Example analysis of biodiversity survey data with R package `gradientForest`

C. Roland Pitcher, Nick Ellis, Stephen J. Smith

February 10, 2011

Contents

1 Introduction

R package `gradientForest` [Ellis et al., 2010] develops flexible non-parametric functions to quantify multi-species compositional turnover along environmental gradients. The flexibility comes from the method's origins in Random Forests [Breiman, 2001]; specifically R package randomForest [Liaw and Wiener, 2002]. This document provides an example to demonstrate the use of `gradientForest` for ecological analysis of biodiversity survey data. A multi-regional application is provided by [Pitcher et al., 2010]. The document assumes some familiarity both with R and with community analysis. The example has some analogues with constrained ordination methods and with Generalized Dissimilarity Modelling [Ferrier et al., 2007], which are both complementary. Package randomForest includes functions for plotting non-linear responses in compositional along environmental gradients, and for using these responses to transform environmental data layers to biological scales. The transformed multi-dimensional biological space can be represented as a biplot and can be mapped in geographic space.

This example demonstrates typical scripts for a gradient forest analysis and provides in the package a sites-by-species (row-by-columns) matrix and a matching sites-by-environment (row-by-columns) dataframe. The number of rows and their order must match between these two data objects. The data should not include NAs. It is assumed that users will be familiar with the data-processing steps necessary to produce such data objects.

2 Gradient Forest basics

2.1 Data

The example data provided in package `gradientForest` are real species data from a cross-shelf survey in the far northern Great Barrier Reef of northeast Australia [Poiner et al., 1998, Burridge et al., 2006]. Of $> 1,000$ species observed, a subset of 110 are included from 197 of the sites sampled. The environmental data include 28 predictors, either measured at each site or attributed to each site by interpolation [Pitcher et al., 2002].

```
> require(gradientForest)
> load("GZ.sps.mat.Rdata")
> dim(Sp_mat)
```

```
[1] 197 110
```

```
> load("GZ.phys.site.Rdata")
> dim(Phys_site)
```

```
[1] 197  28
```

2.2 `gradientForest` analysis

The function `gradientForest` is a wrapper function that calls `extendedForest`, a modified version of `randomForest`, and collates its output across all the species in the data matrix. The key modification in `extendedForest` extracts the numerous tree split values along each predictor gradient and their associated fit improvement, for each predictor in each tree, for the forests and returns that information to `gradientForest`.

Like `randomForest`, `extendedForest` assesses the importance of each variable for prediction accuracy; information that is further collated and processed by `gradientForest`. Often, predictor variables are correlated however. The standard approach in random forests assesses marginal importance of predictor by randomly permuting each predictor in turn, across all sites in the dataset, and calculating the degradation prediction performance of each tree. Package extendedForest can account for correlated predictors by implementing conditional permutation [Ellis et al., 2010], following the strategy outlined by Strobl et al. [2008]. In conditional permutation, the predictor to be assessed is permuted only within blocks of the dataset defined by splits in the given tree on any other predictors correlated above a certain threshold (e.g. $r = 0.5$) and up to a maximum number of splits set by the `maxLevel` option (if required).

```
> nSites <- dim(Sp_mat)[1]
> nSpecs <- dim(Sp_mat)[2]
> lev <- floor(log2(nSites * 0.368/2))
> lev
```

```
[1] 5
```

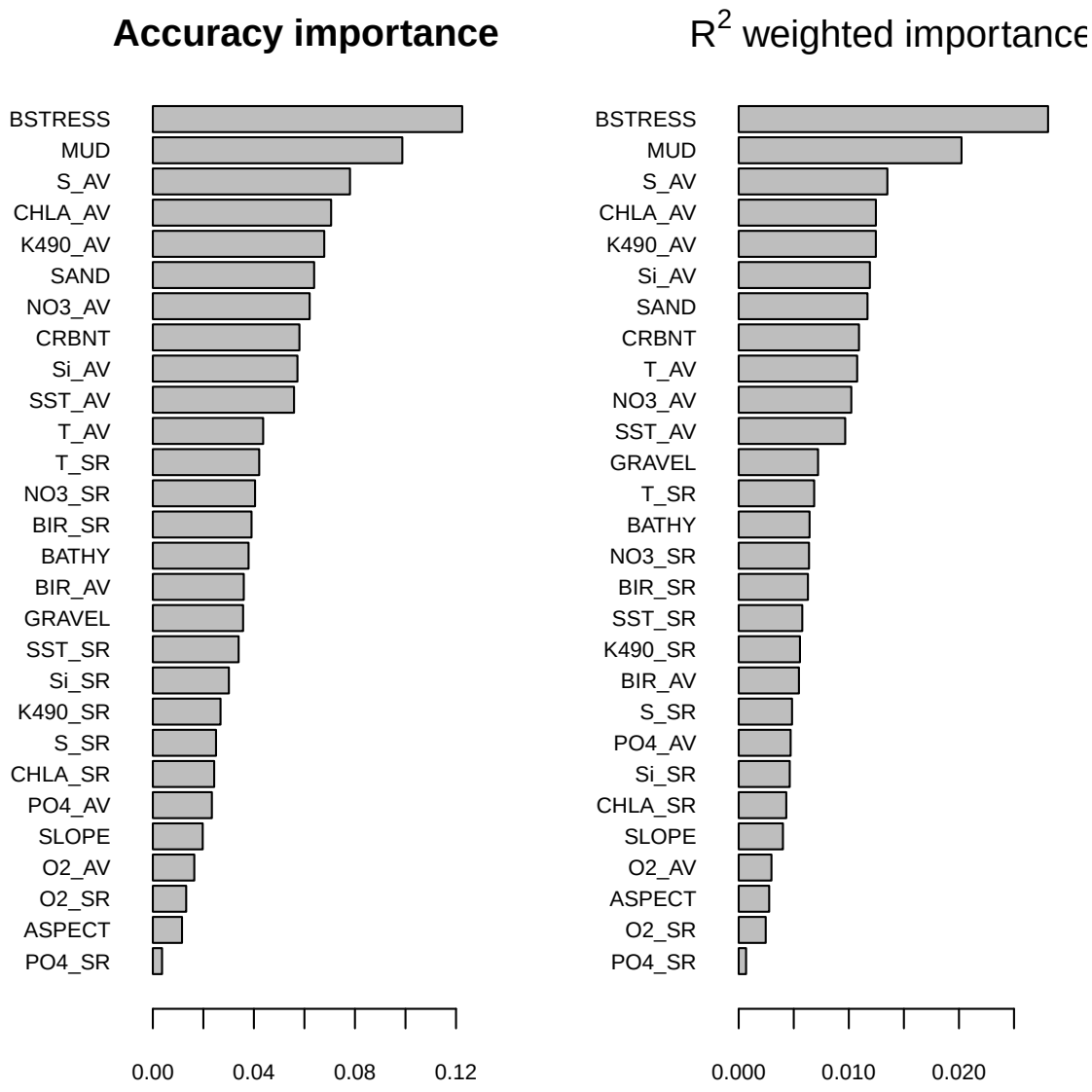
The `gradientForest` may take several minutes to run. Other options that can be set include the number of trees typically 500, whether the splits should be compact into bins (advising to prevent memory problems for large datasets) and the number of bins, and the correlation threshold for conditional permutation. The summary shows the number of species with positive R^2 ie. those species that could be predicted to any extent by the available predictor. The returned object is a list containing the data, predictor importances, species R^2's and other information described in the html help pages under `Value`.

```
> gf <- gradientForest(cbind(Phys_site, Sp_mat),
+     predictor.vars = colnames(Phys_site), response.vars = colnames(Sp_mat),
+     ntree = 500, transform = NULL, compact = T,
+     nbin = 201, maxLevel = lev, corr.threshold = 0.5)

> gf

A forest of 500 regression trees for each of 90 species

Call:

gradientForest(data = cbind(Phys_site, Sp_mat), predictor.vars = colnames(Phys_site),
    response.vars = colnames(Sp_mat), ntree = 500, transform = NULL,
    maxLevel = lev, corr.threshold = 0.5, compact = T, nbin = 201)



Important variables:
[1] BSTRESS MUD     S_AV    CHLA_AV K490_AV

> names(gf)

 [1] "X"               "Y"               "result"
 [4] "overall.imp"     "overall.imp2"    "ntree"
 [7] "imp.rsq"         "species.pos.rsq" "ranForest.type"
[10] "res"             "res.u"           "dens"
[13] "call"
```

