delta downscaling

THIS IS WORK IN PROGRESS, DO NOT USE FOR ANY ANALYSIS

Downscaling a dataset in pastclim `pastclim`

The `terra` package can downscale reconstructions using a bilinear method. Other `R` packages also offer various approaches to downscale rasters.

For palaeoclimate reconstructions, the delta method has been shown to be very effective (Beyer et al, REF), but sea level changes require some care on how to apply such a method. `pastclim` includes functions to use the delta method for downscaling. We will focus on South East Asia, as this area was greatly affected by sea level changes over the last glaciation. We will work on the variable "bio01". However, note that, in most cases, it is better practice to downscale monthly temperature and precipitation estimates, and then compute the "bio" variables from those quantities.

First, we extract a time series for the variable and region of interest:

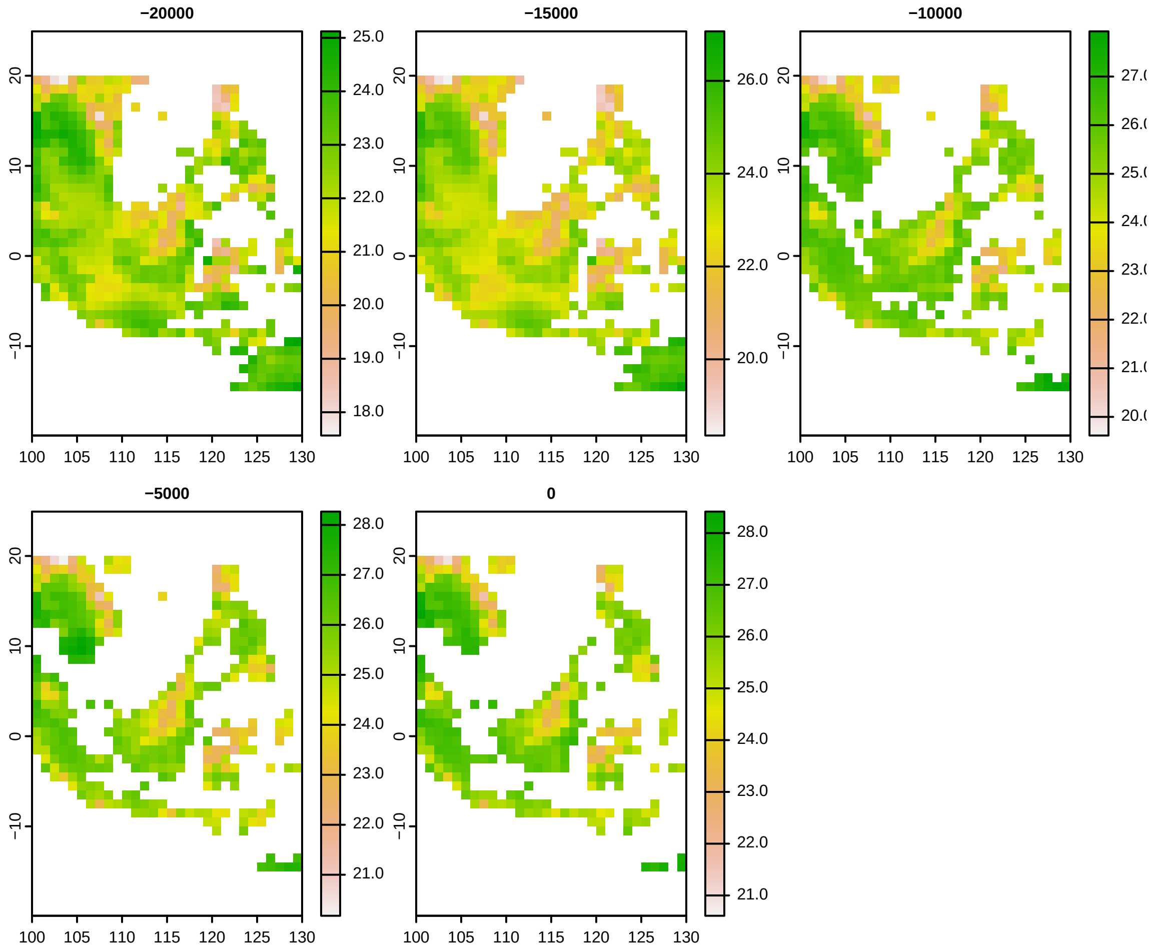
```
library(terra)
#> terra 1.6.47
library(pastclim)
sea_ext<- terra::ext(100, 130, -15, 20)
model_series <- region_series(bio_variables = "bio01",dataset="Example",
                              ext = sea_ext)
model_series
#> class       : SpatRasterDataset
#> subdatasets : 1
#> dimensions  : 35, 30 (nrow, ncol)
#> nlyr        : 5
#> resolution  : 1, 1  (x, y)
#> extent      : 100, 130, -15, 20  (xmin, xmax, ymin, ymax)
#> coord. ref. : lon/lat WGS 84
#> source(s)   : memory
#> names       : bio01
```

