

# CREATING TREE DISTRIBUTION MAPS IN CTFS PLOTS

R. CONDIT

SHORT\_TITLE: Plot maps END\_SHORT\_TITLE

R. Condit

DESCRIPTION: The map package includes functions for creating maps of tree distributions in CTFS plots. All functions are in a single file, *map.r*. There are functions for drawing single maps, on which one or more species can be included, and one or more dbh classes. And there are functions to create many maps, for example of every species in a plot. The functions have many options for adjusting features (point size, colors), and allow maps to be drawn on the screen or exported as various graphics file types. All functions use input data in the standard `web/data_formatCTFS` format. Most use data in the split format, so having R objects in this format saved already is convenient. END\_DESCRIPTION

This tutorial illustrates use of 4 of the functions, *map*, *pdf.allplot*, *png.allplot*, and *complete.plotmap*

## 1. LOADING DATA

The `web/data_formatutilities` function can be used to attach an existing split data object for a single census. Otherwise, load such an object. In the example below, the object *bci.split6* is attached. It is a list of dataframes, with one dataframe per species; at BCI there are 326 species, and the first 6 are listed below, as 6-letter mnemonics.

```
> CTFSplot(plot = "bci", census = 6, "split")
[1] 4
> length(bci.split6)
[1] 319
> head(names(bci.split6))
[1] "acacme" "acaldi" "acalma" "ade1tr" "aegipa" "alchco"
> dim(bci.split6$acacme)
[1] 24 19
> head(bci.split6$poular)[, 1:10]
```

	treeID	stemID	tag	StemTag	quadrat	gx	gy	MeasureID	CensusID	dbh
27	27	NA	000010	<NA>	4923	997.9	467.3	NA	NA	NA
36	36	6	000019	<NA>	4921	998.5	434.0	15	6	338
293	293	1	000280	a	4819	968.5	394.5	214	6	NA
301	301	NA	000288	<NA>	4820	968.6	403.0	NA	NA	NA
316	316	NA	000303	<NA>	4823	976.7	469.3	NA	NA	NA
328	328	NA	000315	<NA>	4724	950.5	488.9	NA	NA	NA

---

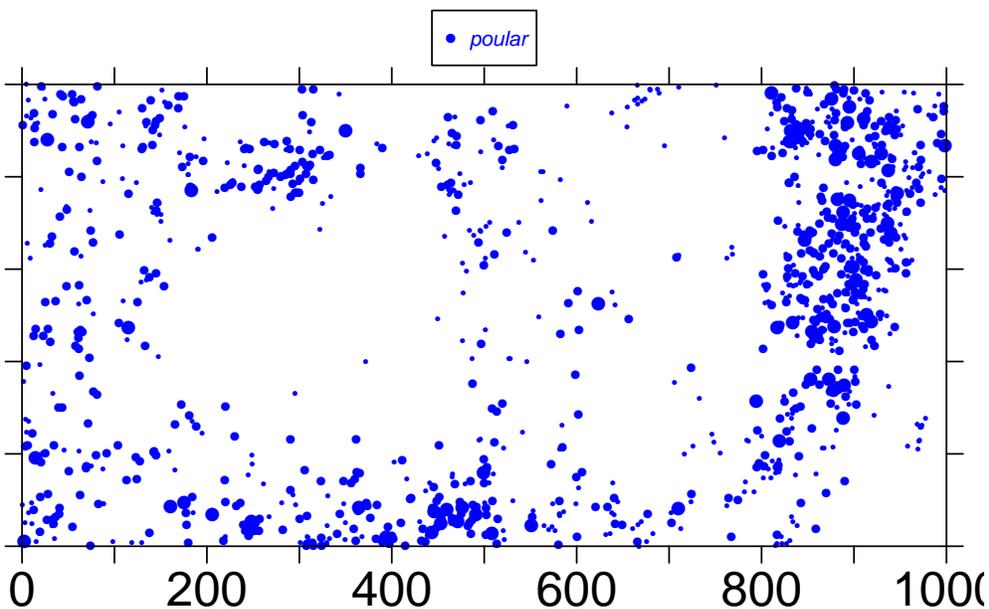
Date: June 17, 2011.

## 2. THE BASIC MAP FUNCTION

This is the core function. It will create a single map of one or more species, with many options available for adjusting the map.

