

Changes documentation

Jess

10/08/2021

This is a source document for changes made on the BCCVL scripts for implementation on the EcoCommons R package.

Functions and script common vocabulary follows Hadley Wickham's style guide (which is based of the Google style guide) <http://adv-r.had.co.nz/Style.html>.

If you have any questions please contact me :}

Downloading the package and understanding the folder structure

You can install the package using the following command `devtools::install_github("AtlasOfLivingAustralia/ecocommons")` `library(ecocommons)`

As BIOMOD2 has a A LOT of dependencies, it can take a while... something to think in the future if we decide to make the package available to the public.

After you download it, you will see 4 documents and 3 folders. README and DESCRIPTION explains a bit about the package. The NAMESPACE shows all the functions that are being exported for the package, hiding the internal functions. It also shows a list of the functions used from other packages.

The R folder contains all the functions. The man folder is generated from roxygen2 and shouldn't be changed by hand. The inst folder contains extra files. Including "variables", where are the dataset I used for internal test, "original_source_files", with the original BCCVL script and "dependency_management", with a script to test the amount of dependencies the R package is using.

General Rules

All the functions created within EcoCommons starts with "EC_". The reason is that some of the functions are really similar to existing ones, e.g. `read_raster`. The algorithm scripts that I am turning into functions starts with "EC_modelling_". The "EC_utilities_" are internal functions and subfunctions. Some are specific to geographical models, to maxent, etc

The "EcoCommons_source.R" is the script I am working on as the initial/example script to run everything. The idea is that you open it and follow the script to run the models, at least internally.

In the begging of each R document there is some information on the roxygen format. Please don't remove this part, but feel free to edit the information if needed. Please note that subfunctions that are only used in one function are nested above the main function.

Step-by-step process

1st STEP

I turned functions that were in a big script into individual files and rename them

2nd STEP

Created the package via devtools

3rd STEP

Added roxygen2 documentation for all functions

4th STEP

Improved comments and familiarized with the functions

5th STEP

- Updated packages when possible
- Transformed algorithm scripts into functions
- Removed unused functions

6th STEP

- Test algorithm scripts (internally)
- Set up workflow document

NEXT STEPS

- Test package
- Recognize authorship of previously written R functions
- Check license of the R packages that are in use - Figure out how to implement it on EcoCommons

FUNCTIONS THAT STILL NEED IMPROVEMENT

- EC_performance_2d
- EC_plot_VIP
- EC_save_dismo_eval
- EC_modelling_maxent

Renaming Functions

BCCVL FUNCTIONS

BCCVL FUNCTION NAME = NEW NAME (WHERE TO FIND)

parameter.as.string = EC_parameters_edition (Not exported)

parameter.print = EC_parameters_edition (Not exported)

bccvl.err.null = NA (Deleted)

bccvl.species.read = EC_read_sp (Exported)

bccvl.data.transform = EC_data_projection (Exported)

bccvl.format.outfilename = NA (Deleted)

bccvl.biomod2.formatData = EC_format_biomod2 (Exported)

bccvl.merge.save = EC_merge_save (Subfunction EC_format_biomod2)

bccvl.log.warning = NA (Deleted)

bccvl.raster.load = EC_read_raster (Exported)

bccvl.raster.common.extent = EC_raster_extent (Subfunction EC_raster_stack)

bccvl.raster.extent.to.str = EC_raster_to_STR (Subfunction EC_raster_stack)

bccvl.rasters.common.resolution = EC_raster_resolution (Subfunction EC_raster_stack)

bccvl.rasters.common.reference = EC_raster_ref (Subfunction EC_raster_stack)

bccvl.rasters.warp = EC_raster_warp (Subfunction EC_raster_stack)

bccvl.rasters.to.common.extent.and.resampled.resolution = EC_raster_resampled (Subfunction EC_raster_stack)

