

grabr - Grids Across Borders



Please add all your code to this GitHub repo: go to clone, then copy the link and make a new version controlled project in RStudio.

<https://github.com/gndaskalova/grabr>

Please add your email, so that we can keep in touch:

gndaskalova@gmail.com

jamie.alison@liverpool.ac.uk

stijn.bruneel@ugent.be

anneleen.rutten@uantwerpen.be

theoni.photopoulou@gmail.com

bob.ohara@ntnu.no

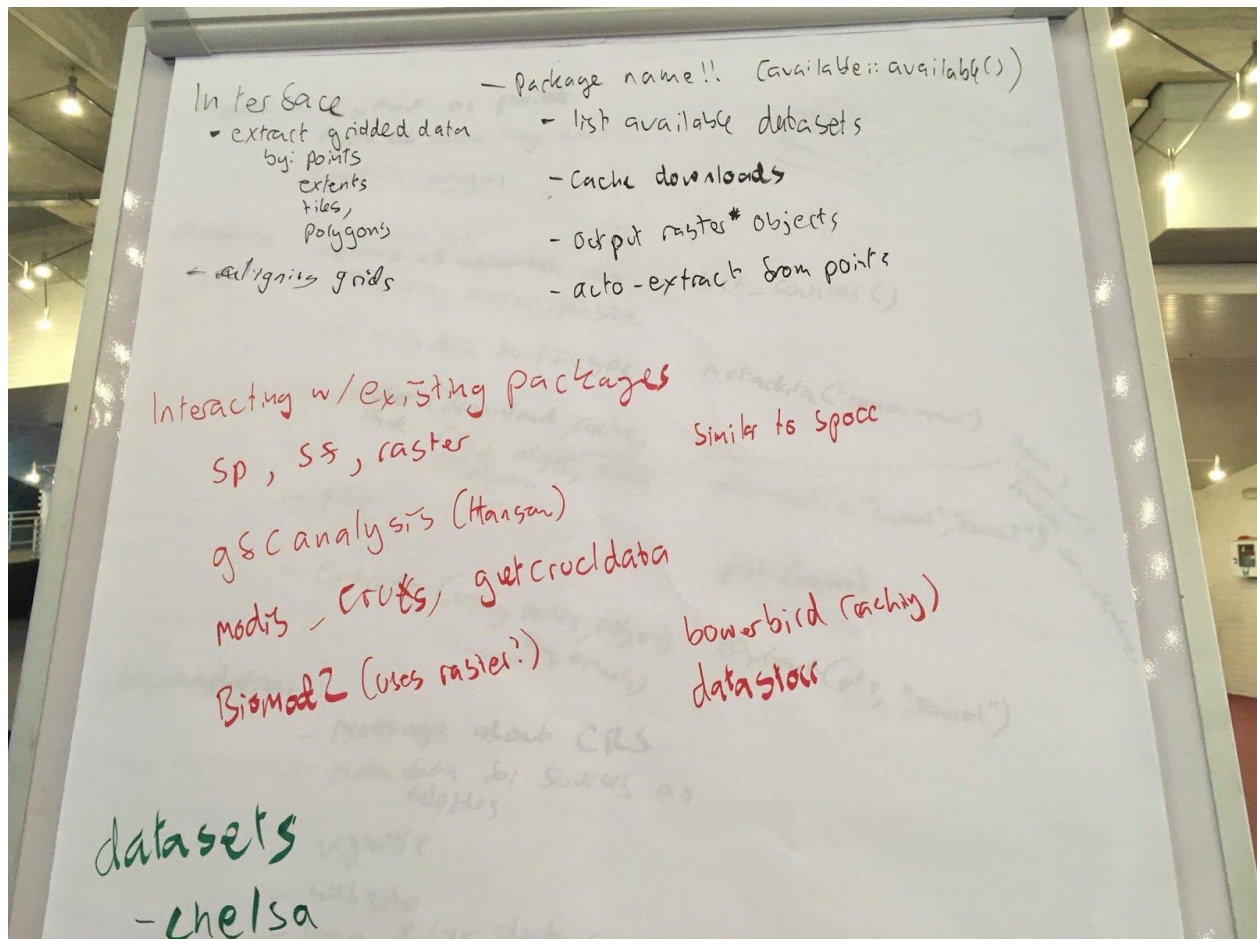
kramer@izw-berlin.de

nico.friess@posteo.de
nick.golding.research@gmail.com

Collaborators: StefanPinkert@gmx.de

<https://ourcodingclub.github.io/2017/02/27/git.html> - Coding Club tutorial on using git through RStudio