2.3 gradientForest plots

Several types of plots are available for the `gradientForest` object. The first is the predictor overall importance plot. This show the mean accuracy importance and the mean importance weighted by species R^2. in this example, both are conditional importance. Seabed stress and sediment mud fraction are clearly the most important variables across these 89 species.

```
> plot(gf, plot.type = "O")
```

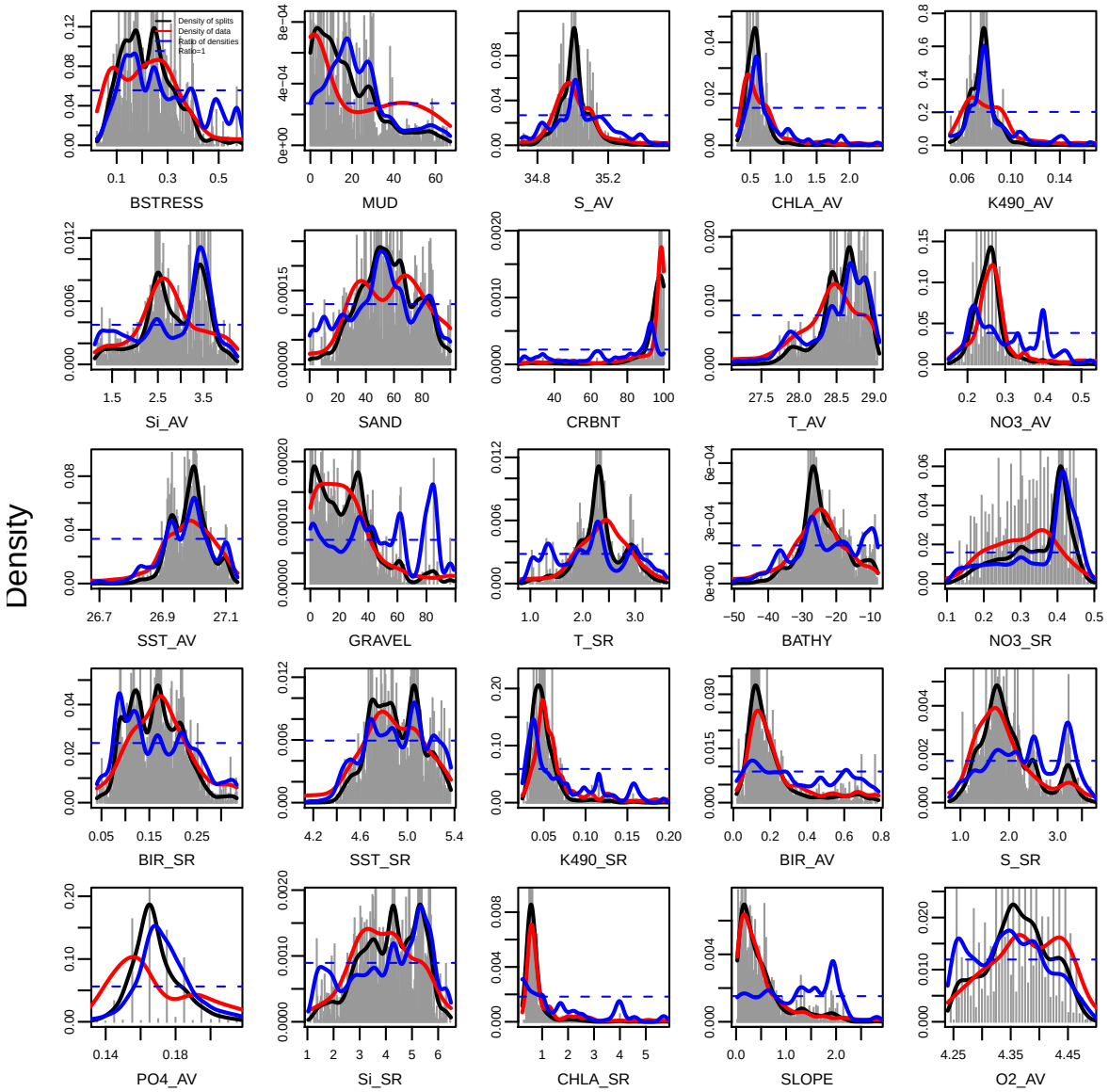


The predictor gradient plots are best presented in order of importance; in this example the top 25 predictors are presented in 5 by 5 panels.

```
> most_important <- names(importance(gf))[1:25]
> par(mgp = c(2, 0.75, 0))
```