The simplest use requires identifying the split data object to use, here *bci.split6*, and a species mnemonic, which of course must be one of the names in the object. (The default plot size is 1000x500 m, so if your plot is 25 ha, you will have to set *plotdim=c(500,500)*.)

```
> map(splitdatafile = bci.split6, species = "poular")
```



To begin tweaking the map, check the arguments, and also read about them at the [web/data\\_formatmap](#) help page. There are a lot of arguments.

```
> args(map)
```

```
function (splitdatafile, species, spplist = NULL, plotdim = c(1000,
  500), xrange = c(0, plotdim[1]), yrange = c(0, plotdim[2]),
  elevdata = NULL, cutoff = c(10, 100, 300, 3000), size = NULL,
  deadtree = FALSE, maintitle = "", titlepos = c(plotdim[1]/2,
  1.32 * plotdim[2]), clr = NULL, bw = FALSE, bgcolor = "white",
  symbols = NULL, addlegend = TRUE, legpos = c(plotdim[1]/2,
  1.16 * plotdim[2]), legsize = 0.75, ht = 6, wd = 9, plotside = 6,
```

```
export = "no", topoint = 0, filepath = "", outfile = NULL)
NULL
```

To make a larger map, use the arguments *ht* and *wd* to set the overall height and width of the graph (inches), and *plotside* to set the width of plot itself. To use these parameters, you have to set *export*="windows", *export*="mac", or *export*="unix", depending on your operating system. I use the latter.

```
> map(splitdatafile=bci.split6,species='poular',wd=11,ht=8,plotside=10,export='unix')
```

To export a map, use *export*='pdf' and set the output using *filepath* and *outfile*

```
> map(splitdatafile=bci.split6,species='poular',wd=11,ht=8,plotside=10,export='pdf',filepath='maps/
```

### 3. TOPO LINES

The most appealing maps include topo lines. This requires elevation data in matrix format, a standard format for CTFS plots (data for most plots are available at <http://ctfs.arnarb.harvard.edu/Public/Datasets/CTFSTopography/>). I have BCI data in *CTFSElev\_bci*, and below I save the matrix portion as *bci.elev*. The matrix has 201 columns, one every 5 m of the 1000-m plot, and 101 rows for the 500-m axis. The tutorial on topography describes this format (<http://ctfs.arnarb.harvard.edu/Public/CTFSRPackage/index.php/web/tutorials/topography/index.html>).

```
> attach_if_needed("~/ctfs.comparisons/elevation/CTFSElev_bci.rdata")
```

```
[1] 3
```

```
> bci.elev = CTFSElev_bci$mat
```

```
> dim(bci.elev)
```

```
[1] 101 201
```

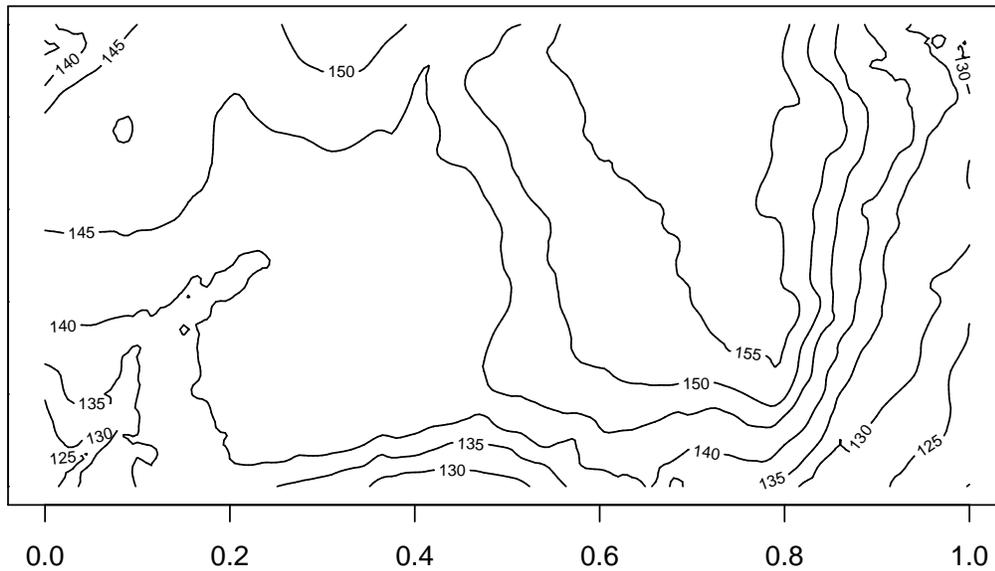
```
> head(bci.elev[, 1:10])
```

```
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9]  [,10]
[1,] 120.63 121.94 123.46 125.07 126.65 127.96 128.88 129.75 130.85 131.81
[2,] 120.87 120.88 122.65 124.18 125.45 126.89 127.99 129.10 130.49 131.62
[3,] 121.12 121.35 122.38 123.52 124.62 125.80 127.19 128.80 130.72 132.07
[4,] 121.36 121.29 121.40 122.00 124.53 125.36 126.33 128.33 130.40 131.73
[5,] 121.29 121.56 121.07 120.98 122.21 123.50 124.67 127.00 128.71 130.05
[6,] 123.45 123.56 123.93 124.27 124.63 124.02 123.94 124.90 125.92 127.62
```