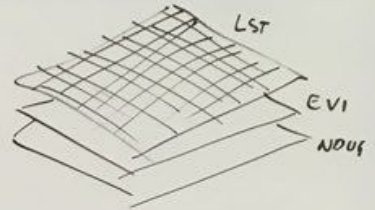
Please add photos here.



datasets

- chelsa
- CRU
- Worldclim 1, 2 (raster::getData())
- halpern ^{marine impacts} ~~forest cover~~
- huason ^{forest cover}
- corine (Europe)
- landcover map (UK)
- Modis (LST, SST, NDVI, EVI)

- Inputs = extent or points
- type of data they want
 - Coashive polygon



- Outputs
- Single summary list
 - names of available data
 - resolution, extent, file size
 - full meta data for file type

`list_sources()`

`metadata("source.name")`

- raster (download, cache, crop, align, mask, sort projection)
- plot

`download(c("source1", "source2"))`

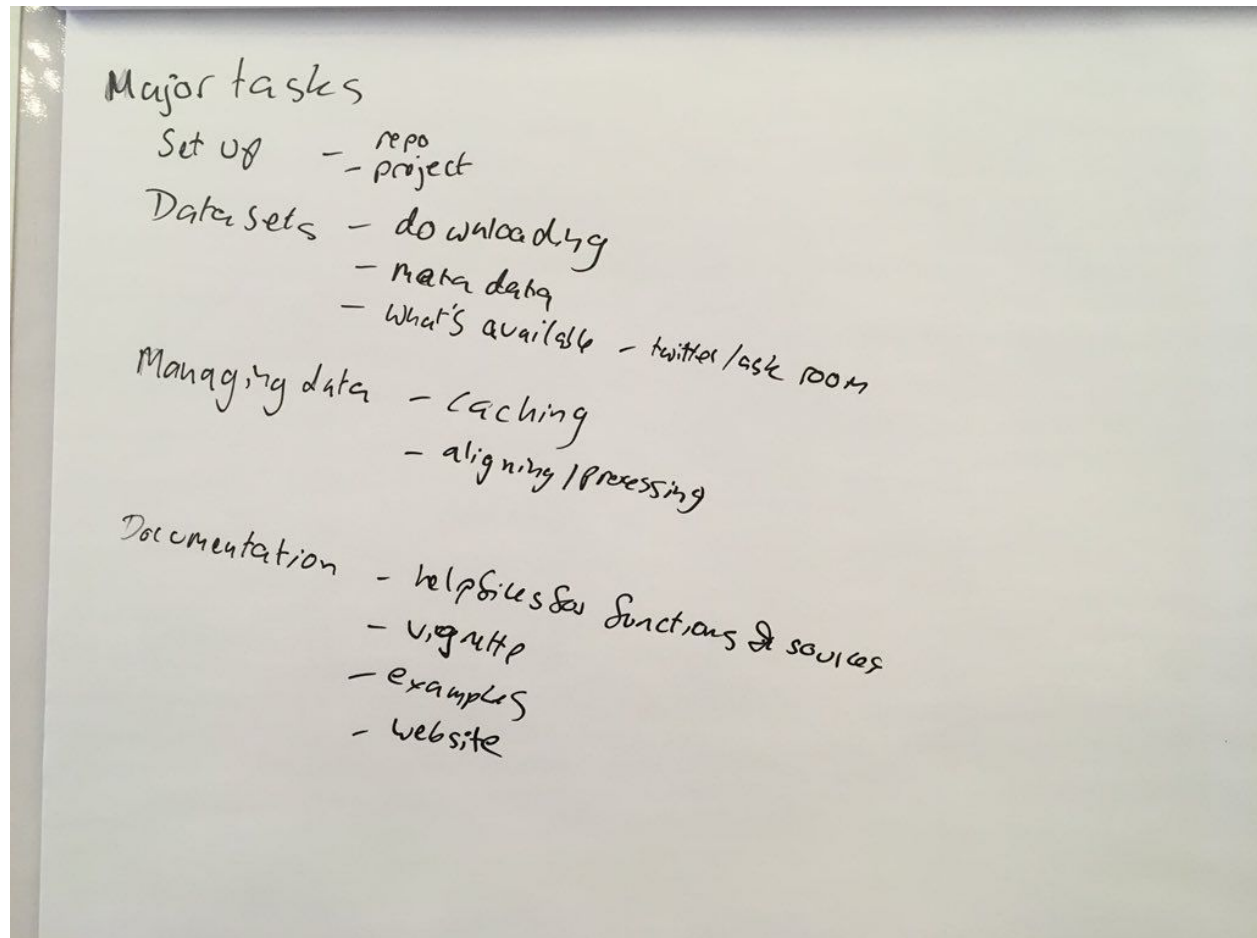
`plot(object)`

- Extract (using points, polygons, map extents)

`extract(pts, "source1")`

Documentation

- Message about CRS
- meta data for sources as help files
- vignette
- website
- logo & hex sticker
- example



Tasks

Getting Datasets

Nicolas - chelsa

Stefan - chelsa and maybe EarthEnv (Landcover, Habitat heterogeneity and Freshwater environmental variables)

Caching

Jamie

Nick G

Stephanie C

Anneleen

Harmonising

Bob
Stijn

Writing Functions & documentation

Gergana
Theoni

Possible Data sets

- Chelsa
- CRU
- WorldClim 1, 2
- Halpern
- Hanson
- CORINE (Europe)
- Landcover
- MODIS
- EarthEnv

Harmonising Data

<https://github.com/bowlerbear/geographyDrivers/blob/master/processing.R>
<https://github.com/bowlerbear/harmonizeRasters/blob/master/harmonizeRastersfunction.R>
<https://www.nceas.ucsb.edu/scicomp/usecases/resamplerasterimages>

Tasks for harmonisation

Check resolution -> project to common resolution (need to decide common resolution)

Check projection -> project to common projection

Clipping to required extent

Get required years