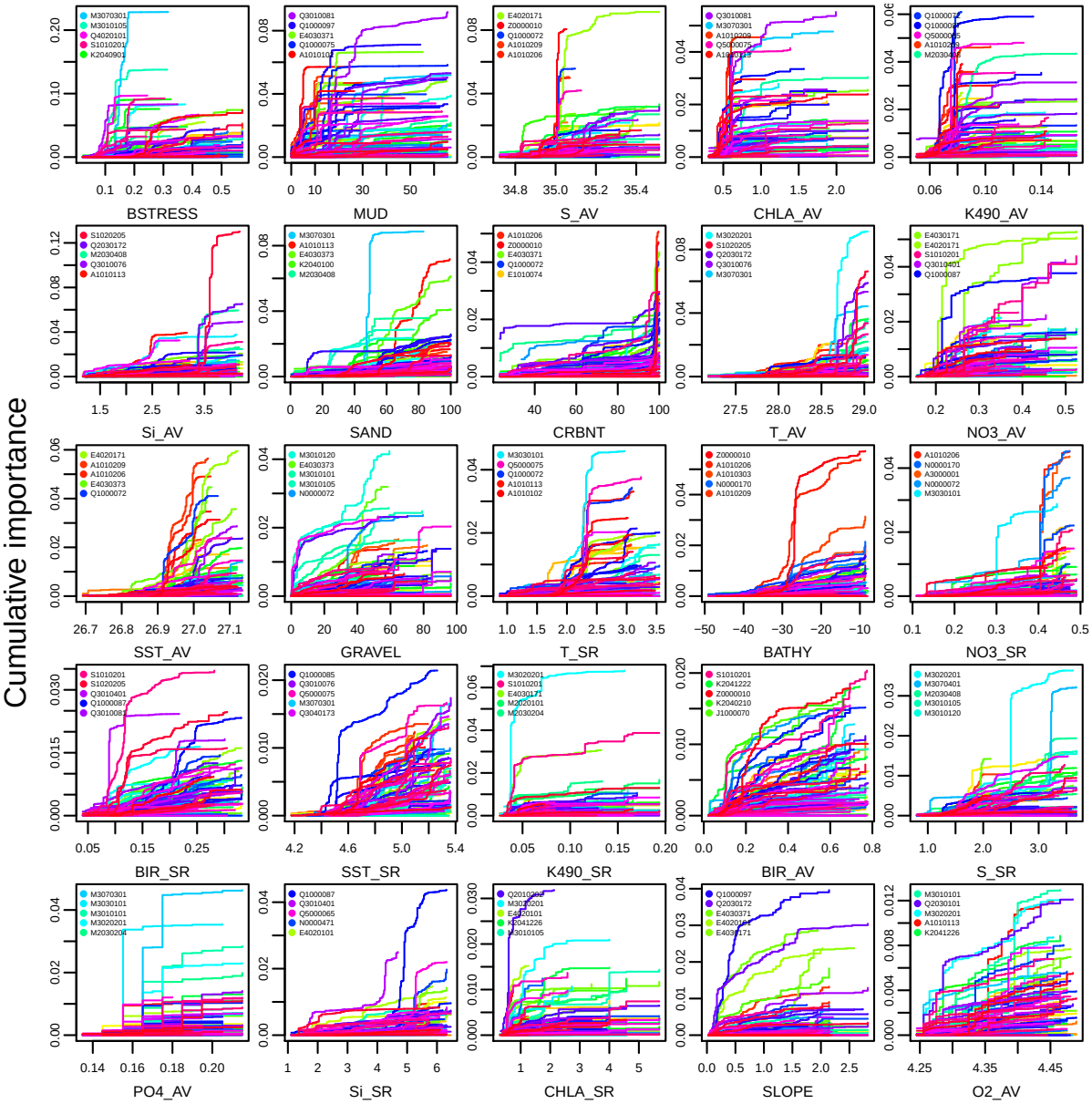
The second plot is the splits density plot (`plot.type="S"`), which shows binned split importance and location on each gradient (spikes), kernel density of splits (*black lines*), of observations (*red lines*) and of splits standardised by observations density (*blue lines*). Each distribution integrates to predictor importance. These show where important changes in the abundance of multiple species are occurring along the gradient; they indicate a composition change rate. Many of the usual plot options can be set in the call.

```
> plot(gf, plot.type = "S", imp.vars = most_important,
+     leg.posn = "topright", cex.legend = 0.4, cex.axis = 0.6,
+     cex.lab = 0.7, line.ylab = 0.9, par.args = list(mgp = c(1.5,
+         0.5, 0), mar = c(3.1, 1.5, 0.1, 1)))
```



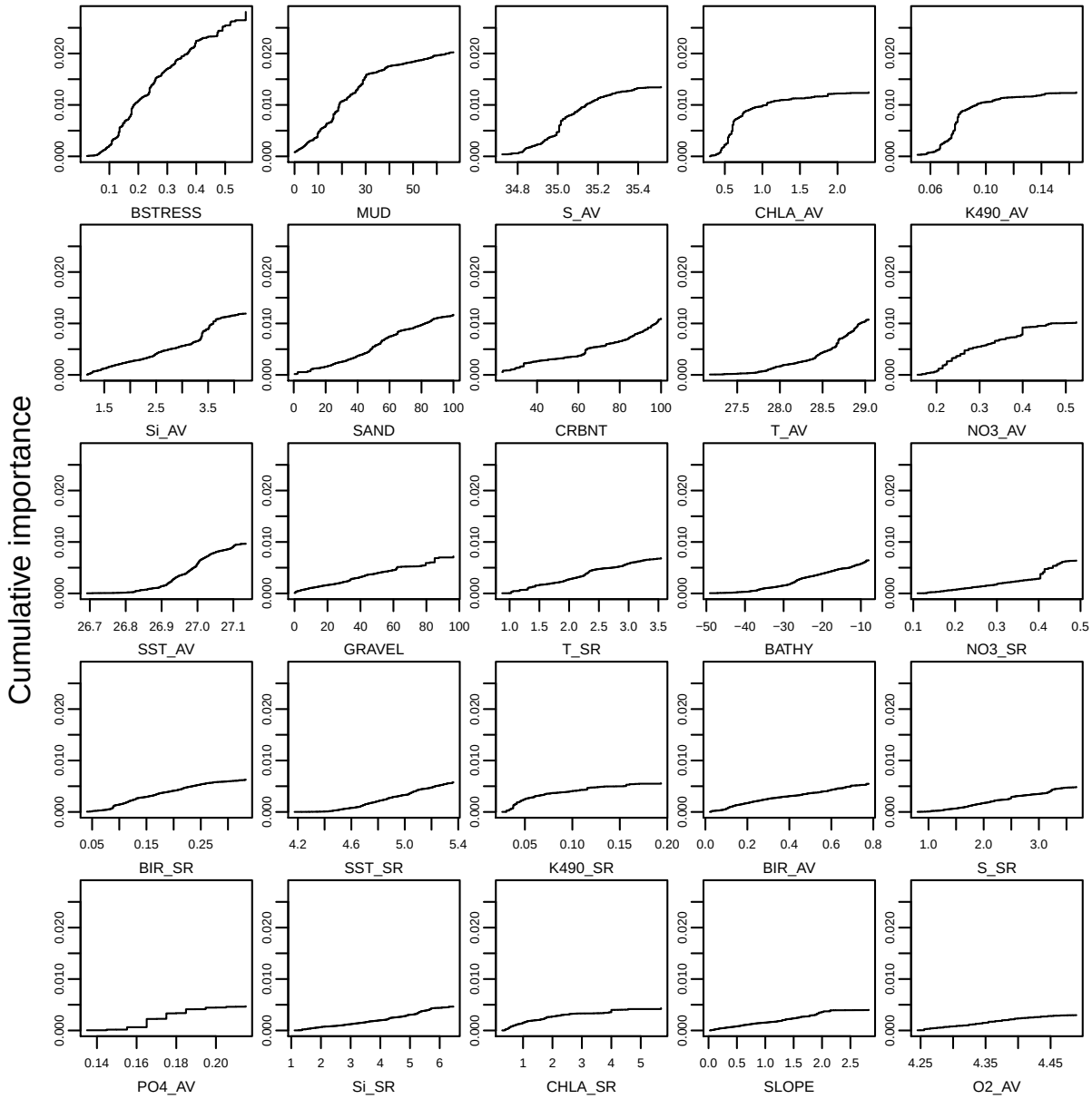
The third plot is the species cumulative plot (`plot.type="C"`, `show.overall=F`), which for each species shows cumulative importance distributions of splits improvement scaled by R^2 weighted importance, and standardised by density of observations. These show cumulative change in abundance of individual species, where changes occur on the gradient, and the species changing most on each gradient. Again many of the usual plot options can be set in the call; in this example the legend identifies the top 5 most responsive species for each predictor

```
> plot(gf, plot.type = "C", imp.vars = most_important,
+     show.overall = F, legend = T, leg.posn = "topleft",
+     leg.nspecies = 5, cex.lab = 0.7, cex.legend = 0.4,
+     cex.axis = 0.6, line.ylab = 0.9, par.args = list(mgp = c(1.5,
+         0.5, 0), mar = c(2.5, 1, 0.1, 0.5), omi = c(0,
+         0.3, 0, 0)))
```