The R function *contours* uses data in this matrix format to map contour lines. Since the plot is 1000x500 m, use *par* to set the size of the figure so that it is twice as wide as long.

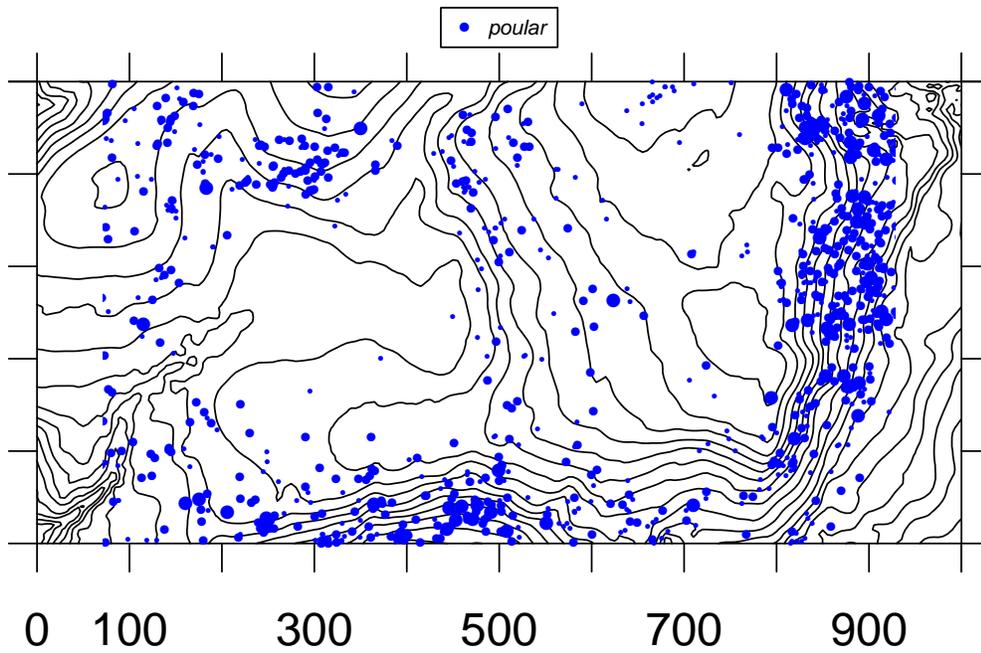
```
> par(pin = c(6, 3))
```

```
> contour(t(bci.elev))
```



The `map` function makes use of `contour` to add topo lines to the species map. Below is the map with default options. Changing the size or exporting would be identical to the examples above.