Aggregate (raster::disaggregate)

Writing Functions & documentation

Tasks:

- Make a list of functions
- A skeleton vignette

- An R script for each function
- Document what the functions do - for the help file
- Error messages

The GraBR package

This package aims to...(building on sdmpredictors) list, extract and process gridded environmental datasets for use with ecological datasets (?)

Core functions:

listDatasets() - lists available gridded datasets for the specified polygon/points

- Name of dataset, resolution, size

list_sources()

list_sources(extent)

metaData() - provides full metadata for a selected dataset

query() - query(extent, source = dataset, var = variable, ...) # returns data frame/list

requery() - query(query(), extent, source = dataset, var = variable, ...) # returns data frame/list

grab() - grab(query()) # returns data, from query

snatch() - snatch(extent, source = dataset, var = variable, ...) # returns data, from query() args

data_store() # get the path to the data store directory

data_store("some_url") # set the data store directory

plot(object)

extracts(points, raster layers)

harmonizeRasters()

Hidden Functions

Within grab(extent,sources,vars,...):

query_lookup(lookup,extent,sources,vars...)

Where lookup is the master data frame where each row is a unique combination of source, variable and any temporal subdivisions, for example:

source	var	month	url
chelsea	tmin	jan	https://www.wsl.ch/lud/chelsea/data/bioclim/integer/

chelsa	tmin	feb	https://www.wsl.ch/lud/chelsa/data/bioclim/integer/
chelsa	bio1	NA	https://www.wsl.ch/lud/chelsa/data/bioclim/integer/
worldclim	tmin	jan	(worldclim URL...)

The output of query_lookup is essentially a relevant subset of the master data frame, called “lu”.