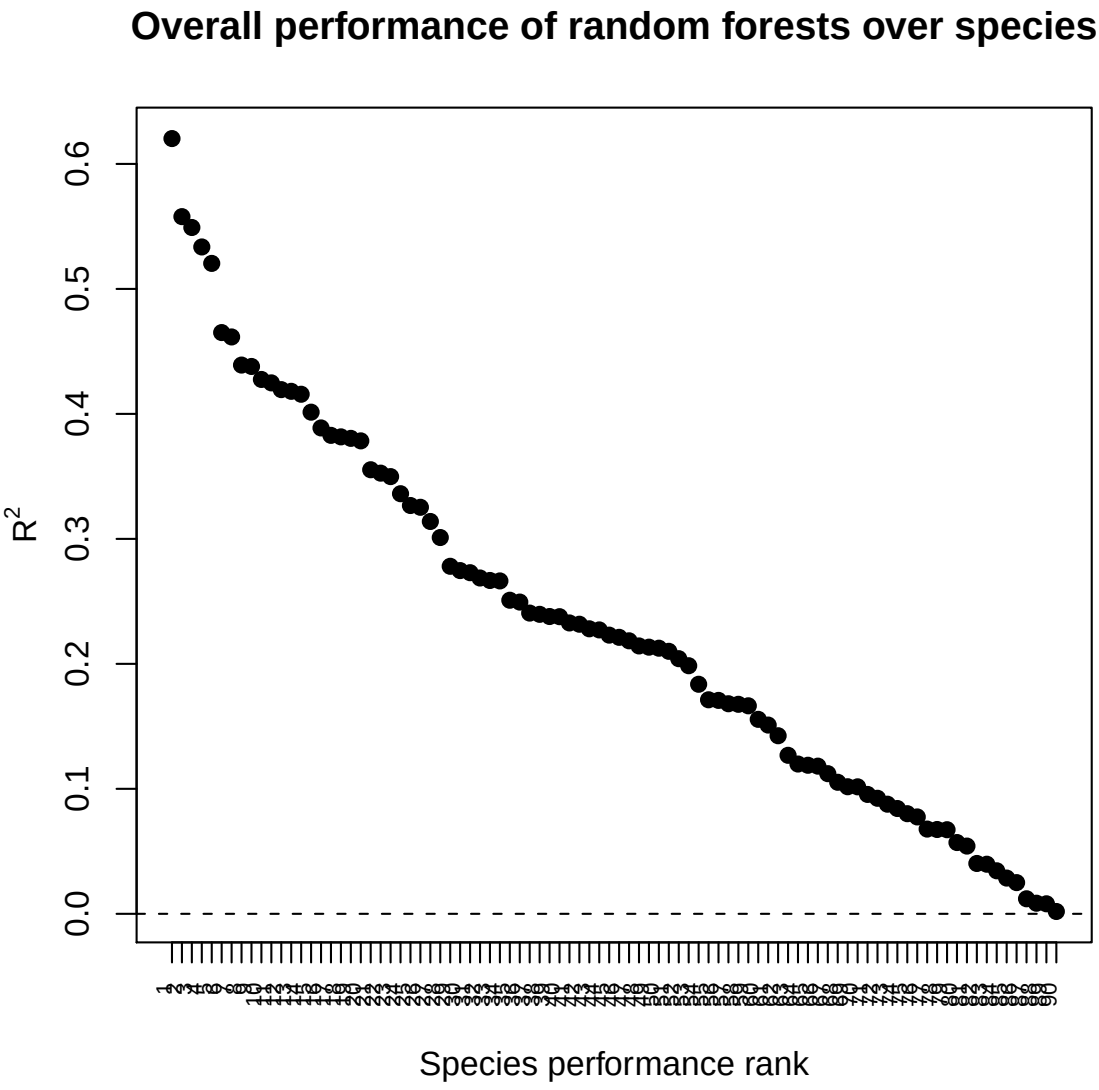
The fourth plot is the predictor cumulative plot (`plot.type="C"`, `show.species=F`), which for each predictor shows cumulative importance distributions of splits improvement scaled by R^2 weighted importance, and standardised by density of observations, averaged over all species. These show cumulative change in overall composition of the community, where changes occur on the gradient. Again many of the usual plot options can be set in the call; in this example `common.scale=T` ensures that plots for all predictors have the same y-scale as the most important predictor.

```
> plot(gf, plot.type = "C", imp.vars = most_important,
+     show.species = F, common.scale = T, cex.axis = 0.6,
+     cex.lab = 0.7, line.ylab = 0.9, par.args = list(mgp = c(1.5,
+         0.5, 0), mar = c(2.5, 1, 0.1, 0.5), omi = c(0,
+         0.3, 0, 0)))
```