Note that a `SpatRasterDataset` has been returned, which contains only one `SpatRaster` for "bio01". We want to work on that `SpatRaster`, so we extract it with:

```
model_rast <- model_series$bio01
model_rast
#> class       : SpatRaster
#> dimensions  : 35, 30, 5  (nrow, ncol, nlyr)
#> resolution  : 1, 1  (x, y)
#> extent      : 100, 130, -15, 20  (xmin, xmax, ymin, ymax)
#> coord. ref. : lon/lat WGS 84
#> source(s)   : memory
#> varname      : bio01
#> names        : bio01_-20000, bio01_-15000, bio01_-10000, bio01_-5000,  bio01_0
#> min values   :     17.56506,     18.35822,     19.61696,    20.17858, 20.60729
#> max values   :     25.12129,     27.06657,     27.92988,    28.26682, 28.41459
#> time (years): -18050 to 1950
```

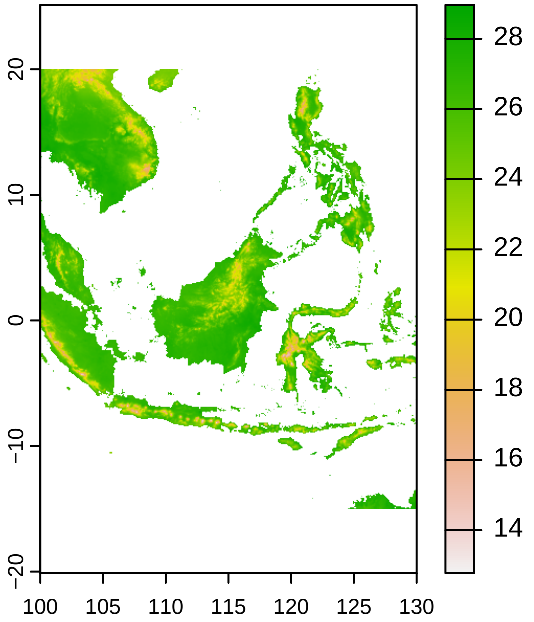
And we can now plot it:

```
plot(model_rast, main=time_bp(model_rast))
```



We can see how that the reconstructions are rather coarse (the Example dataset uses 1x1 degree cells). We now need a set of high resolutions observations for the variable of interest that we will use to generate the delta raster used to downscale reconstructions. We will use data from Worldclim2 at 5 minute resolution (but other datasets such as CHELSA would be equally suitable).

```
high_res_file <- system.file("extdata/wc2.1_5m_bio_1_sea.tif",package="pastclim")
high_res_obs<-terra::rast(high_res_file)
plot(high_res_obs)
```



We need to make sure that the extent of the modern observations is the same as the extent of the model reconstructions:

```
ext(high_res_obs)==ext(model_rast)
#> [1] TRUE
```

If that was not the case, we would use `terra::crop` to match the extents.

We also need a high resolution global relief map (i.e. integrating both topographic and bathymetric values) to reconstruct past coastlines following sea level change. The relief raster will need to have the same extent and resolution as the high resolution observations. We can download one (based on ETOPO2022) with:

```
topo_rast <- pastclim::download_relief(high_res_obs)
#> Registered S3 methods overwritten by 'adehabitatMA':
#>   method                       from
#>   print.SpatialPixelsDataFrame sp
#>   print.SpatialPixels          sp
#> Querying NOAA database ...
#> This may take seconds to minutes, depending on grid size
#> Building bathy matrix ...
```

We can quickly confirm that the resulting relief raster has the same extent as the model reconstructions, as well as the same resolution of the high resolution climate observations:

```
ext(topo_rast) == ext(model_rast)
#> [1] TRUE

ext(topo_rast) == ext(high_res_obs)
#> [1] TRUE
```

