	1.44
PIGL	1.30	1.55	1.63	1.37	1.46	1.37	1.54	1.31	1.34	1.21	1.57	1.36	1.33	1.31	1.43	1.30	1.65	1.34	1.40
PIOM	1.61	1.22	1.57	1.42	1.66	1.36	1.23	1.64	1.37	1.56	1.21	1.60	1.51	1.64	1.29	1.51	1.62	1.45	1.43

	ABBA	ACPL	ACRU	ACSABE	ALBEP	ALADEL	LALAPI	ABPIGL	PIOMPIRE	PIRU	PIST	PISY	QURQ	QURU	THO	TICO			
PIRE	1.44	1.52	1.65	1.65	1.41	1.61	1.40	1.59	1.63	1.36	1.62	1.21	1.29	1.31	1.37	1.68	1.45	1.62	1.62
PIRU	1.36	1.58	1.56	1.41	1.40	1.44	1.62	1.36	1.37	1.33	1.52	1.29	1.20	1.39	1.30	1.36	1.59	1.58	1.64
PIST	1.64	1.54	1.48	1.48	1.46	1.33	1.59	1.30	1.67	1.31	1.64	1.31	1.40	1.20	1.62	1.41	1.41	1.39	1.39
PISY	1.53	1.52	1.67	1.64	1.62	1.35	1.28	1.36	1.30	1.43	1.24	1.34	1.31	1.61	1.21	1.29	1.32	1.42	1.43
QURQ	1.59	1.54	1.62	1.52	1.58	1.40	1.67	1.55	1.50	1.31	1.43	1.69	1.37	1.40	1.28	1.20	1.71	1.70	1.24
QURU	1.62	1.64	1.33	1.64	1.28	1.29	1.26	1.43	1.62	1.64	1.64	1.41	1.60	1.40	1.33	1.69	1.20	1.36	1.52
THO	1.57	1.30	1.29	1.34	1.45	1.40	1.52	1.43	1.33	1.35	1.44	1.62	1.59	1.38	1.43	1.68	1.36	1.20	1.63
TICO	1.55	1.28	1.35	1.60	1.36	1.52	1.44	1.61	1.43	1.39	1.43	1.59	1.65	1.38	1.43	1.24	1.52	1.61	1.20

We see that both the mean number of neighbours and the mean distance are very similar across species, so we don't need to worry about spatial segregation of species.

# Functional Diversity

## Functional Traits Database

For this study, we characterized three components of functional diversity: functional difference, functional dispersion and effective species richness. In all cases, the dataset of traits used is the same:

	Wd	SeedMass	LMA	Nmass	Amass	Llo	ShadeT	DroughtT	WaterT
ABBA	0.33	7.6	143.00	1.45	50.00	84.0	5.01	1.00	2.00
ACPL	0.52	130.3	46.45	1.62	83.07	6.0	4.20	2.73	1.46
ACRU	0.49	23.7	71.09	1.91	111.15	5.6	3.44	1.84	3.08
ACSA	0.56	55.2	70.63	1.83	84.58	5.5	4.76	2.25	1.09
BEAL	0.55	1.0	46.08	2.20	206.00	5.5	3.17	3.00	2.00
BEPA	0.48	0.3	85.51	2.21	195.00	5.5	1.54	2.02	1.25
LADE	0.47	7.1	86.29	1.96	71.62	6.0	1.46	2.31	1.10
LALA	0.49	2.0	120.00	1.36	59.42	6.0	0.98	2.00	3.00
PIAB	0.37	7.0	235.18	1.19	29.31	103.0	4.45	1.75	1.22
PIGL	0.33	2.4	302.86	1.28	35.58	50.0	4.15	2.88	1.02
PIOM	0.36	2.9	285.73	1.18	29.39	62.0	4.65	2.75	1.03
PIRE	0.41	9.0	294.12	1.17	24.00	36.0	1.89	3.00	1.00
PIRU	0.37	3.3	263.00	1.20	32.00	55.0	4.39	2.50	2.00
PIST	0.34	17.0	121.92	1.42	43.80	20.0	3.21	2.29	1.03
PISY	0.42	6.0	292.76	1.22	36.85	28.0	1.67	4.34	2.63
QURO	0.56	3378.0	64.85	1.86	85.10	6.0	2.45	2.95	1.89
QURU	0.56	3143.0	88.03	2.03	148.55	6.0	2.75	2.88	1.12
THOC	0.30	1.5	223.00	1.02	32.22	33.0	3.45	2.71	1.46
TICO	0.42	50.9	49.11	2.12	83.85	4.8	4.18	2.75	1.83