The fifth plot shows the R^2 measure of the fit of the random forest model for each species, ordered in various ways.

```
> plot(gf, plot.type = "P", show.names = F, horizontal = F,
+     cex.axis = 1, cex.labels = 0.7, line = 2.5)
```



Several other alternative formats of the R^2 fit performance plot are available, e.g.:

```
> plot(gf, plot.type = "P", show.names = T, horizontal = F,
+     cex.axis = 1, cex.labels = 0.7, line = 2.5)
> plot(gf, plot.type = "P", show.names = F, horizontal = T,
+     cex.axis = 1, cex.labels = 0.6, line = 2.5)
> plot(gf, plot.type = "P", show.names = T, horizontal = T,
+     cex.axis = 1, cex.labels = 0.6, line = 2.5)
```

3 Gradient Forest predictions

In addition to examining compositional change along environmental gradients, the predictor cumulative functions can also be used to transform grid data layers of environmental variables to a common biological importance scale. This transformation of the multi-dimensional environment space is to a biological space in which coordinate position represents composition

associated with the predictors. These inferred compositional patterns can be mapped in biological space and geographic space in a manner analogous to ordination, but takes into account the non-linear and sometimes threshold changes that occur along gradients.

3.1 Transforming predictors

The example provided includes gridded environmental variables for a roughly 10,000 km^2 area of the far northern Great Barrier Reef where the biological surveys were conducted. The data include North and East coordinates plus 28 predictors at 8,682 grid cells. The grid data must include the same predictors with the same names as sites included in the `gradientForest` call.

```
> load("GZ.phys.grid.Rdata")

> dim(Phys_grid)

[1] 8682   30

> names(Phys_grid)

 [1] "NORTH"   "EAST"    "BATHY"   "SLOPE"   "ASPECT"
 [6] "BSTRESS" "CRBNT"   "GRAVEL"  "SAND"    "MUD"
[11] "NO3_AV"  "NO3_SR"  "PO4_AV"  "PO4_SR"  "O2_AV"
[16] "O2_SR"   "S_AV"    "S_SR"    "T_AV"    "T_SR"
[21] "Si_AV"   "Si_SR"   "CHLA_AV" "CHLA_SR" "K490_AV"
[26] "K490_SR" "SST_AV"  "SST_SR"  "BIR_AV"  "BIR_SR"
```

The grid variables are transformed using the `gradientForest` `predict` function.

```
> imp.vars <- names(importance(gf))
> Trns_grid <- cbind(Phys_grid[, c("EAST", "NORTH")],
+     predict(gf, Phys_grid[, imp.vars]))
```

It is useful to also transform the site environmental predictors, which are available from `gf$X`.

```
> Trns_site <- predict(gf)
```

3.2 Biplot of the biological space

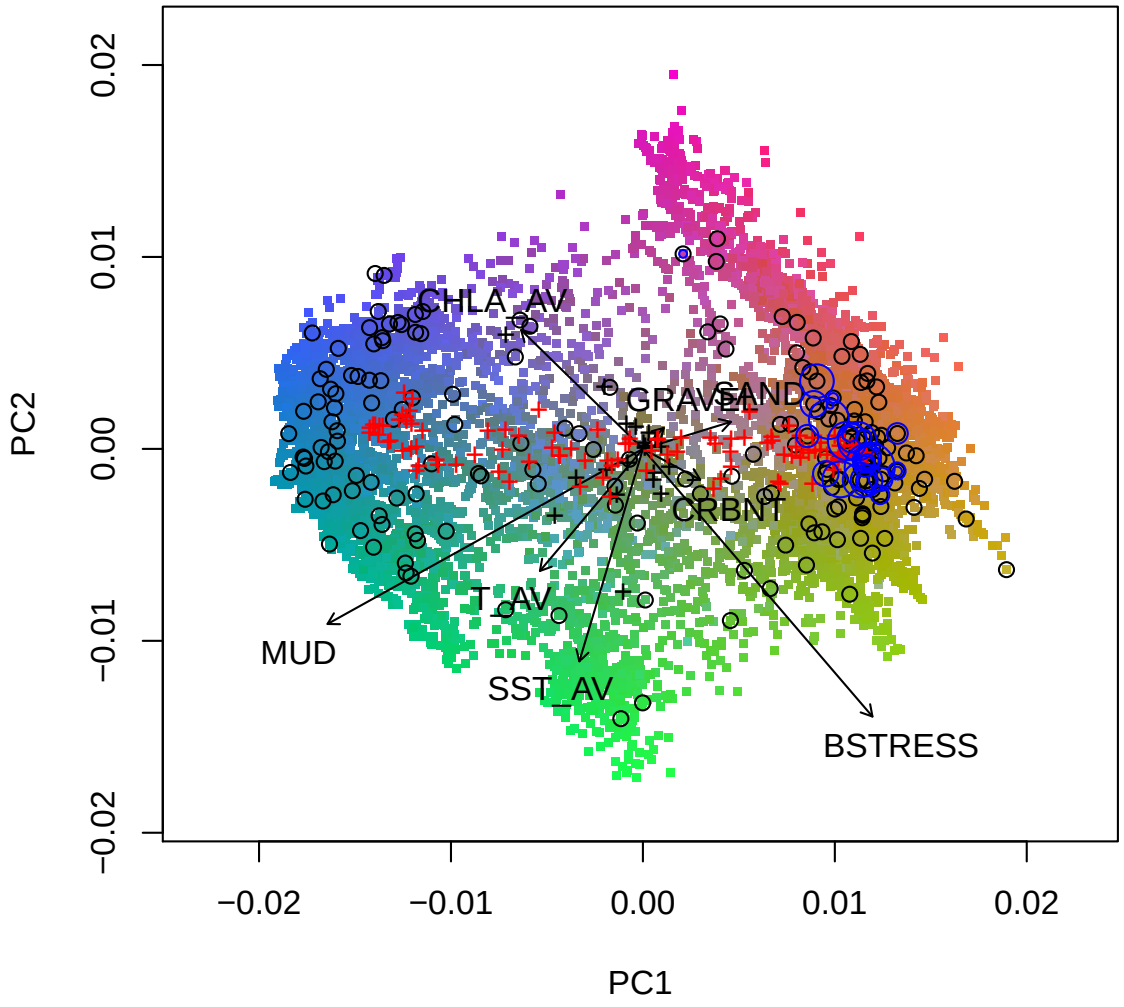
The multi-dimensional biological space can most effectively be represented by taking the principle components of the transformed grid and presenting the first two dimensions in a biplot. It must be acknowledged that while most variation in patterns is captured by the first dimensions, additional compositional pattern contained in the higher dimensions is not shown. A user defined RGB colour palette is set up based on the first 3 dimensions.

```
> PCs <- prcomp(Trns_grid[, imp.vars])
> a1 <- PCs$x[, 1]
> a2 <- PCs$x[, 2]
> a3 <- PCs$x[, 3]
> r <- a1 + a2
> g <- -a2
> b <- a3 + a2 - a1
> r <- (r - min(r))/(max(r) - min(r)) * 255
> g <- (g - min(g))/(max(g) - min(g)) * 255
> b <- (b - min(b))/(max(b) - min(b)) * 255
```

