**What’s happening?**

**Folder structure:**

*Working directory:* ~/Dropbox/trade-biodiversity discovery/discovery\_paper\_1

*R*: all scripts

*documentation*: this file and all other documentation

*raw\_data*: all the raw data used in this analysis, don’t change folder structure or else R scripts won’t work

*processed\_data*: all processed data. Contains layers\_sdm for sdm layers, masks for mask layers, layers\_clues for processed clues input layers and processed occurrences csv

**Step 1 - make SDM layers from Raw data.**

*Script*:dp1\_preprocessing.R

Summary:

* There is a document with raw data sources. All raw data are in raw\_data folder.
* Script dp1\_preprocessing.R turns raw data into processed layers in processed\_layers folder.
* Run twice, once for the entire tile29 (to fit models) and once for Vietnam (for clues and SDM predictions)
* SDM layers for vietnam are later reprojected and rewritten for clues, so this is the first step of this analysis.
* More info in code

Naming convention for layers for processed layers

variable\_region\_type\_scenario\_timestep.

Naming condition: character string after variable always has the same length, so we can later go back and do pattern matching etc to subset layers.

*variable:* abbreviation for the variable name

*region:* til for tile29 (the “large” region we fit models on, see worldclim data base) and vnm for Vietnam

*type:* sta, dyn and luc. sta is static, so doesn’t change through time. dyn is dynamic, does change through time (in this case all bioclim under rcp85 scenario), luc is land use (also dynamic but needs different identifier so we can distinguish it easily from other dyn variables).

*scenario:* only important for lu, just placeholder in all other vars (in keeping with naming condition). 000 means no land use scenario applied, which is the case for all layers apart from land use. scenarios ttp and bau always also imply climate change because we produce future land use maps under consideration of climate change in clues.

**Step 2 – make clues layers.**

*Script:* dp1\_clues.R

Summary:

* Load layers produced in step 1
* run correlation analysis between predictor variables (to select uncorrelated variables), reproject to utm for dyna clue, sync NA across all layers
* output layers under clue naming convention, in asc format, in processed\_data/layers\_clues. From there they can be copied straight into the clues program folder.
* We need to output \*.asc layers using SDMtools::write.asc, because raster::writeRaster does something werid to the formatting so that clues can’t read it (or should I say clues has weird requirements for input layers…why not tif?!)
* *Other outputs*: script also does simple regression analysis (lu class ~ variables) and produces the alloc.reg1 file with significant regression parameters as input for clues as well as a file containing the names of significant predictors and a table with original layers names and their clues variable names

File naming for Dyna-clue

Clues needs a specific naming convention for its input layers.

* “Sc1gr\*.fil.asc, where \* is the variable identifier, starting from 0. This goes for each dynanimc and static layer used.
* Dynamic layers additionally follow “Sc1gr\*.\*\*.asc, where \* is variable identifier and \*\* is the time step also starting from 0 up to the number of time steps used. So the first time step of each layer is in the folder twice, under two different names.
* Number of variables depends on the result of correlation analysis.
* regionpavnm.fil.asc, which is a map of protected areas and regionvnm.fil.asc is just a mask of the country
* cov\_all.0.asc is the current land use layer

**Step 3 – make demand trajectories**

*Script:* dp1\_gtap\_trajectories.R

Summary:

* trajectories are needed as clues input – they are in ha. 1 cell = 100 ha. So when you open the lu map in R, the cell count (table(r[]) should be the first demand file row divided by 100.
* 5 classes: 0 crop, 1 grass/shrubland, 2 forest, 3 urban/artificial, 4 barren etc
* Commodity trajectories from Tom and Ha are % change between 2015 and 2225 (or similar), in various changing time steps.
* The script uses 2015 FAO estimates of harvested area in each agricultural commodity in south east Asia to weight the total change of all agricultural sectors by the contribution of each individual commodity. If we would just take the mean between commodities, we’d assume that the land demand for each commodity is equally large, which is completely not right. For example, tomatoes might make for 50% of Spain’s agricultural output, but they only occupy 2% of agricultural land.
* Crop land changes proportionally to the weighted mean change in agricultural commodities
* Grass/shrubland changes proportionally to changes in cattle
* Forest changes inversely proportional to changes in forestry sector, assuming that larger forestry sector means smaller forest land and vice versa.
* Urban changes proportionally from urban population predictions by World Bank
* Barren doesn’t change at all.
* Demand in each year has to sum to the total area of vnm. So what ever is unallocated or overallocated gets deducted or added to grass/shrubland, assuming that this class comprises not just pasture, but also abandoned agriculture and former forestland, or can serve as a source for new forests and agricultural land.

**Step 4 – run clues (can only be run in a Windows OS)**

Add details about the CLUES/Dyna-CLUES software and reference and link for download.

* <https://www.environmentalgeography.nl/site/data-models/data/clue-model/>
* <http://www.ivm.vu.nl/en/Organisation/departments/spatial-analysis-decision-support/Clue/index.aspx>

In clues folder, there are some parameter files I’ll describe here.