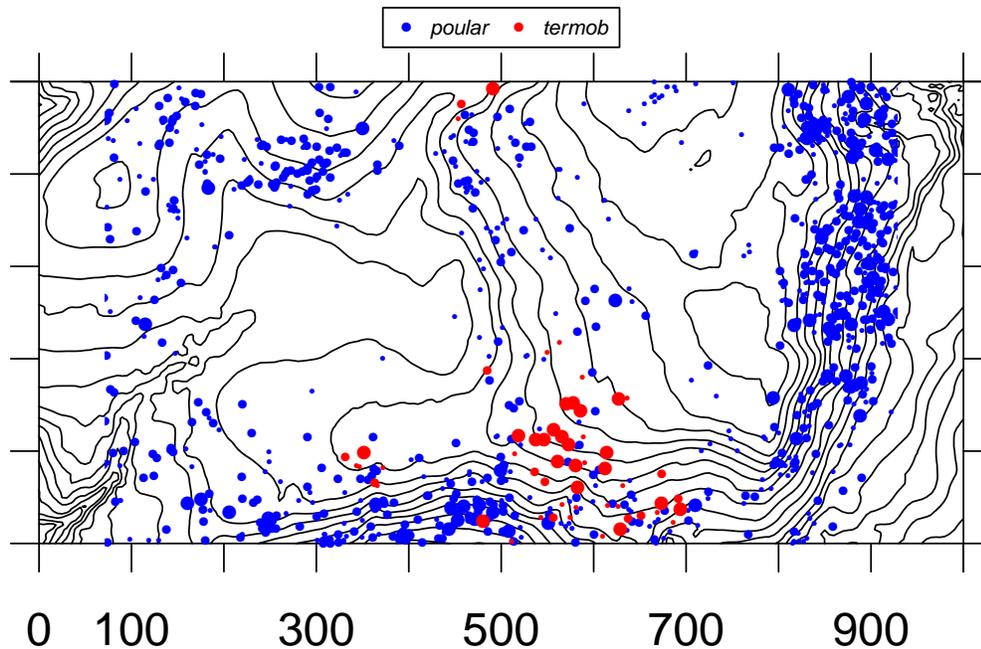
```
> map(splitdatafile = bci.split6, species = "poular", elevdata = bci.elev)
```



#### 4. MULTIPLE SPECIES

Two or more species are mapped by simply adding species names. Check options for color and symbol size and shape to refine.

```
> map(splitdatafile = bci.split6, species = c("poular", "termob"), symbols = c(16, 16))
```



## 5. MULTIPLE MAPS

There is a function `pdf.allplot` that automatically generates a set of pdf maps for all species in a plot. This example uses a table of species names to include Latin names; most CTFS plots have species tables available at their own database web sites (for BCI: <http://ctfs.arnarb.harvard.edu/bci/RAnalyticalTables>). Be sure to adjust `plotdim` to the size of your plot, and you will have to experiment with the size of the graph by adjusting `plotside`, `h`, and `w`.

```
> CTFSplot(plot = "bci", census = 6, type = "split")
```

```
[1] 4
```

■ `head(bci.spptable)[,1:5]` This creates a single pdf with 326 pages, a map for each species:

```
> pdf.allplot(splitdata = bci.split6, spplist = bci.spptable, elev = bci.elev, plotdim =
+ 100, 3000), plotside = 9, path = "/home/condit/data/maps/bci/")
```

This creates 326 pdfs, one separate file for each species:

```
> pdf.allplot(splitdata = bci.split6, spplist = bci.spptable, elev = bci.elev, plotdim =
+ 100, 3000), plotside = 9, singlefile = FALSE, path = "/home/condit/data/maps/bci/")
```

There is an equivalent function, `png.allplot`, that creates 326 separate maps in png format.