To calculate the several diversity indices we need to take into account the relative abundances of each species on each neighborhood, based on the sum of diameters per species. We will compute this separately for the target tree and the neighbours

```
# Organize the files to get accumulated diameter per each neighbour species
melt_sum <- neighbours %>%
  group_by(Tree_ID_IN, CodeSp_NEAR) %>%
    summarise_each (funs(sum(DB2009)))
neig_sps_diam <- dcast(data=melt_sum, Tree_ID_IN~CodeSp_NEAR, value.var= "DB2009")
neig_sps_diam <- as.matrix(neig_sps_diam [, -1])

# Organize target files to get the diam value for each target, and in the same format as neighbours
targ_sps_diam <- dcast(targets, Tree_ID~CodeSp, value.var="DB2009")
targ_sps_diam <- as.matrix(targ_sps_diam [, -1])
```

...and now we can compute the several FD indices using the FD package

```
library(FD)

# Neighbour Traits (CWM of neighbours)
#FD_Voisins <- funtcomp(std_traits, neig_sps_diam)

FD_neighbours <- dbFD(std_traits, neig_sps_diam,
  calc.FRic = F, calc.FGR = F, calc.FDiv = F,
```

```

    clust.type = "ward")

## FEVe: Could not be calculated for communities with <3 functionally singular species.
## FDis: Equals 0 in communities with only one functionally singular species.

# Target Traits (CWM of target)
#FD_Cible <- functcomp(std_traits, targ_sps_diam)
FD_targets <- dbFD(std_traits,targ_sps_diam,
                  calc.FRic = F,calc.FGR = F,calc.FDiv = F,
                  clust.type = "ward")

## FEVe: Could not be calculated for communities with <3 functionally singular species.
## FDis: Equals 0 in communities with only one functionally singular species.

```

## Functional Difference (FDif)

The first variable of interest is functional difference (FDif), calculated as the difference in trait values between the target and its neighbours. Actually, there are two possible ways to calculate this:

### (1) Based on mean of the differences between traits

We know CWM for each trait for target and neighbours, and FDif is the difference between these two values. Since CWM is trait-specific, we have to calculate a unidimensional variable, so we compute the quadratic mean of the differences. By definition, FDif can get values of 0 when all the neighbours of a target tree are conspecific, so we modify this by adding 1 to all FDif values.

```

# Difference between target and neighbour CWM values
traits_dif <- (FD_targets$CWM - FD_neighbours$CWM)

# Quadratic mean
FDif <- data.frame("FDif"= sqrt(rowSums(traits_dif^2)/
                                     ncol(traits_dif))+1)
targets <- cbind (targets, FDif)

```

### (2) Based on the difference in functional identity

Calculating CWM would provide one value per neighborhood and per trait, so in order to obtain a single value we first perform a PCA with trait values, and then calculate a CWM of the first component of the PCA.

```

# Perform PCA to determine Functional Identity

# Perform the PCA
pca_traits <- princomp(traits, cor=T)
#summary(pca_traits)
#loadings(pca_traits)
#scores(pca_traits)
PC1=data.frame(Axis1=pca_traits$scores[,1])
kable(round(PC1,3),format = "markdown")

