OpenDAP, NetCDF + R

Fedor Baart

July 15, 2010

NetCDF

Available modules for reading NetCDF in R.

NetCDF

Available modules for reading NetCDF in R.

- RNetCDF
- netcdf
- ncdf
- ncdf4

Which one?

What library should I use?

Name	Maintained?	NetCDF4?	OpenDAP?	Windows/Linux/OSX?
RNetCDF	✓	✗	✗	✓ / ✓ / ✗
netcdf	✗	✗	✗	✗ / ✗ / ✗
ncdf	✓	✗	✓	✓ / ✓ / ✓
ncdf4	✓	✓	✓	✗ / ✓ / ✓

ncdf4

Reading an opendap file

```
> library(ncdf4)
> url <- "http://dtvirt5.deltares.nl:8080/thredds/dodsC/opendap/rijkswaterstaat/jarkus/profiles/transect.nc
> con <- nc_open(url)
> print.ncdf4(con)
[1] "File http://dtvirt5.deltares.nl:8080/thredds/dodsC/opendap/rijkswaterstaat/jarkus/profiles/transect.n
[1] ""
[1] "     21 variables:"
[1] "        int id[alongshore]   "
[1] "            long_name: identifier"
[1] "            comment: sum of area code (x1000000) and alongshore coordinate"
[1] "        int areacode[alongshore]   "
[1] "            long_name: area code"
[1] "            comment: codes for the 15 coastal areas as defined by rijkswaterstaat"
[1] "        char areaname[stringsize,alongshore]   "
[1] "            long_name: area name"
[1] "            comment: names for the 15 coastal areas as defined by rijkswaterstaat"
[1] "        int crs[]   "
[1] "            grid_mapping_name: Oblique Stereographic"
[1] "            semi_major_axis: 6377397.155"
[1] "            semi_minor_axis: 6356078.96281819"
[1] "            inverse_flattening: 299.1528128"
[1] "            latitude_of_projection_origin: 52.0922178"
[1] "            longitude_of_projection_origin: 5.23155"
[1] "            false_easting: 155000"
[1] "            false_northing: 463000"
[1] "            scale_factor_at_projection_origin: 0.9999079"
[1] "        double angle[alongshore]   "
[1] "            long_name: angle of transect"
[1] "            units: degrees"
[1] "            comment: positive clockwise 0 north"
[1] "        double mean_high_water[alongshore]   "
[1] "            long_name: mean high water"
```

Reading

Read some variables. They are stored as array.

```
> id <- ncvar_get(con, "id")
> cross_shore <- ncvar_get(con, "cross_shore")
> time <- ncvar_get(con, "time")
> z <- ncvar_get(con, "altitude", c(1, which(id == 7003600), 1),
+     c(-1, 1, -1))
> z[z == ncatt_get(con, "altitude")$`_FillValue`] <- NA
```