The environmental variables may be shown as vectors, perhaps limited to the most important predictors — in this example, variables to show as vectors are selected.

```
> nvs <- dim(PCs$rotation)[1]
> vec <- c("BSTRESS", "MUD", "SST_AV", "T_AV", "CHLA_AV",
+     "SAND", "CRBNT", "GRAVEL")
> lv <- length(vec)
> vind <- rownames(PCs$rotation) %in% vec
> scal <- 40
> xrng <- range(PCs$x[, 1], PCs$rotation[, 1]/scal) *
+     1.1
> yrng <- range(PCs$x[, 2], PCs$rotation[, 2]/scal) *
+     1.1
> plot((PCs$x[, 1:2]), xlim = xrng, ylim = yrng,
+     pch = ".", cex = 4, col = rgb(r, g, b, max = 255),
+     asp = 1)
> points(PCs$rotation[!vind, 1:2]/scal, pch = "+")
> arrows(rep(0, lv), rep(0, lv), PCs$rotation[vec,
+     1]/scal, PCs$rotation[vec, 2]/scal, length = 0.0625)
> jit <- 0.0015
> text(PCs$rotation[vec, 1]/scal + jit * sign(PCs$rotation[vec,
+     1]), PCs$rotation[vec, 2]/scal + jit * sign(PCs$rotation[vec,
+     2]), labels = vec)
```

Different coordinate positions in the biplot represent differing compositions, as associated with the predictors. Further information may be added to the biplot including the location of sites in biological space, the weight mean location of species, and selected species may be identified interactively.




```
> PCsites <- predict(PCs, Trns_site[, imp.vars])
> points(PCsites[, 1:2])
> SpsWtd <- sweep(gf$Y, 2, apply(gf$Y, 2, min),
+     "-")
> SpsWtdPCs <- (t(SpsWtd) %*% (PCsites[, 1:2]))/colSums(SpsWtd)
> points(SpsWtdPCs, col = "red", pch = "+")
```

If required, the abundance of any given species may be plotted on the biplot. For example the first species from `gf$Y` = `A1010102`, an alga from the family Caulerpaceae that appears to prefer carbonate gravelly sand area with moderate bedstress and lower temperature.

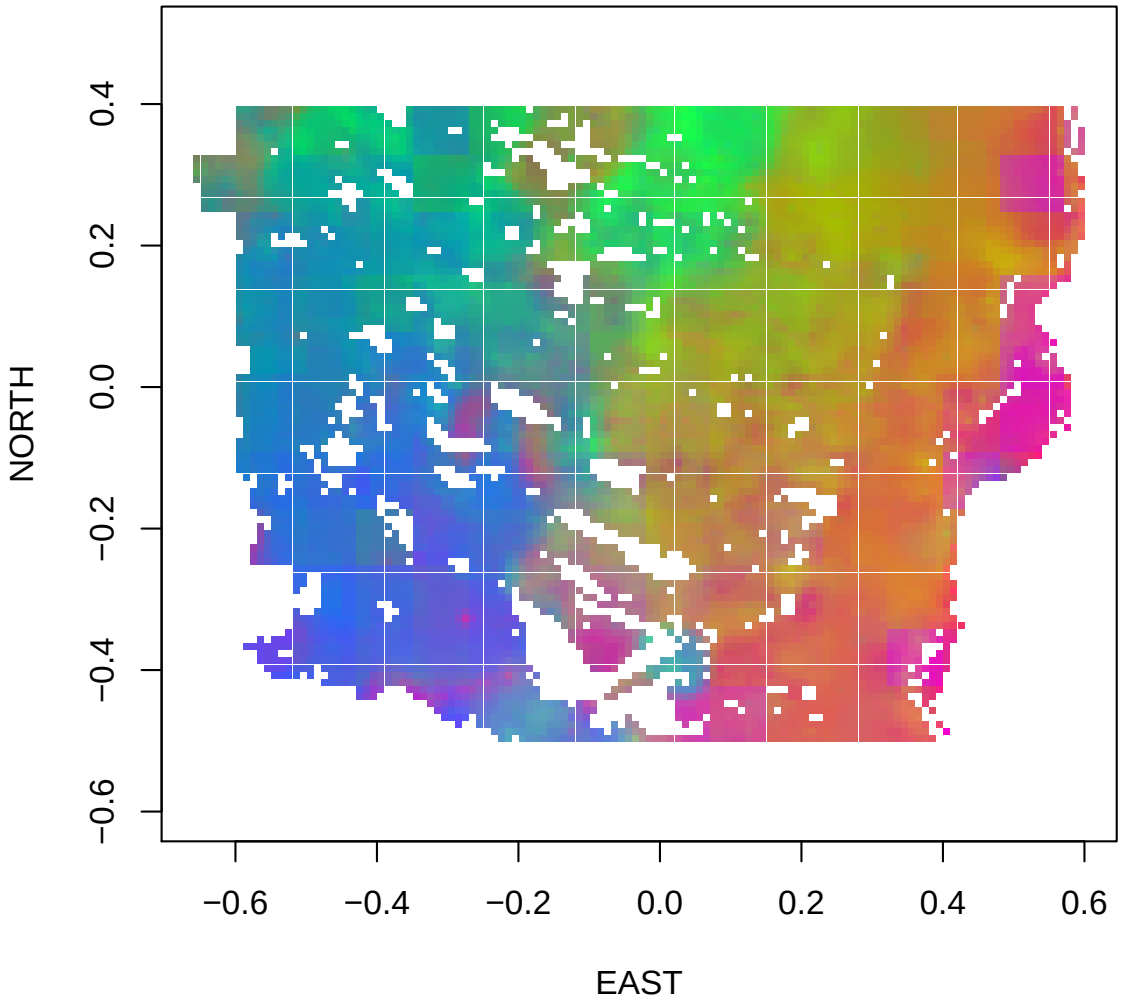
```
> sp <- colnames(SpsWtd)[1]
> points(PCsites[, 1:2], col = "blue", cex = SpsWtd[,
+     sp]/2)
```

Alternatively, specifically named examples could be plotted: e.g. `E4030373` a Fungiid coral; `M2020101` a Strombid mollusc; or `S1010671` a Perophorid ascidian to name a few.