Summary:

* This produces land use maps for specified time steps. Download and read clue-s manual from internet to understand how it works.

Required steps:

* Move clues layers from step 2 into folder “clues”
* Move alloc.reg1 file from step 2 into folder “clues”
* Allow.txt file (transition matrix, rows are from cols are to, 1 means transition allowed, 0 means transition not allowed, same order as classes on map:
* Main.1 file - clues parameters: there is a file in documentation folder describing what the different rows mean. I’ve adjusted the parameters, but if clues doesn’t converge you’ll have to start playing around and see what happens. But it does, so let’say it works. (please?). it says 2015 2020 in one line because we have to cheat it into thinking we’re modelling 5 years, although it’s 5 10-year timesteps. It’s how clues works….,…
* ……
* demand.in1 and demand.in2: copy the two demand trajectories in there (demand.1 bau and demand.2 is tpp). They start with a single number in the first row that indicates the total number of time steps, here 6. First row has to be the current demand.
* Now: sync the project dropbox to a windows computer and run the thingo there. Don’t forget to make a generous sacrifice to the clues gods.
* Open clues.exe, select the non-pa mask (regionvnm or whatever) and the demand file you want and it run.
* It produces a landuse map for each year. Cov\_all.5.asc is the final time step here, i.e. 2070. We nee dthis for sdm predictions, so remove this from folder and store somewhere before running the other demand file. Name those files following the SDM layer naming convention and put into processed layers folder. That folder goes up on boab for SDMs.

**Step 5 – make bias layer**

*Script: dp1\_bias\_layer.R*

The script makes the bias layer we need to sample background points. We want to account for sampling bias in the observation data when building models, so we want to sample more background points in areas with high sampling effort and vice versa (detailed explanation in “MaxEnt for ecologists” by JE.). The script should be fairy self-explanatory. Basically, maxent is used to model occurrence points ~ X, with X being layers of anthropogenic drivers, for example distance to roads, built up areas, population density etc. The output is a map of tile 29 that gives relative probabilities of an occurrence point having been sampled in a cell, as a function of those variables. When sampling background points, this map serves to give weights to each cell, so for example we are more likely to get background points in greener areas.

**Step 6 – upload layers and species data to boab and run sdms in maxent.**

*Scripts:*

*dp1\_process\_clues\_output*

*dp1\_sdms.R* (run on boab)

*dp1\_background\_points.R* (run on boab)

*Landuse map:* The clues output map is asc and in utm, we need to convert that into longlat as the rest of the layers. Run *dp1\_process\_clues\_output* to do so.

There is a folder structure set up on boab, about the same as in dropbox. Files are quickly uploaded using filezilla. My deets for the container are skapitza d28838f66

Run dp1\_background\_points.R first, it samples 10000 background points for tile 29, weighted by the bias input layer in the layers\_bias folder. The points are saved on disk, in the output folder. It also does correlation analysis and stores the names of the final predictor set. Both saved files are pulled up in modelling script. Run dp1\_sdms.R, top to bottom. Sometimes, when dismo::maxent doesn’t work, do the commented stuff at the top. It reinstalls dismo and rjava and moves the maxent.jar file into the package folder. You can find out if maxent works by running maxent() and check if there is an error.

Run the script. It does the following:

* **Runs maxent models** for each species three times: first time with jackknifing to idenfity predictor importance, second time with a reduced predictor set and including 5 fold crossvalid, (without preds that had importance < 1% first time around). The results of this cross valid model are also stored and returned. Third time on reduced pred set without cross valid (so a final model on the full data). The philosophy is that there is no sensible way to combine 5 submodels resulting from crossvalid to a final model, because they will all be subpar to the final model that is fitted on the full data set. However, we need the crossvalidation results still.
* **Predicts maxent models** to three time steps: present, “bau” and “tpp”. Those latter two scenarios combine climate change predictions of dynamic varlaibels and future land use time steps.
* **Output:** No more maps!!!! YAY – the script reads info from eaech predicted map – the counted number of cells above maxent’s inbuilt maxSSS threshold. Found a few papers on that, so it’s a good way to do it I think. So we get 3 values of cell counts for three different distribution maps, present, bau and tpp. This output, together with the cross-validated model results and the current iterator (so we can use the species object to track which result belongs to which species), is returned. Final thing at the bottom: Store the output (i.e. the results object produced in iterations).
* This are our raw results, more about what to do with them at a later stage once I get there.

**NOTES**

**On land-use change**

Analysis from 2014 – 2070 (56 time steps)

5 land-use classes: 0 crop, 1 grass/shrubland, 2 forest, 3 urban/artificial, 4 barren

Land-use maps for 2070 under bau and tppp scenarios generated in CLUEs

Scenarios:

* *000* – This is not a scenario but just a current (2015) land-use map from USGS <https://lta.cr.usgs.gov/GLCC>
* *bau* – current trading schemes remain static (get details from TK)
* *tpp* – TPP implemented in 2015 across the participating countries and tariffs are sequentially removed through time (details from TK)