## 6. FULL STEM MAP

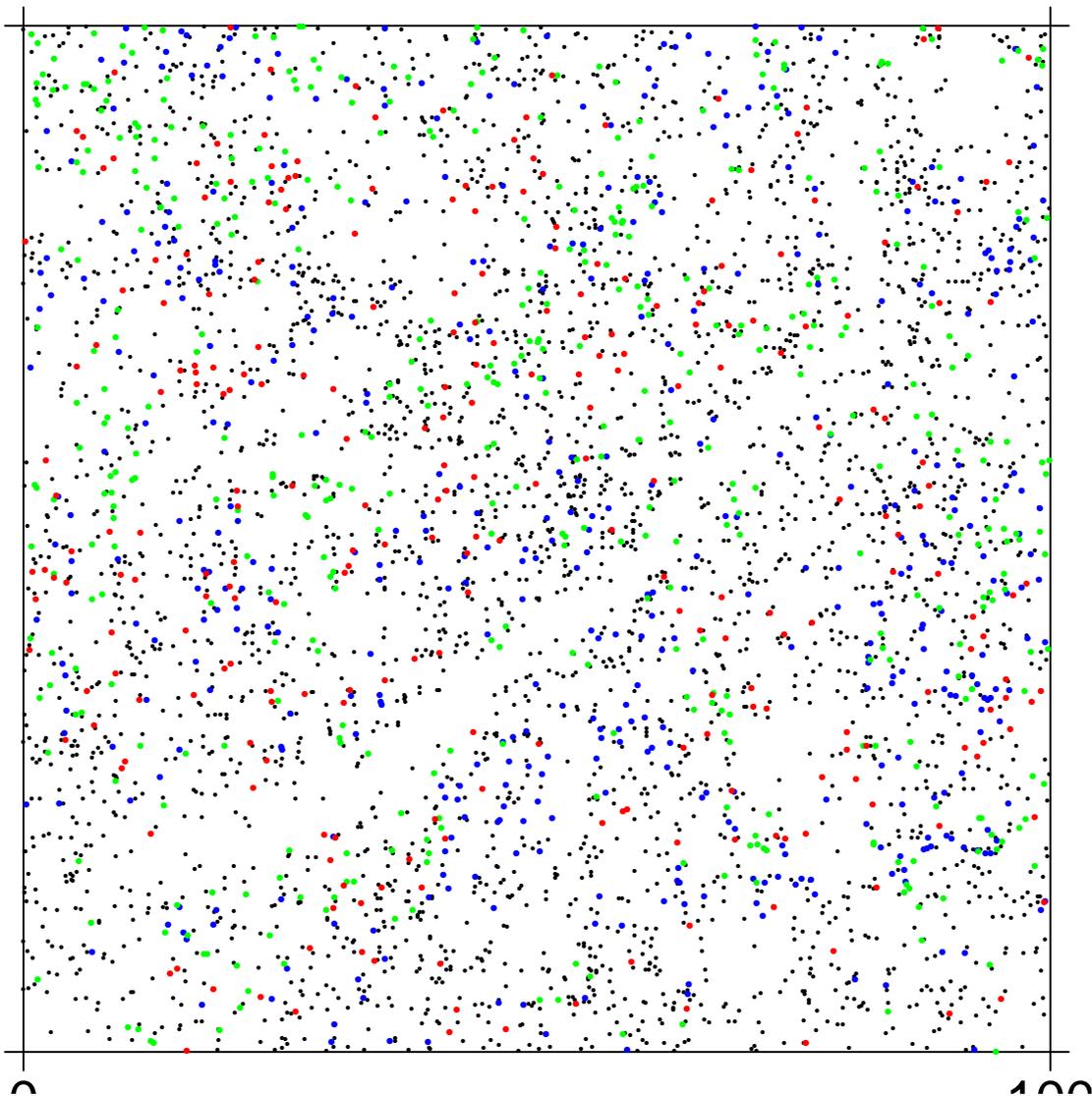
One function, `complete.plotmap`, makes a map of every individual of all species in black dots. Then it overlays a few individual species in different colors. The map can be any rectangular portion, or the entire plot, based on the options `xrange` and `yrange`.

The basic use draws a map of all individuals, then overlays the 3 most common species in green, blue, and red, in a 100x100 m section of the plot.

```
> CTFSplot(plot = "bci", census = 6, type = "full")
```