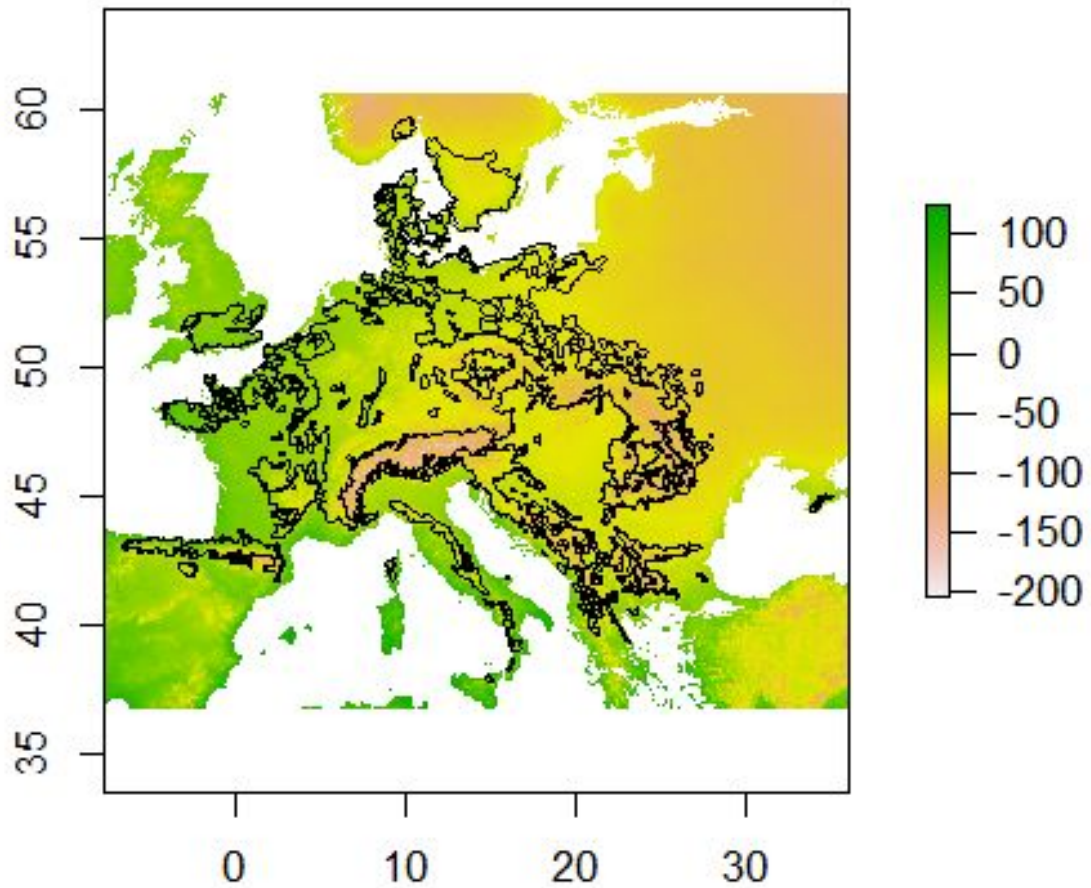
download_cached(lu)

This function takes unique(lu\$url) and checks if they are already downloaded. If they aren't, the relevant file is downloaded to the relevant directory.

load_cached(lu,extent)

This function takes all rows of lu, extracts the appropriate raster layers and crops them to the appropriate extent. Within load_cached a number of other functions will be implemented relevant to specific data sources, for example load_chelsa() which loads rasters from downloaded chelsa datasets. In the process of stacking the rasters, harmonizeRasters() will ensure they have compatible resolutions and projections etc. The stack will then be cropped to the appropriate extent. To make this process easier the lookup master data frame could include:

source	var	month	url	load_function	extent
chelsa	tmin	jan	...	load_chelsa(tmin,jan)	
chelsa	tmin	feb	...	load_chelsa(tmin,feb)	
chelsa	bio1	NA	...	load_chelsa(bio1,NA)	
worldclim	tmin	jan	...	load_chelsa(tmin,jan)	



```
tif1 <- grab(extent = fagus1, set = "CHELSA", var = "tmin", month = "jan", buf = 1,  
             download = T)
```

Build on top of sdmpredictors:

Our package will have more datasets and we will be able to extract using an extent.

For metadata, have code that creates an .Rdata file. A function, which pulls the metadata in and puts it in an .Rdata file.

Background packages:

raster, sdmpredictors, gfcanalysis