3.3 Mapping in geographic space

The biplot and the colour palette in the previous section can be used as a key to visualise compositional patterns mapped in geographic space. The following map plots predicted PC scores in geographic coordinates, using the same colour palette as above, and represents continuous changes in inferred compositional patterns associated with the predictors.

```
> plot(Trns_grid[, c("EAST", "NORTH")], pch = ".",
+      cex = 3, asp = 1, col = rgb(r, g, b, max = 255))
```

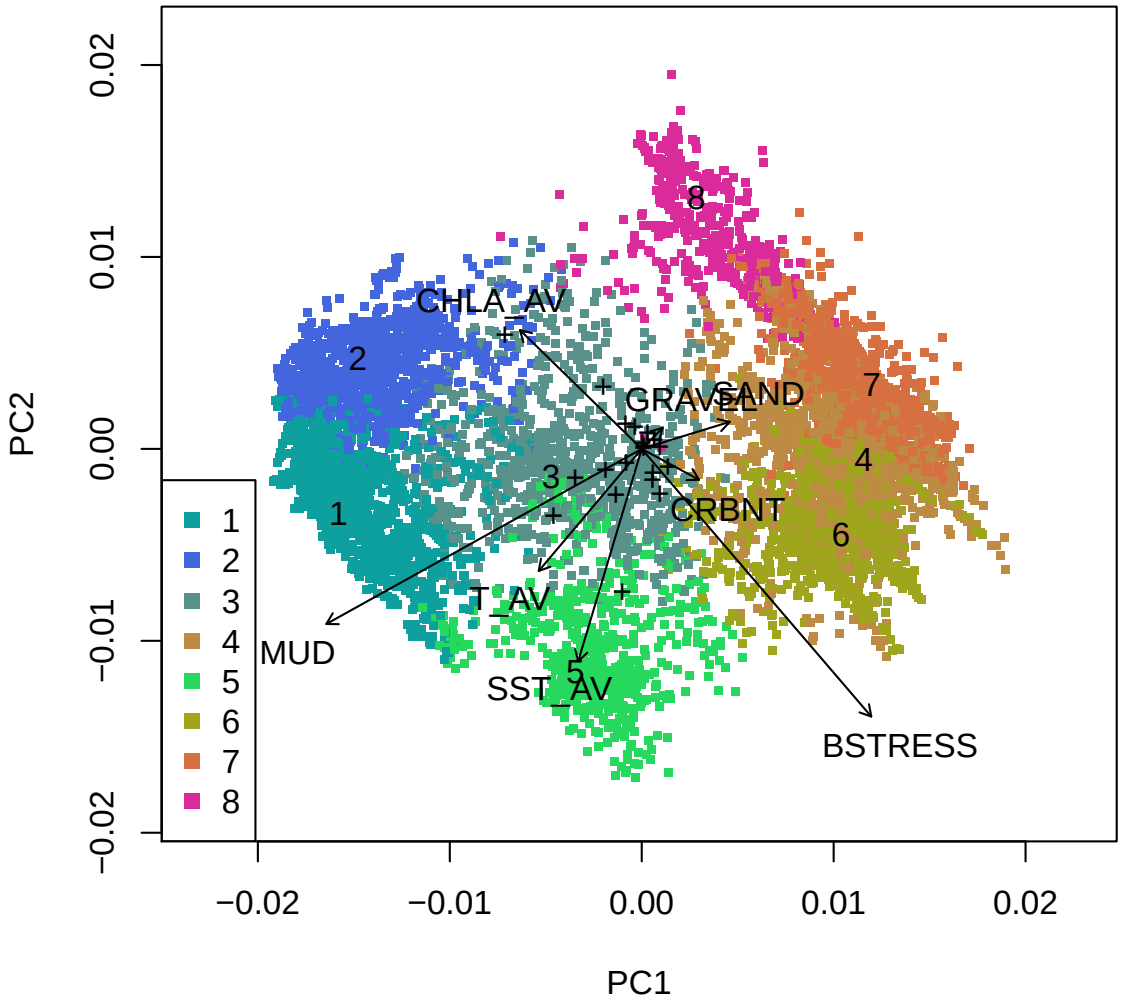


3.4 A clustered version

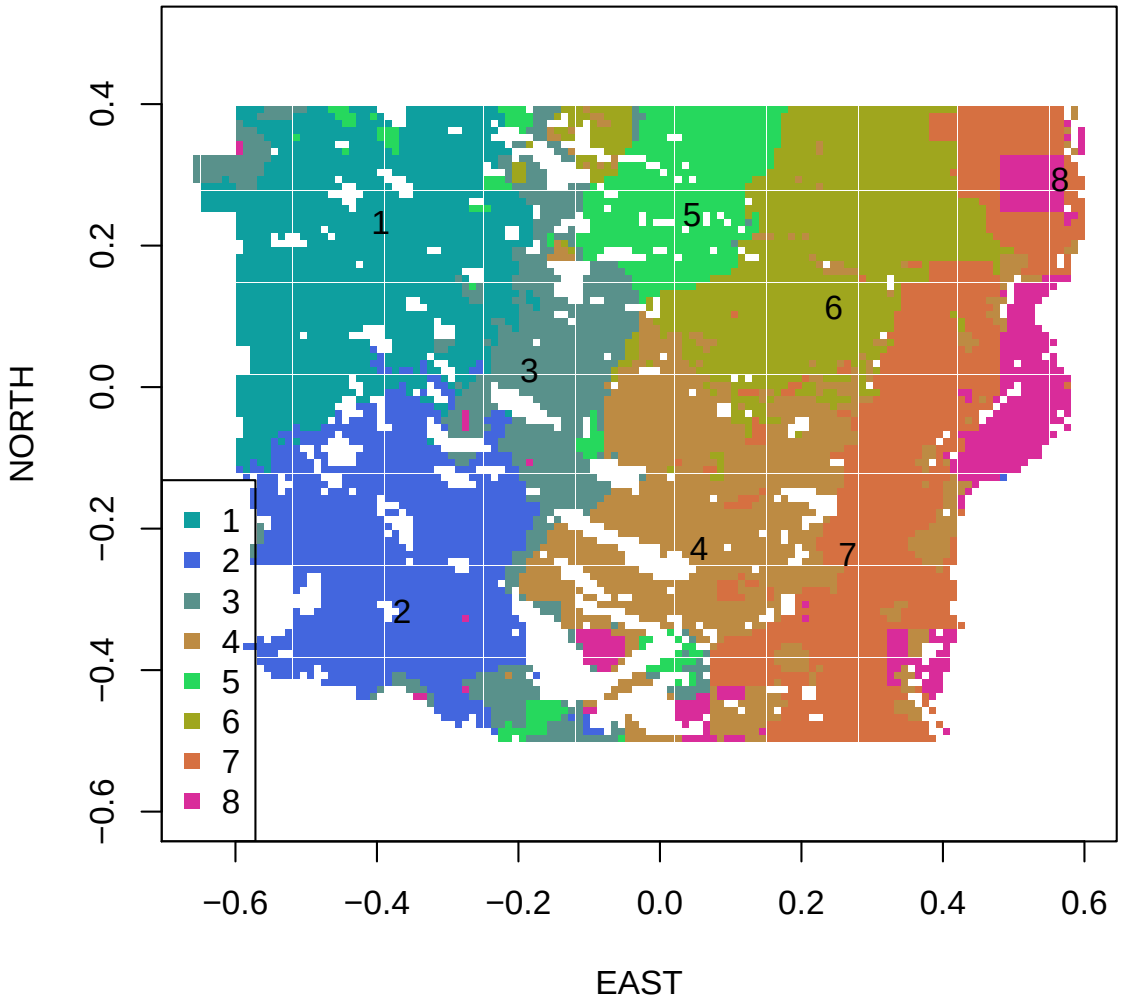
Some applications may require a hard clustered output, representing inferred assemblages, rather than a continuous representation of biodiversity composition. The following example uses `clara` to make 8 clusters. This is a fast claustering algorithm suitable for large data sets. The medoids are labelled and the colour key takes the value for each medoid. Other clustering methods may