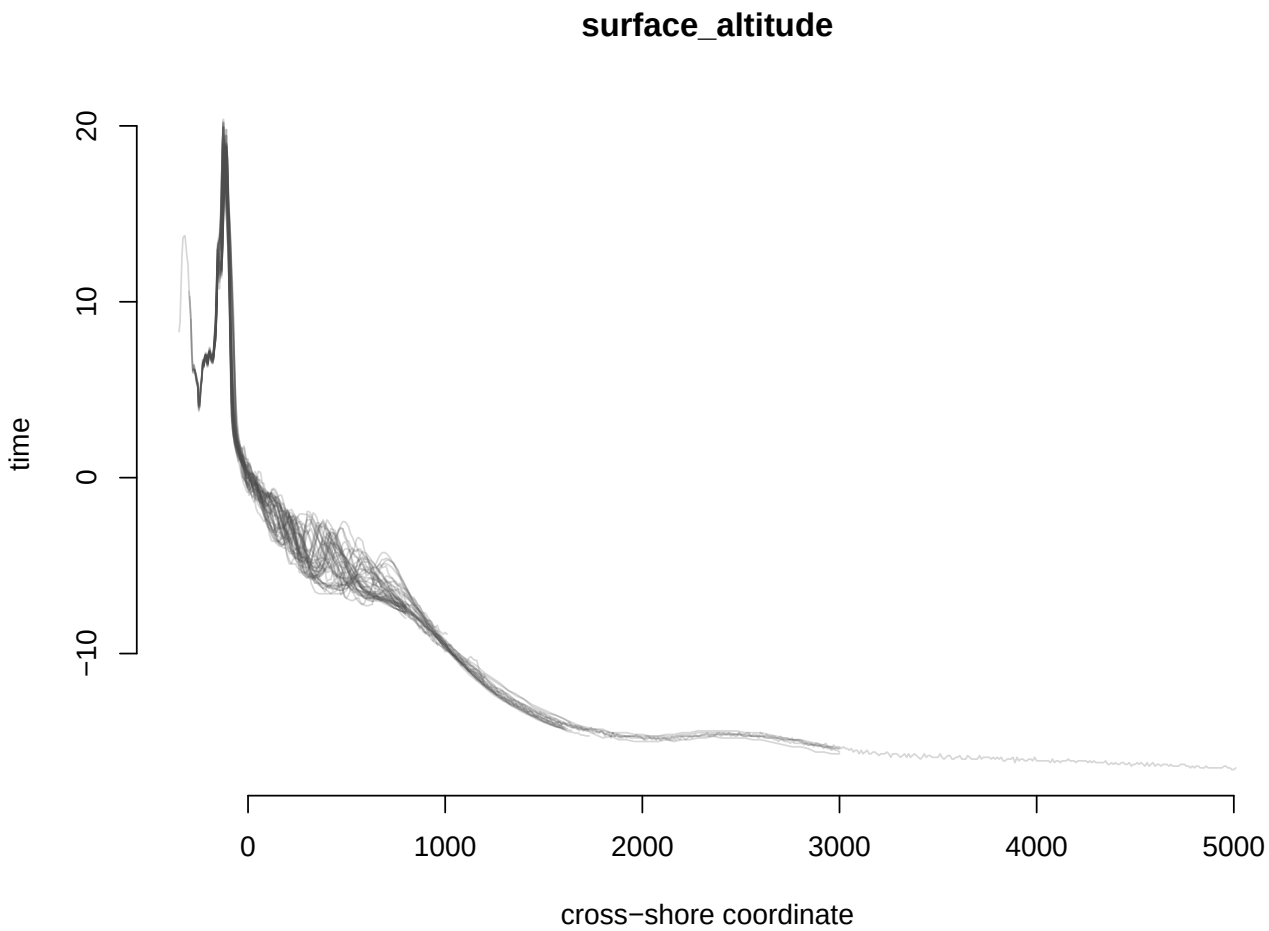
Reading

Transform the variables into something useful.

```
> dim(cross_shore)

[1] 1925

> dim(time)

[1] 45

> dim(z)

[1] 1925   45

> nonmissings <- !apply(is.na(z), 1, all)
> plot(c(), c(), type = "l", bty = "n", xlim = range(cross_shore[nonmissings]),
+     ylim = range(z, na.rm = TRUE), xlab = ncatt_get(con, "cross_shore")$long_name,
+     ylab = ncatt_get(con, "time")$standard_name, main = ncatt_get(con,
+         "altitude")$standard_name)
> for (i in 1:length(time)) {
+     col <- rgb(0.3, 0.3, 0.3, 0.2 + 0.8 * (1/length(time)))
+     points(cross_shore[!is.na(z[, i])], z[, i][!is.na(z[, i])],
+         col = col, type = "l")
+ }
```

Reading

This is the plot we created



Creating a NetCDF file

Create an NetCDF file, using an example dataset from ToxLim.

```
> library(ToxLim)
> names(LIMbarents)

 [1] "file"         "NUnknowns"    "NEquations"   "NConstraints" "NComponents"
 [6] "NExternal"    "NVariables"   "A"            "B"            "G"
[11] "H"            "Cost"         "Profit"       "Flowmatrix"   "VarA"
[16] "VarB"         "Parameters"   "Components"   "Externals"    "rates"
[21] "markers"      "Variables"    "costnames"    "profitnames"  "eqnames"
[26] "ineqnames"    "Unknowns"     "ispos"
```

Creating a NetCDF file

Let's store 1 variable as an example

```
> species.names <- dimnames(LIMbarents$Flowmatrix)[[1]]
> dimSpecies <- ncdim_def("species", units = "", vals = 1:length(species.names))
> varFlowmatrix <- ncvar_def(name = "flow", units = "count", dim = list(dimSpecies,
+     dimSpecies), missval = NA, longname = "Count of species found inside another species")
> dimString <- ncdim_def("stringlength", units = "", vals = 1:max(sapply(species.names,
+     nchar)), create_dimvar = FALSE)
> varName <- ncvar_def("speciesname", "", list(dimString, dimSpecies),
+     NA, prec = "char")
> con <- nc_create("limbarents.nc", list(varFlowmatrix, varName))
> ncvar_put(con, varFlowmatrix, Flowmatrix(LIMbarents))
> ncvar_put(con, varName, species.names)
> nc_close(con)
```