```

	Axis1
ABBA	2.049
ACPL	-1.025
ACRU	-1.639

	Axis1
ACSA	-1.208
BEAL	-3.044
BEPA	-2.545
LADE	-1.265
LALA	-0.783
PIAB	2.958
PIGL	2.486
PIOM	2.672
PIRE	1.686
PIRU	2.207
PIST	0.872
PISY	1.004
QURO	-2.547
QURU	-2.908
THOC	2.197
TICO	-1.167

```

# CWM of Axis1 for Neighbours
FId_Voisins=functcomp(PC1, neig_sps_diam)

# CWM of Axis1 for Target
FId_Cible=functcomp(PC1, targ_sps_diam)

# Determine the difference (in absolute value)
FDif2<- abs(FId_Voisins - FId_Cible) +1
colnames(FDif2) <- "FDif2"
targets <- cbind (targets, FDif2)

```

Let's see how this new variable looks like depending on how we calculate it

```

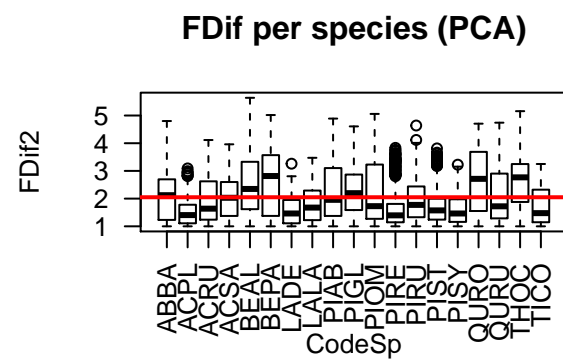
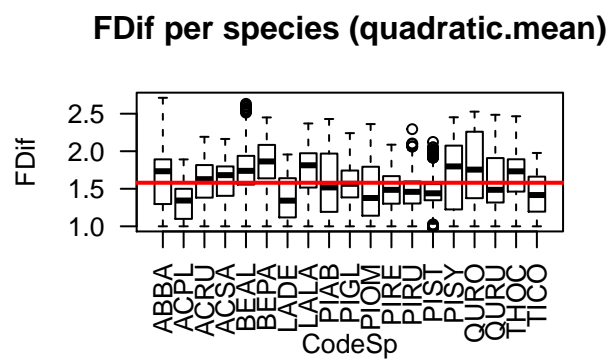
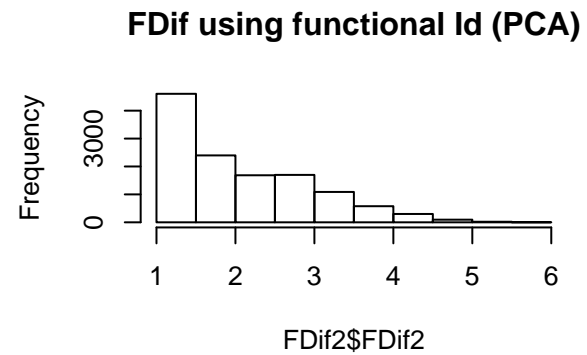
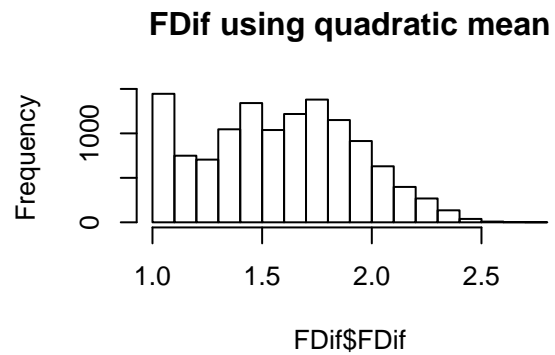
par (mfrow = c(2,2))
hist(FDif$FDif, main="FDif using quadratic mean")
hist(FDif2$FDif2, main="FDif using functional Id (PCA)")

plot(FDif~ CodeSp, data=targets, las =2, main = "FDif per species (quadratic.mean)")
abline(mean(targets$FDif, na.rm=T), 0 , col="red", lwd=2, cex=0.8)

plot(FDif2~ CodeSp, data=targets, las =2, main = "FDif per species (PCA)")
abline(mean(targets$FDif2, na.rm=T), 0 , col="red", lwd=2, cex=0.8)