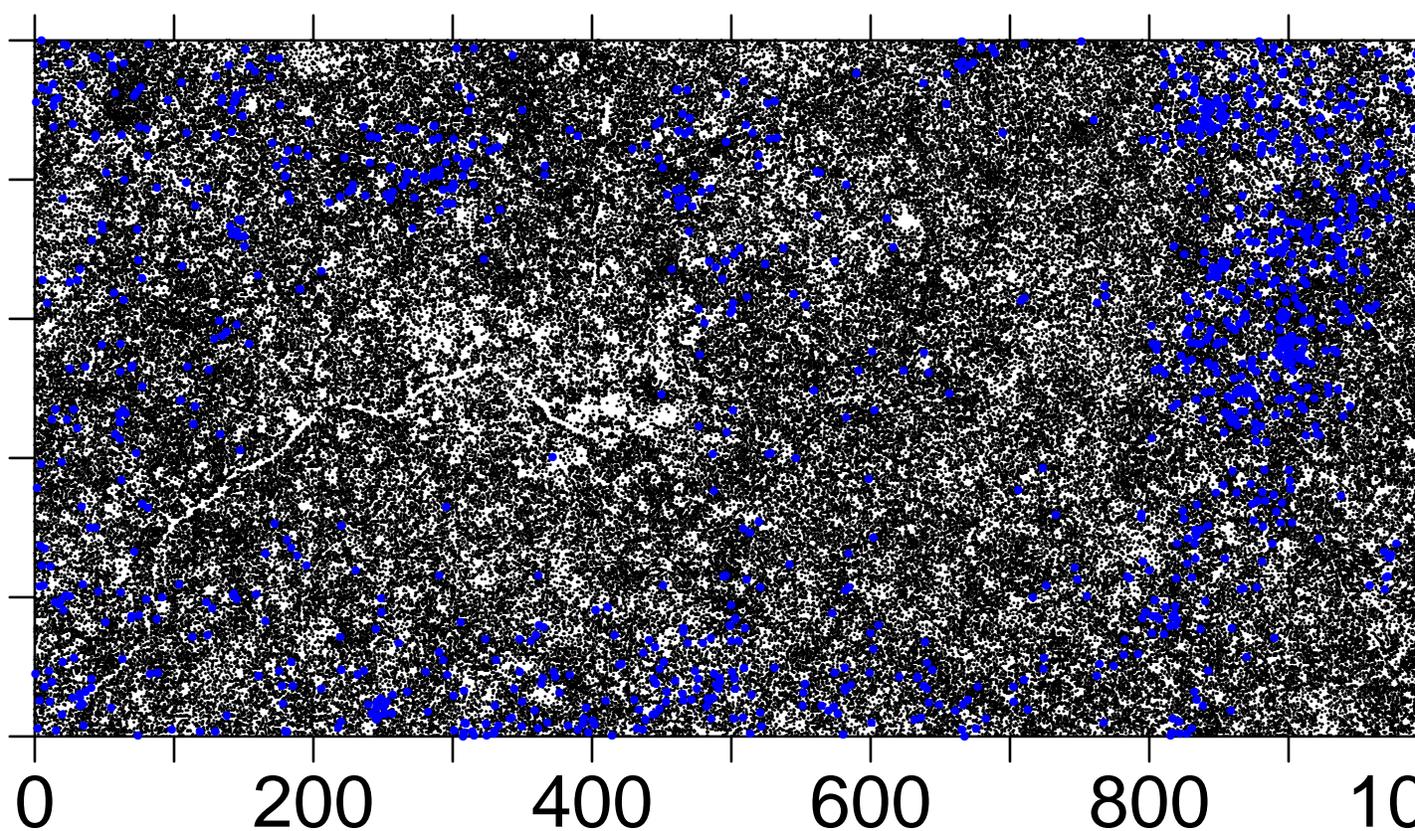
```
[1] 2
```

```
> complete.plotmap(cns = bci.full16)
```



To make a full plot map, it is useful to make `ptsize` very small. The stream in the southwest corner of the BCI plot is evident in the stem map.

```
> complete.plotmap(cns = bci.full16, spnames = "poular", xrange = c(0, 1000), yrange = c(0, 1000))
```



The following code exports the same map as a pdf to the file named:

```
> complete.plotmap(cns = bci.full16, spnames = "poular", xrange = c(0, 1000), yrange = c(0, 1000),  
+   export = "pdf", wd = 11, ht = 8.5, side = 8, filepath = "/home/condit/data/maps/complete.poular.pdf")
```

Output created by

---

```
> Sweave("~/programs/CTFSRpackage/CTFSmap.Rnw")
> system("htlatex CTFSmap.tex")
> system("pdflatex CTFSmap.tex")
> system("mv CTFSmap* ~/programs/CTFSRpackage/map")
> system("mv ~/programs/CTFSRpackage/map/CTFSmap.html ~/programs/CTFSRpackage/map/index.html")
> system("cp ~/programs/CTFSRpackage/map/* /var/www/Public/CTFS_RPackage/files/tutorials")
> system("rsync -avh -e ssh ~/programs/CTFSRpackage/map rcondit@ctfs.arnarb.harvard.edu")
```