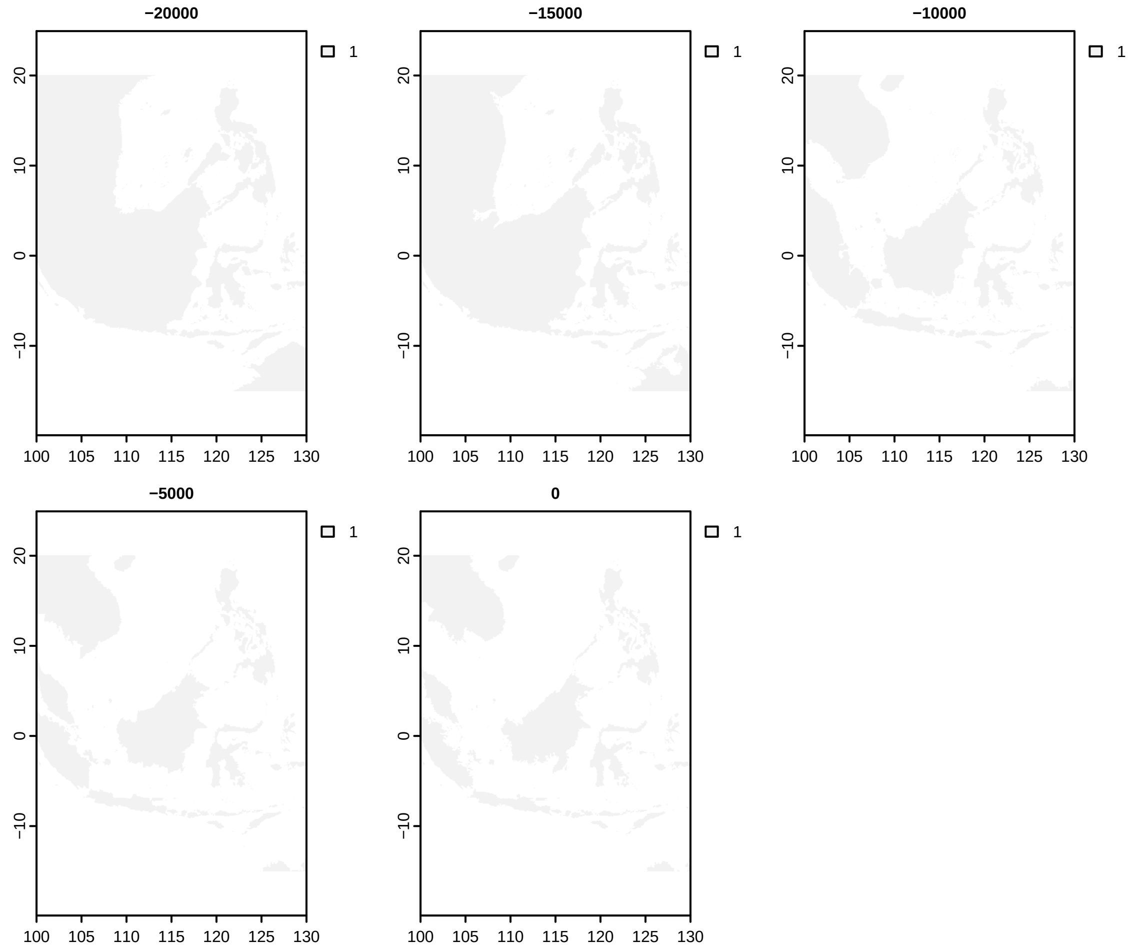
```
ncol(topo_rast) == ncol(high_res_obs)
#> [1] TRUE
nrow(topo_rast) == nrow(high_res_obs)
#> [1] TRUE
```

#TEMPORARY UNTIL THIS FUNCTIONS ARE INTEGRATED INTO PASTCLIM

Note that we will use the draft functions for downscaling, which can be accessed by usign the prefix `pastclim:::`. The `:::` access internal functions. You will also get a message reminding you that these functions are not final and thus should not be used for real analysis.

We can now generate a high resolution land mask for the periods of interest. By default, we use the sea level reconstructions from Spratt et al 2016, but a different reference can be used by setting sea levels for each time step (see the man page for `make_land_mask` for details):

```
high_res_mask <- pastclim:::make_land_mask(topo_rast = topo_rast,
                                           time_bp = time_bp(model_rast))
#> This function is still under development; do not use it for real analysis
plot(high_res_mask, main=time_bp(high_res_mask))
#> Warning in time_bp(high_res_mask): the time units of SpatRaster are not 'years'
#> it might be a problem with the time units not being properly set in the original
#> nc file
```

