StableClim - Process piControl simulations

Stuart C Brown

Contents

Processing CMIP5 Pre-industrial control simulations

The code below processes the CMIP5 pre-industrial control (piControl) simulations.

N.B. Only the temperature data is processed during this analysis.

The piControl simulations are multi-century unforced climate simulations, where the initial model conditions are set based on atmospheric gas concentrations prior to large-scale industrialisation. The simulations have non-evolving boundary conditions (e.g. non-evolving land use and greenhouse gas concentrations) relevant to the chosen start year (Taylor *et al*). Unlike, for example TraCE-21ka, these simulations are **not** reconstructions of temporally explicit pre-industrial climate, but are instead used to simulate internal model variability, which can be used as a proxy for natural (unforced) climate variability.

The code processes the data using CDO tools and Windows Subsystem for Linux.

Some minor post-processing is performed using NCO.

The code performs the following functions for each model:

1) Regrid to a common 2.5°x2.5° grid using bilinear interpolation
2) Convert the data to annual averages
3) Convert the units
4) Append some meta-data to the output NetCDF files

Set-up

Set up the various directories etc. that will be used in the processing. Need a list of model names, rcp scenarios, and variables to iterate through

```r
library(raster)
library(ncdf4)
model_list <- list.dirs("../data/CMIP5/piControl/ts/",
  full.names = FALSE,
  recursive = FALSE
)
# Remove the CNRM-CM5 model
## Issue with volcanic forcing (see https://doi.org/10.1073/pnas.1210514109)
model_list <- model_list[!grepl("CNRM-CM5", model_list, perl = TRUE)]
clim_vars <- "ts"
```

Here we have access to 19 models. Each of the model outputs is named and identified with a triad of integers (**N**, **M**, **L**) for example "r**1**i**1**p**1**". This identifier is used to distinguish among closely related simulations from a given model. The models have a different number of ensemble members (hereafter, realisations) available. We elected to use only the first realisation (r1i1p1) from each model for the pre-industrial control runs as all models, with the exception of the Community Climate System Model ver. 4 (CCSM4; Gent *et al*), only had a single realisation for pre-industrial control conditions. Furthermore, the additional pre-industrial realisations (r2i1p1 and r3i1p1) for the CCSM4 model only ran for 156 and 120 years, respectively.

Creating the common grid

This sections creates the common 2.5°x2.5° grid for the bilinear interpolation

```
r <- raster(res = 2.5, crs = crs(raster()))
r[] <- 1
xvals <- unique(values(init(r, "x")))
yvals <- unique(values(init(r, "y")))
nx <- length(xvals)
ny <- length(yvals)
lon <- ncdim_def("longitude", "degrees_east", xvals)
lat <- ncdim_def("latitude", "degrees_north", yvals)
mv <- 0
var_temp <- ncvar_def(
  name = "grid",
  units = "",
  dim = list(lon, lat),
  longname = "grid",
  missval = mv,
  prec = "byte"
)
ncout <- nc_create(
  filename = "C:/tmp/cdo_processing/dest_grid.nc",
  list(var_temp), force_v4 = TRUE
)
## put the data in the file
ncvar_put(
  nc = ncout,
  varid = var_temp,
  vals = values(r),
  start = c(1, 1),
  count = c(-1, -1)
)
nc_close(ncout)
```

Processing the piControl simulations

This next section iterates through each of the files and follows the processing steps outlined above

```
baseComm <- "wsl cd /mnt/c/tmp/cdo_processing/;"
for (model in model_list[c(10, 11)]) {
  ## find and copy the piControl files to the processing directory
  realFiles <- list.files(
    paste("../data/CMIP5/piControl/ts", model, "r1i1p1", sep = "/"),
    "\\.nc$",
    full.names = TRUE
  )
  file.copy(realFiles, to = "C:/tmp/cdo_processing/")
  ## Now use CDO to process the files
  inFiles <- list.files("C:/tmp/cdo_processing/", full.names = FALSE)
  inFiles <- inFiles[!grepl("dest_grid", inFiles)]
  inFiles <- sub("_[^_]+$", "*", inFiles[1])
  outFile <- paste0(
    paste("piControl_ts", model, "r1i1p1_merged", sep = "_"),
    ".nc"
  )
  comm <- paste(baseComm, "cdo cat", inFiles, outFile, sep = " ")
  comm <- gsub(pattern = "\\(", replacement = "\\\\(", x = comm)
  comm <- gsub(pattern = "\\)", "\\\\)", comm)
  shell(comm, mustWork = TRUE)
  inFile <- outFile
  outFile <- paste0(paste("ensAvg_regridAnnAvg_piControl_ts", model,
    "r1i1p1",
    sep = "_"
  ), ".nc")
  ## The command for all of the processing steps
  comm <- paste(baseComm, "cdo -f nc4 --cmor -k grid -b F32
               remapbil,dest_grid.nc -settaxis,0001-01-16,00:00,1year
               -setunit,'degC' -subc,273.15 -setname,ts -yearmean
               -setmissval,-999", inFile, outFile, sep = " ")
  comm <- gsub(pattern = "\\(", replacement = "\\\\(", x = comm)
  comm <- gsub(pattern = "\\)", "\\\\)", comm)
  shell(comm, mustWork = TRUE)
  ## Now clean up the files
  ## Copy the two output files to a new directory
  moveFiles <- paste0("C:/tmp/cdo_processing/", outFile)
  file.copy(moveFiles, "C:/tmp/cdo_outputs/")
  ## delete all files from processing directory
  rmFiles <- list.files("C:/tmp/cdo_processing/", "\\.nc$", full.names = TRUE)
  rmFiles <- rmFiles[!grepl("dest_grid.nc", rmFiles)]
  file.remove(rmFiles)
}
```

All the CMIP5 piControl simulations have now been converted to annual averages and are on a common 2.5° x 2.5° grid.

Convert the outputs to short and add scale_factor to the files

To save disk-space the output files are converted to short integers using NCO tools, with the `scale_factor` attribute added to the header for each of the files.

Files are then compressed using `ncks`.

This process reduces the output file size from ~480MB to ~240MB, and retains 2 decimal places of precision.

```
cd /mnt/c/tmp/cdo_outputs
ls | egrep '_piControl_ts_' | while read file; do
    ncap2 -O -s 'ts=short(ts/0.01);ts@scale_factor=0.01' $file $file
    ncks -L 5 -O $file $file
done
```