bccvl.enviro.stack = EC_raster_stack (Exported)

```

bccvl.remove.rasterObject = EC_raster_remove (Exported)
bccvl.sp.transform = EC_data_projection (Subfunction EC_SDM_geoconstrained)
bccvl.mask = NA (Deleted)
bccvl.sdm.geoconstrained = EC_SDM_geoconstrained (Exported)
generate_background_data = EC_create_background (Subfunction EC_utilities_maxent)
generate_biased_background_data = EC_create_biased_background (Subfunction EC_utilities_maxent)
generate_response_curve = EC_plot_response_curve (Subfunction EC_utilities_maxent)
exploratoryPlots.contin.occ = EC_plot_contin_OCC (Subfunction EC_utilities_maxent)
exploratoryPlots.categ.occ = EC_plot_categ_OCC (Subfunction EC_utilities_maxent)
exploratoryPlots.contin.bg = EC_plot_contin_BG (Subfunction EC_utilities_maxent)
exploratoryPlots.categ.bg = EC_plot_categ_BG (Subfunction EC_utilities_maxent)
saveEvaluationStatistics = NA (Deleted)
computeAreaExtentOccupancy = NA (Deleted)
bccvl.generateThresholdMap = EC_threshold_map (Subfunction EC_utilities_maxent)
bccvl.plotProjection = NA (Deleted)
bccvl.get_filepath = EC_file_path (Exported)
bccvl.saveProjectionImage = NA (Deleted)
bccvl.generateOccurrenceProbChangeMetric = NA (Deleted)
bccvl.generateSpeciesRangeChangeMetric = NA (Deleted)
bccvl.generateCentreOfGravityMetric = NA (Deleted)
bccvl.saveModelProjection = NA (Deleted)
bccvl.save = EC_save (Exported)
bccvl.write.csv = EC_write_csv (Exported)
bccvl.getModelObject = NA (Deleted)
bccvl.grdtogtiff = EC_GRD_to_GTIFF (Subfunction EC_utilities_algorithm)
bccvl.checkModelLayers = NA (Deleted)
family_from_string = NA (Deleted)
grid.info = NA (Deleted)
distance = NA (Deleted)
sample.factor.levels = NA (Deleted)
.SampleFactorLevels.raster = NA (Deleted)
.SampleFactorLevels.data.frame = NA (Deleted)

```

EVAL FUNCTIONS

```

bccvl.saveModelEvaluation = EC_save_model_eval (Exported)
bccvl.saveProjection = EC_save_projection (Exported)
absmean = absmean (on EC_performance_2D)
absdiff = absdiff (on EC_performance_2D)
pick1min = pick1min (on EC_performance_2D)
performance.2D = EC_performance_2D (Exported)
dev.save = NA (Deleted)
bccvl.createMarginalResponseCurves = EC_create_response_curve (Subfunction EC_save_dismo_eval)
bccvl.calculateVariableImpt = EC_calc_variable_impt (Subfunction EC_save_dismo_eval)
bccvl.calculatePermutationVarImpt = EC_calc_permutation (Subfunction EC_save_dismo_eval)
bccvl.saveDISMOModelEvaluation = EC_save_dismo_eval (Exported)
bccvl.saveBIOMODModelEvaluation = NA (Deleted)
bccvl.savePdf = EC_save_pdf (Subfunction EC_plot_VIP)
bccvl.VIPplot = EC_plot_VIP (Exported)

```

NEW FUNCTIONS

```

EC_read_json
EC_utilities_...
EC_modelling_...

```

Syntax rules

Notation and Naming

- Avoid weird separators
- Give your files short and meaningful names
- File names should be unique
- VARIABLES all in lower case with words separated by dots; nouns; avoid single letters
- FUNCTIONS initial capital letters and no dots; verbs

Comments

- Descriptive names can help to avoid unnecessary comments
- Comments should not state the obvious
- Start comments with # and one space; short codes can be placed after the code, with two spaces, #, and then one space
- Functions need special comments, with one sentence description of the function, a list of arguments with a description (including data type) and description of the return value

Syntax

- Always use <- when assigning names to objects and avoid using = for assignment " Even though this distinction doesn't matter for the majority of the time, it is a good habit to use <- as this can be used anywhere, whereas the operator = is only allowed at the top level. In addition = closely resembles ==, which is the logical operator for equals to"
- Avoid long line length (~80 characters)
- Place spaces around all binary operators (=, +, -, <-, ==, !=)
- Always put a space after a comma and never before
- Curley braces opening should never go on its own line and should always be followed by a new line. A closing curly brace should always go on its own line, unless followed by else, which should be contained within outward facing curly braces >}else{. Indent the code within curly braces

PROBLEMS

- JSON files - every time you run it generated one JSON file for each algorithms? That means that if you select all algorithms you will have 17 JSON files? Is there a way to implement everything in one json? I know that some variables are repeated for different algorithms but can have different values
- On the script, it is set to intersect the convex hull with the constrained region, but it doesn't seems to be an option on the platform
- Can we adapt it to run multiple algorithms at the same time? BIOMOD allows it (except for geographical, profiles and BRT)

Ideas to implement in the (near) future

- Add Author(s) per function, and include link to report bugs
- Change raster to terra

- Write up an user case for EcoCommons blog
- Pseudo-absences- same for all algorithms; perform before clipping; Rob is lookinf for alternatives to improve it
- Display default settings for users
- Check for auto-correlation between variables for statistical regression algorithms
- VIP plots per algorithm (making the process faster)
- Filtering of input data (e.g. Galah)
- Check where random seed is used - and how
- Improve outputs in general - nicer graphs, datasheet for all algorithms
- Ways to run multiple algorithms at the same time?