```



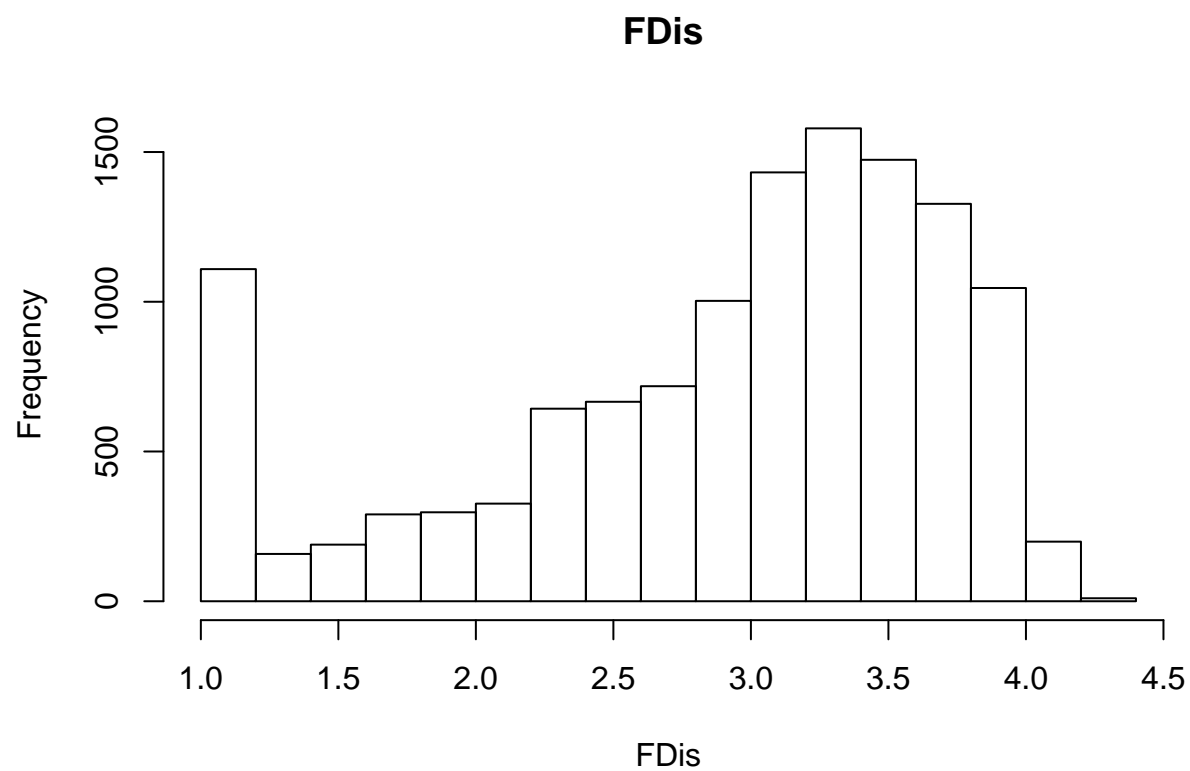


## Functional Dispersion

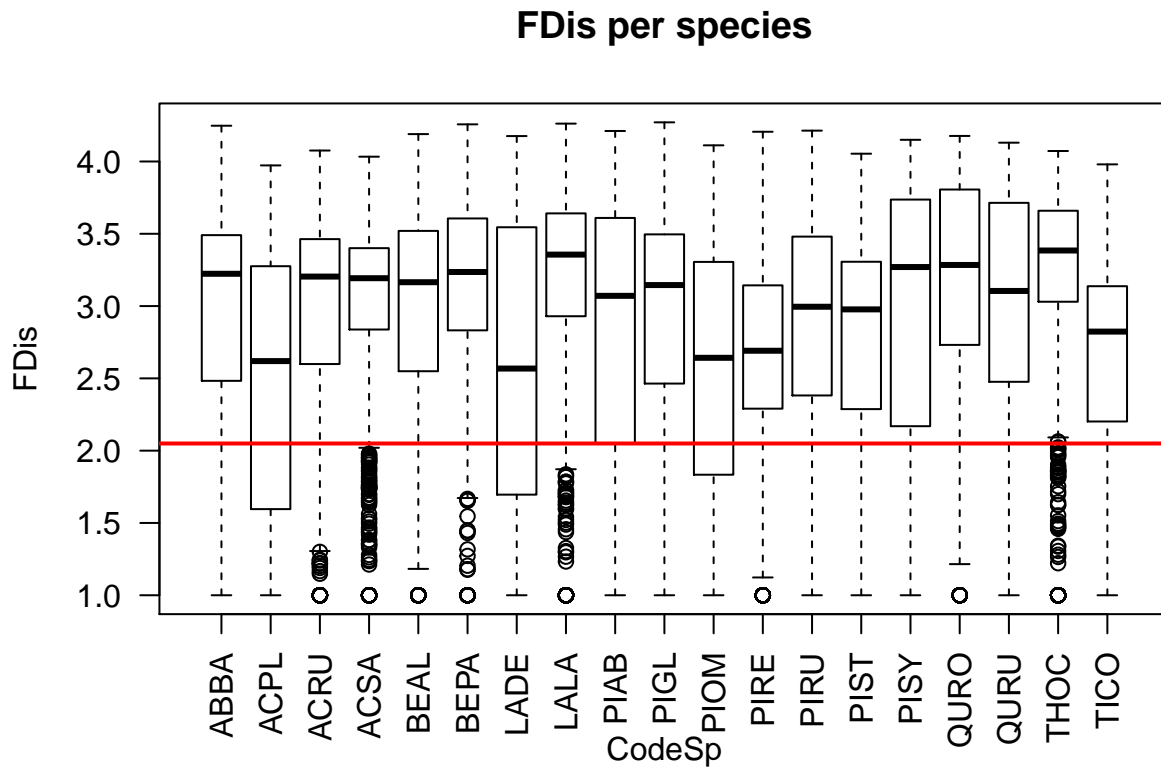
Functional dispersion was calculated previously. We also add 1 to all values to avoid 0's.

```
FDif <- FD_neighbours$FDif + 1
targets <- cbind (targets, FDif)
```

```
hist(FDif, main="FDif ")
```



```
plot(FDis~ CodeSp, data=targets,las =2,main = "FDis per species")
abline(mean(targets$FDif2, na.rm=T),0 , col="red", lwd=2, cex=0.8)
```