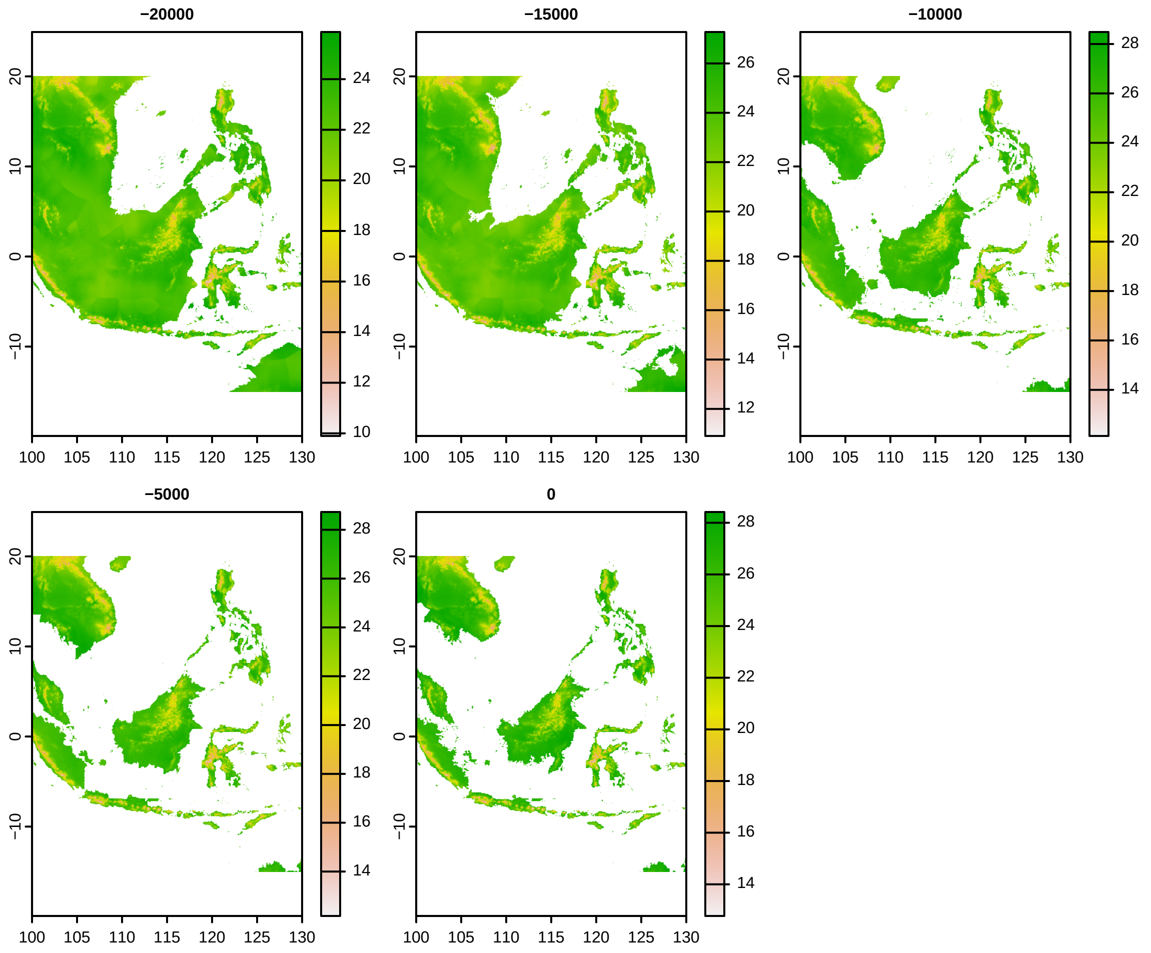


We can now compute a delta raster and use it to downscale the model reconstructions:

```
delta_rast<-pastclim:::delta_compute(x=model_rast, ref_time = 0, obs = high_res_obs)
#> This function is still under development; do not use it for real analysis
#> [inverse distance weighted interpolation]
model_downscaled <- pastclim:::delta_downscale (x = model_rast, delta_rast = delta_rast,
                                 x_landmask_high = high_res_mask)
#> This function is still under development; do not use it for real analysis
#> [inverse distance weighted interpolation]
#> [inverse distance weighted interpolation]
#> [inverse distance weighted interpolation]
#> [inverse distance weighted interpolation]
#> [inverse distance weighted interpolation]
model_downscaled
#> class       : SpatRaster
#> dimensions  : 420, 360, 5  (nrow, ncol, nlyr)
#> resolution  : 0.08333333, 0.08333333  (x, y)
#> extent      : 100, 130, -15, 20  (xmin, xmax, ymin, ymax)
#> coord. ref. : lon/lat WGS 84
#> source(s)   : memory
#> names       : bio01_-20000, bio01_-15000, bio01_-10000, bio01_-5000,  bio01_0
#> min values  :     9.895848,     10.90229,     12.13324,    12.17908, 12.77442
#> max values  :    25.863934,     27.27367,     28.48588,    28.72786, 28.41692
#> time (years): -18050 to 1950
```