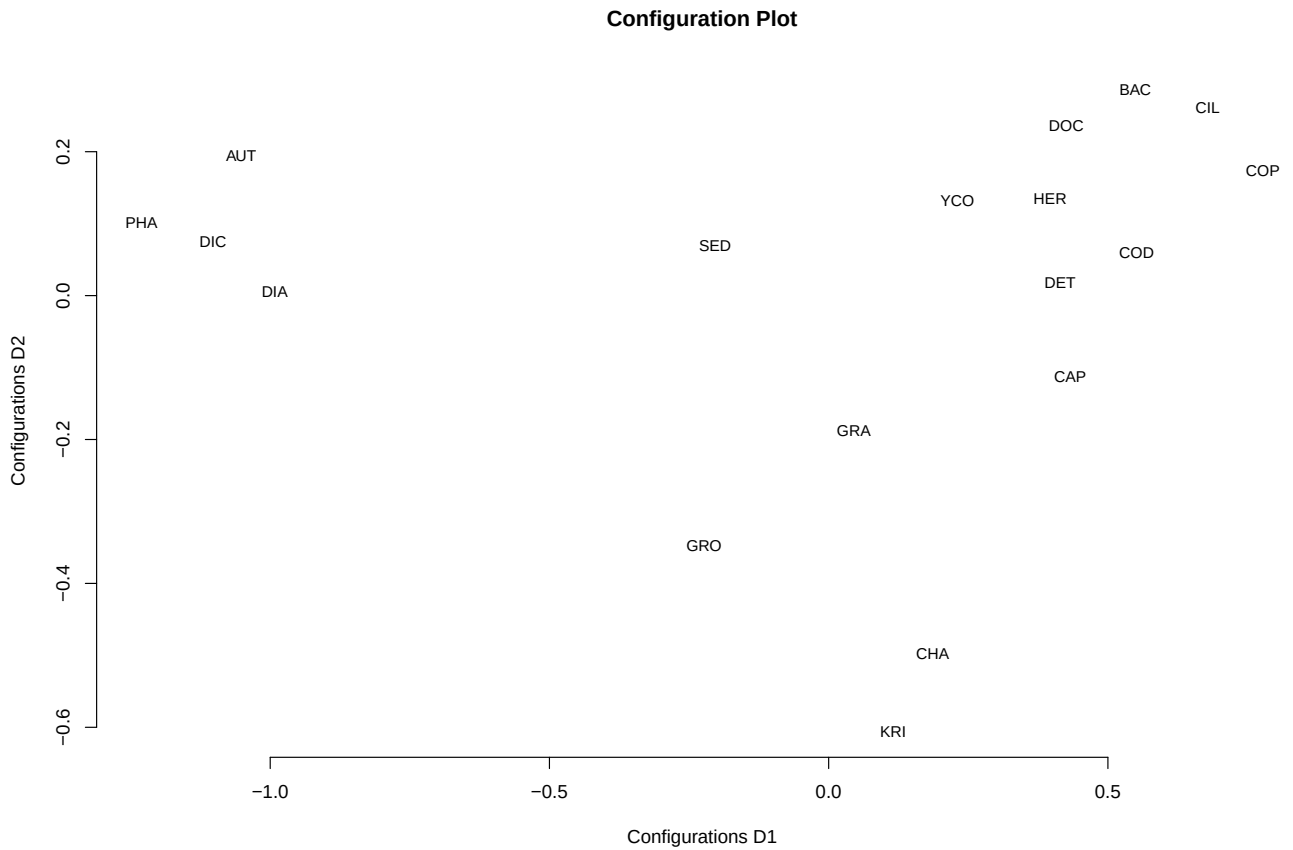
Creating a NetCDF file

Check to see if everything's there

```
> library(smacof)
> flow <- flowmatrix[c(-7, -19), c(-7, -19)]
> plot(smacofSym(max(flow) - flow, metric = FALSE), bty = "n")

> con <- nc_open("limbarents.nc")
> flowmatrix <- ncvar_get(con, "flow")
> species.name <- ncvar_get(con, "speciesname")
> dimnames(flowmatrix) <- list(species.names, species.name)
> library(smacof)
> flow <- flowmatrix[c(-7, -19), c(-7, -19)]
> plot(smacofSym(max(flow) - flow, metric = FALSE), bty = "n")
```

Creating a NetCDF file

Nodes plotted using the smacof algorithm (based on similarity).



1. `http://cran.r-project.org` information about ncdf, ncdf4.
2. `http://www.unidata.ucar.edu/software/netcdf/` information about netcdf.
3. `http://cf-pcmdi.llnl.gov/` information about CF convention.
4. `http://public.deltares.nl/display/OET/Data+tutorials` tutorials.