Climate Data Processing

Acquisition -> Regridding -> Merging -> Converting to Raster

CMIP6 Data Acquisition

Seven CMIP6 models (CESM2-WACCM, CMCC-ESM2, GFDL-ESM4(gr), MPI-ESM1-2-HR, NorESM2-LM(gr)) were selected and their monthly thetao (ocean temperature), so (ocean salinity), no3 (nitrate concentration) and ph (ocean pH) products were downloaded. These model sources were selected for their availability all variable products. Experiment IDs used were historical (1850-2014), ssp26 (SSP1-2.6), ssp45 (SSP2-4.5) and ssp85 (SSP5-8.5). Variant label was r1i1p1f1. If gn and gr products were available, gr was used. These products are the raw .nc (NetCDF) files that are processed.

1. Regridding

Regridding scripts overlay a custom 1 degree grid onto the .nc data and regrid using bilinear interpolation. This uses the regrid.nc function to create cdo commands to regrid raw files. This uses cdo command "remapbil" The following line in the scripts produces "cdo -P 10 -remapbil, gridfile input .nc output .nc" which regrids the input files.

```
    system(paste(paste("cdo -P 10 -remapbil,", file.grd, sep = ""), files.nc[j],
paste0(opath[i], '/', basename(files.nc[j])), sep = (" "))) #creates the cdo command
(-P 10 = use 10 cores) (remapbil = bilinear interpolation)
```

2. Merging

.nc files from CMIP6 models come in different numbers of time bins depending on the model source. To make these consistent, the merging scripts merge all time bins for a model/variable/time-period into one .nc file to work with in downstream analyses. This uses the cdo command "mergetime". The following line in the scripts produces the "cdo -P 10 mergetime (multiple input .nc files) output .nc"

```
system(paste(paste("cdo -P 10 mergetime", paste(parameter.list[[j]], collapse=' ')),
paste0(opath[i], '/', 'Merged_',parameter.name,'_',model.name,'_',time.period,'.nc')))
```

3. Converting .nc to .tiff raster format

To make the climate data files more accessible in R for downstream analyses they should be converted to .tiff raster files. To extract the data from these files the raster::brick() function is used. The climate .nc files have different numbers of depth levels as they originate from different sources, however brick() can only extract over a certain number of levels. To navigate this issue we first extract information on the files depth levels and run the brick() function for each depth level up to 150m. While there is no agreed upon depth range for *Seriola lalandi*, 98% of occurrences from OBIS and GBIF (with depth information) were recorded in the first 150m of water. This approach generates separate .tiff files for each native depth level up to 150m. This script is run individually on each merged .nc file, outputting processed .tiff raster files.