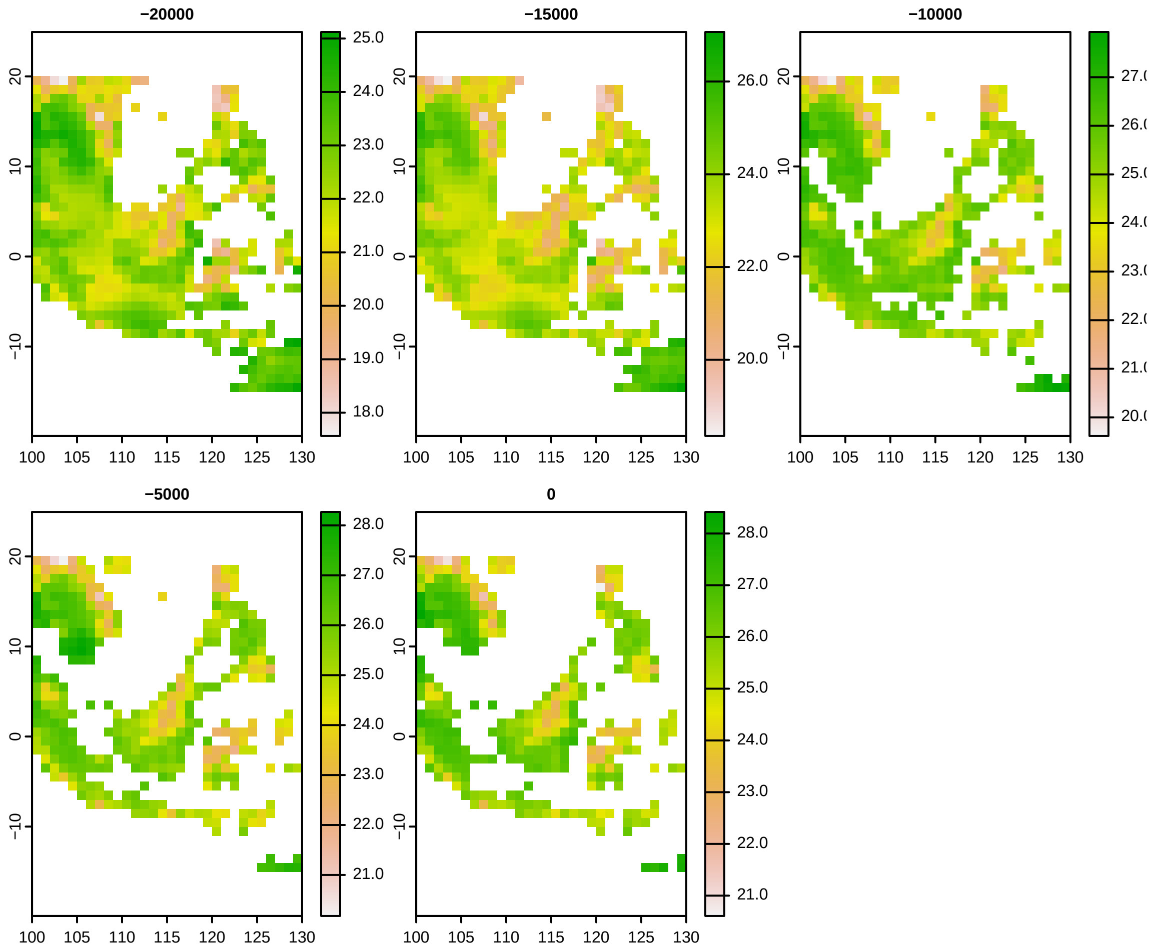
Let’s inspect the resulting data:

```
plot(model_downscaled, main=time_bp(model_downscaled))
```



And, as a reminder, the original reconstructions (note that the colour scales are not the same! `terra` chooses a scale for each time step based on the time specific range):

```
plot(model_rast, main=time_bp(model_rast))
```


−20000
20
10
0
−10
100
105
110
115
120
125
130
25.0
24.0
23.0
22.0
21.0
20.0
19.0
18.0
−15000
20
10
0
−10
100
105
110
115
120
125
130
26.0
24.0
22.0
20.0
−10000
20
10
0
−10
100
105
110
115
120
125
130
27.(
26.(
25.(
24.(
23.(
22.(
21.(
20.(
−5000
20
10
0
−10
100
105
110
115
120
125
130
28.0
27.0
26.0
25.0
24.0
23.0
22.0
21.0
0
20
10
0
−10
100
105
110
115
120
125
130
28.0
27.0
26.0
25.0
24.0
23.0
22.0
21.0