## Species richness

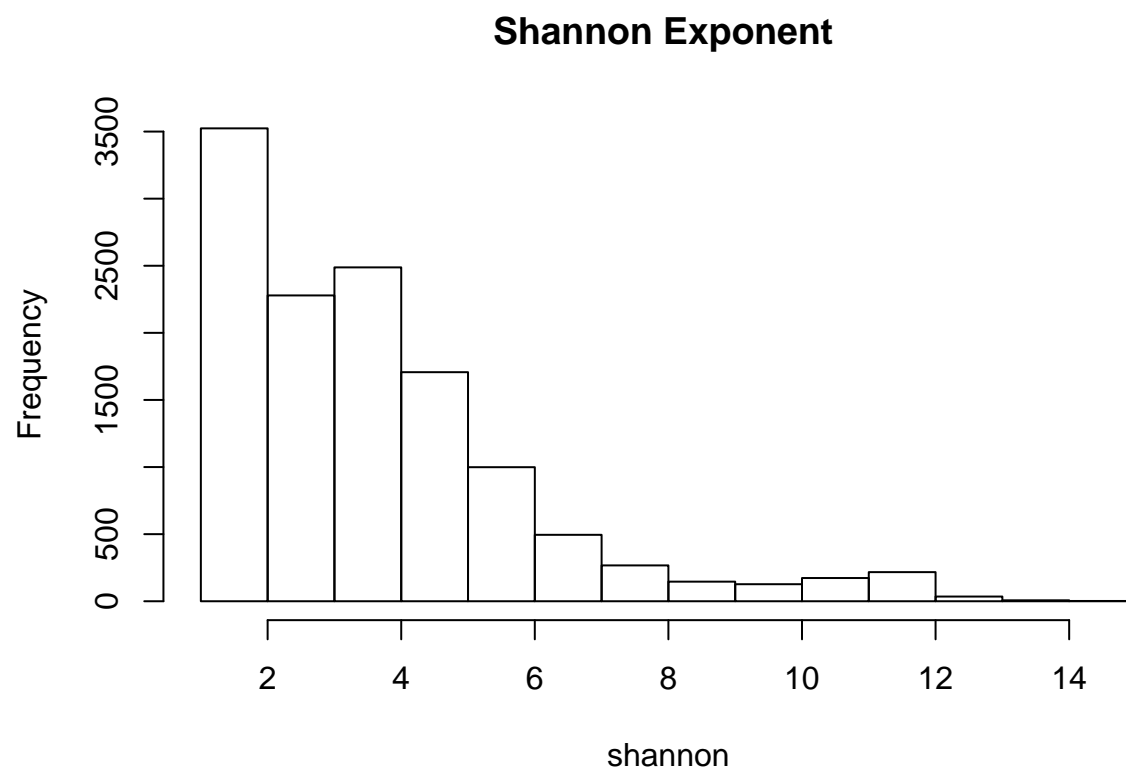
First, we determine the species richness in the neighborhood of each target tree. Then, we calculate the “effective number of species” as the exponential of the Shannon entropy

```
library(vegan)

sp_neighbours <- neighbours %>% group_by(Tree_ID_IN)%>%
  summarise(n_sp=length(unique(CodeSp_NEAR)))

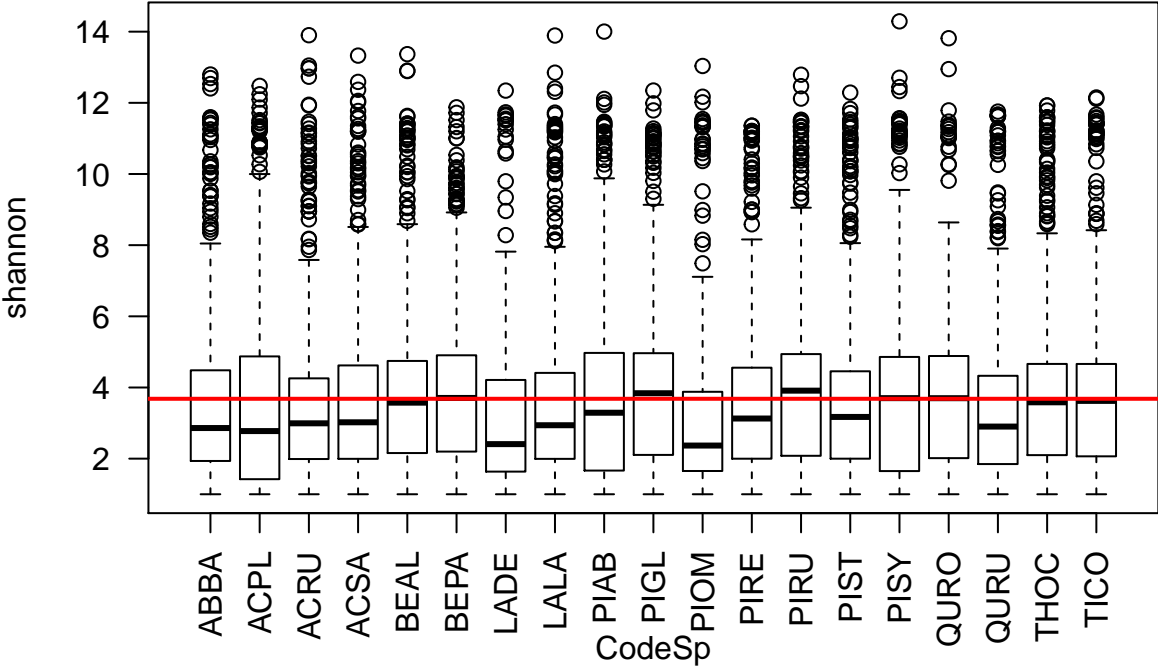
# Use the matrix of abundances based on dbh, but substitute NA for 0
neig_sps_diam_NoNA <- neig_sps_diam
neig_sps_diam_NoNA[is.na(neig_sps_diam_NoNA)] <- 0
shannon<- exp(diversity(neig_sps_diam_NoNA, index = "shannon"))
targets <- cbind (targets, shannon)

hist(shannon, main="Shannon Exponent ")
```



```
plot(shannon~ CodeSp, data=targets, las =2, main = "Shannon exponent per species")  
abline(mean(targets$shannon, na.rm=T), 0, col="red", lwd=2, cex=0.8)
```