be used (for example, `pam` would take several minutes) as alternatives, and their various cluster diagnostics may provide a guide to the appropriate numbers of clusters.

```
> require(cluster)
> ncl <- 8
> clPCs <- clara(PCs$x, ncl, sampsize = 1000)
> medcolR <- r[clPCs$i.med]
> medcolG <- g[clPCs$i.med]
> medcolB <- b[clPCs$i.med]
> plot((PCs$x[, 1:2]), xlim = xrng, ylim = yrng,
+     pch = ".", cex = 4, col = rgb(medcolR[clPCs$clustering],
+         medcolG[clPCs$clustering], medcolB[clPCs$clustering],
+         max = 255), asp = 1)
> points(PCs$rotation[!vind, 1:2]/scal, pch = "+")
> arrows(rep(0, lv), rep(0, lv), PCs$rotation[vec,
+     1]/scal, PCs$rotation[vec, 2]/scal, length = 0.0625)
> text(PCs$rotation[vec, 1]/scal + jit * sign(PCs$rotation[vec,
+     1]), PCs$rotation[vec, 2]/scal + jit * sign(PCs$rotation[vec,
+     2]), labels = vec)
> text(clPCs$medoids[, 1:2], labels = seq(1, ncl))
> legend("bottomleft", as.character(seq(1, ncl)),
+     pch = 15, cex = 1, col = rgb(medcolR, medcolG,
+         medcolB, max = 255))
```




```
> plot(Trns_grid[, c("EAST", "NORTH")], pch = ".",
+     cex = 3, asp = 1, col = rgb(medcolR[clPCs$clustering],
+         medcolG[clPCs$clustering], medcolB[clPCs$clustering],
+         max = 255))
> points(Trns_grid[clPCs$i.med, c("EAST", "NORTH")],
+     pch = as.character(seq(1, ncl)))
> legend("bottomleft", as.character(seq(1, ncl)),
+     pch = 15, cex = 1, col = rgb(medcolR, medcolG,
+         medcolB, max = 255))
```



4 Session information

The simulation and output in this document were generated in the following computing environment.

- R version 2.10.1 (2009-12-14), `i386-pc-mingw32`
- Locale: `LC_COLLATE=English_Australia.1252`, `LC_CTYPE=English_Australia.1252`, `LC_MONETARY=English_Australia.1252`, `LC_NUMERIC=C`, `LC_TIME=English_Australia.1252`
- Base packages: base, datasets, graphics, grDevices, methods, stats, tools, utils
- Other packages: cluster 1.12.1, extendedForest 1.5, gradientForest 0.1-11, lattice 0.18-3
- Loaded via a namespace (and not attached): grid 2.10.1

References

L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

C.Y. Burridge, C.R. Pitcher, B.J. Hill, T.J. Wassenberg, and I.R. Poiner. A comparison of demersal communities in an area closed to trawling with those in adjacent areas open to trawling: a study in the Great Barrier Reef Marine Park, Australia. *Fisheries Research*, 79: 64–74, 2006.

N. Ellis, S.J. Smith, and C.R. Pitcher. Gradient forests: calculating importance gradients on physical predictors. submitted manuscript. 2010.

S. Ferrier, G. Manion, J. Elith, and K. Richardson. Using generalized dissimilarity modelling to analyse and predict patterns of beta diversity in regional biodiversity assessment. *Diversity and Distributions*, 13(3):252–264, 2007.

Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2(3): 18–22, 2002. URL `http://CRAN.R-project.org/doc/Rnews/`.

C.R. Pitcher, W. Venables, N. Ellis, I. McLeod, M. Cappo, F. Pantus, M. Austin, P. Doherty, and N. Gribble. Gbr seabed biodiversity mapping project: Phase 1 report to crc-reef. Technical report, CSIRO/AIMS/QDPI Report, 2002. URL `http://www.reef.crc.org.au/resprogram/programC/seabed/Seabedphase1rpt.htm`.

C.R. Pitcher, P. Lawton, N. Ellis, S.J. Smith, L.S. Incze, C-L. Wei, M.E. Greenlaw, N.H. Wolff, J. Sameoto, and P.V.R. Snelgrove. The role of physical environmental variables in shaping patterns of biodiversity composition in seabed assemblages. submitted manuscript. 2010.

IR Poiner, J. Glaister, CR Pitcher, C. Burridge, T. Wassenberg, N. Gribble, B. Hill, SJM Blaber, DA Milton, D. Brewer, et al. The environmental effects of prawn trawling in the far northern section of the Great Barrier Reef Marine Park: 1991–1996. Final Report to GBRMPA and FRDC, 1998. URL `http://www.publish.csiro.au/books/bookpage.cfm?PID=2419`.

C. Strobl, A.L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis. Conditional variable importance for random forests. *BMC Bioinformatics*, 9(1):307, 2008.