Shannon exponent per species



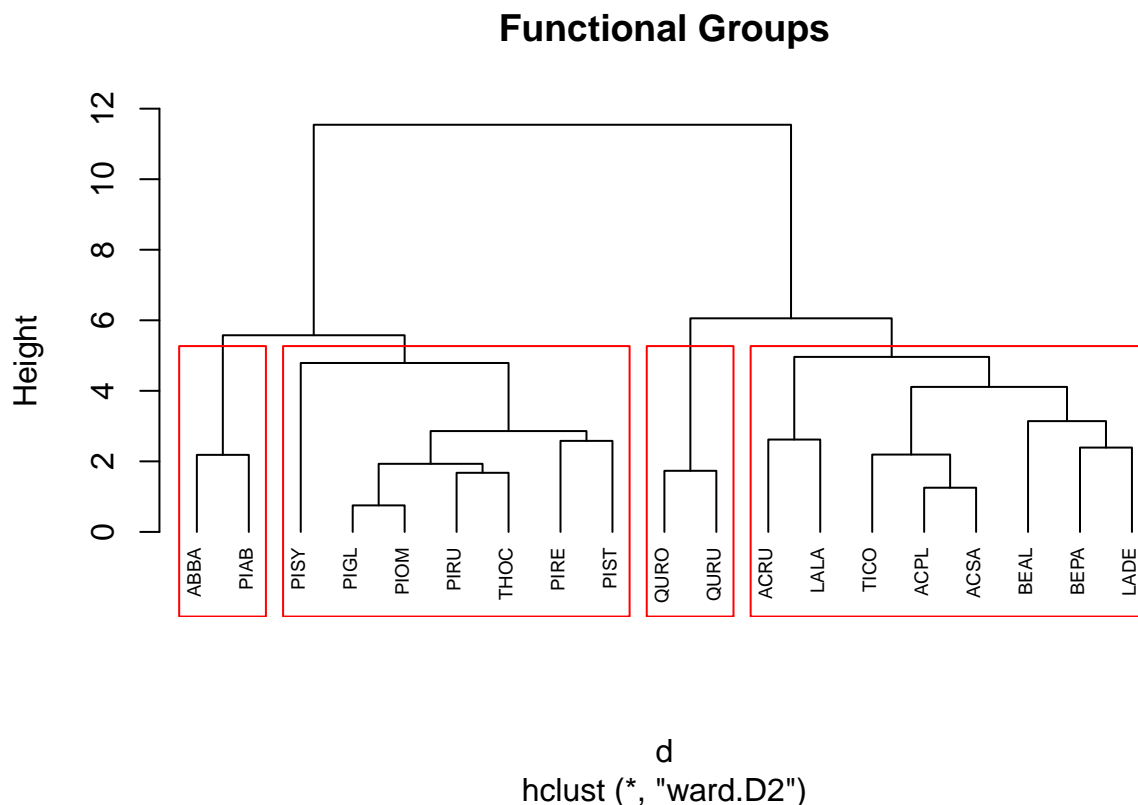
## Regroupement of species

Since some combinations of target-neighbour are rare, we may be forced to group the neighbour species into a limited number of classes. We use here two approaches: in the first, called “Functional Groups”, we performed a cluster analysis based on the values of functional traits provided above. In the second, species were classified into three groups as a function of their shade tolerance: tolerants, intermediate and intolerants.

### Based on Functional Groups

```
d<-vegdist(std_traits,method="euclidean",na.r=T)
fit <- hclust(d, method = "ward.D2")

# Plot dendrogram
plot(fit, cex=0.6,
     main = " Functional Groups ",
     hang = -1)
# Add red boxes to separate groups
rect.hclust(fit, 4)
```



We can now compute how many neighbours of each functional group there are at the neighborhood of each target species

```
# Get the cluster for each species
cluster_groups <- data.frame(CodeSp= levels(targets$CodeSp),cluster=cutree(fit, 4))
cluster_groups$cluster <- as.character(cluster_groups$cluster)
```

```

targets <- left_join(targets, cluster_groups, by=c("CodeSp"))
neighbours <- left_join(neighbours, cluster_groups, by=c("CodeSp_NEAR"="CodeSp"))

targets$cluster <- as.factor(targets$cluster)
neighbours$cluster <- as.factor(neighbours$cluster)

cluster_neighbours <- table(neighbours$CodeSp_IN, neighbours$cluster)
kable(addmargins(cluster_neighbours,2), format="markdown")

```

	1	2	3	4	Sum
ABBA	12546	7420	4882	556	25404
ACPL	858	12440	2160	438	15896
ACRU	2552	21163	5581	1424	30720
ACSA	2634	20613	7710	486	31443
BEAL	702	19260	4875	2360	27197
BEPA	1397	20327	8264	3445	33433
LADE	236	7086	1845	618	9785
LALA	1443	20127	8698	1135	31403
PIAB	9180	3042	4537	409	17168
PIGL	2221	8735	24573	1031	36560
PIOM	333	1891	6445	216	8885
PIRE	1170	3190	22686	807	27853
PIRU	1893	4521	19917	1076	27407
PIST	800	10113	29223	2200	42336
PISY	1207	3069	13238	1103	18617
QURO	460	2338	2973	7380	13151
QURU	529	8291	5525	12083	26428
THOC	1395	9156	18462	1564	30577
TICO	291	9102	2270	567	12230

So we see it actually solves the problem of the lack of species

## Based on Shade tolerance Groups

In this case, standardizing or not has no effect on the results, since we only work with one variable (shade tolerance). We performed a cluster analysis and grouped species in three categories (intolerant, tolerant and intermediate)

```

TolS <- as.matrix(traits$ShadeT) #cluster pour la tol?rance ? l'ombrage
rownames(TolS)=row.names(traits)

# Ward Hierarchical Clustering
d_TolS <- dist(TolS, method = "euclidean") # distance matrix
fit_TolS <- hclust(d_TolS, method="ward.D2")

# Plot dendrogram
plot(fit_TolS, cex=0.7,
     main = " Shade Tolerance Groups ")
# Add red boxes to separate groups
rect.hclust(fit_TolS, 3)

```

## Shade Tolerance Groups



Let's see how many neighbours of each shade tolerance group there are at the neighborhood of each target species

```
# Get the shade clusters for each species
shade_groups <- data.frame(CodeSp= levels(targets$CodeSp),shade=cutree(fit_TolS, k=3))
shade_groups$shade <- as.character(shade_groups$shade)
targets<- left_join(targets, shade_groups, by=c("CodeSp"))
neighbours <- left_join(neighbours, shade_groups, by=c("CodeSp_NEAR"="CodeSp"))

targets$shade <- as.factor(targets$shade)
neighbours$shade <- as.factor(neighbours$shade)

shade_neighbours <- table(neighbours$CodeSp_IN, neighbours$shade)
kable(addmargins(shade_neighbours,2), format="markdown")
```

	1	2	3	Sum
ABBA	18801	3879	2724	25404
ACPL	11585	2505	1806	15896
ACRU	5639	22812	2269	30720
ACSA	21843	5474	4126	31443
BEAL	4805	20140	2252	27197
BEPA	7215	8847	17371	33433
LADE	790	1106	7889	9785
LALA	8090	5979	17334	31403
PIAB	11536	2693	2939	17168
PIGL	22321	7452	6787	36560



	1	2	3	Sum
PIOM	7089	863	933	8885
PIRE	6945	5387	15521	27853
PIRU	16785	4052	6570	27407
PIST	8463	24427	9446	42336
PISY	4024	2941	11652	18617
QURO	2741	8536	1874	13151
QURU	2324	18645	5459	26428
THOC	6947	20306	3324	30577
TICO	8726	2731	773	12230