

Information on the BLIZ T1.6 scripts

BLIZ T.1.6 “Data visualisation of simulation output” script [here](#)

Step 1 - Import and merge output files

Lines 30-47 → merge model outputs together from individual folders into a single .tsv file (with all species and scenarios)

Lines 51-53 → import the complete .tsv file created before

Step 2 – Create new variables

Lines 59-86 → calculate **occupancy, abundance change, occupancy change and habitat suitability change variables** and giving columns names

Lines 88-110 → fix species and scenarios names and filtering the dataset to remove *Viola arvensis* and *Eriophorum vaginatum* species

Step 3 → Population level plots

Lines 124-137 → calculate **mean values of abundance, reproduction, carry, habitat and bevmort variables per timestep** and write the dataframe into a .csv file

Lines 158-198 → produce and save **Abundance through time plot**

Lines 201-211 → quantify abundance between t=115 and t=25 and write dataframe to .csv file

Lines 218-241 → produce and save **Reproduction through time plot**

Lines 247-269 → produce and save **Habitat suitability through time plot**

Lines 275- 297 → produce and save **Carrying capacity through time plot**

Lines 303-325 → produce and save **Mortality through time plot**

Lines 332-345 → calculate **abundance mismatch**

Lines 347-503 → produce and save the multiple versions of **Abundance mismatch boxplots**

- Lines 347-359 → Abundance mismatch plots WITH outliers
- Lines 366-384 → Abundance mismatch plots WITHOUT outliers
- Lines 391-420 → Abundance mismatch plots with negative mean values (of abundance mismatch) red and positive mean values blue (without outliers)
- Lines 425-445 → Abundance mismatch plots with gradient, based on mean values, cross all species (without outliers)
- Lines 450-478 → Abundance mismatch plots with a gradient, based on SCALED mean values, for each species (without outliers)
- Lines 483-503 → Abundance mismatch plots with a gradient, based on SCALED mean values, for each species (with outliers)

Step 4 – Cell-level plots

Lines 510-520 → calculate **abundance change variable** and select only the values for t=115

Lines 532-562 → produce and save **Proportion of Abundance change maps**

Lines 565-574 → Quantify abundance change for t=115 and write result to .csv

Lines 581-604 → produce and save **Hotspots of abundance change maps**

Lines 639-680 → produce and save **Correlation plots** with just the trend

BLIZ T1.6 “Sensitivity analysis” script [here](#)

Step 1 – Import and merge output files per sensitivity run

Lines 38-78 → **import, sort and merge into a single dataframe each of the sensitivity runs**

Lines 81-97 → separate a column into multiple one to have import variables in the following steps (species, SR (=sensitivity run) and var changed)

Step 2 – Correct columns names for 095 and 105 sensitivity runs

Lines 106-108 → split the combined 095 dataframe into a list of multiple dataframes, by the variable that was changed

Lines 110-119 → create function to name columns to be original column name plus the name of the changed parameter and amount it varied

(here 105 = +5%, e.g. *abundance_growthrate_105* when we are assessing the abundance values if we changed the growth rate parameter by +5%)

Lines 121-124 → apply function

Lines 125-130 → restore dataframes names (inside the split list) and remove unnecessary variables

Lines 131-132 → specify columns by which we can join all the dataframe inside the split list

Lines 134-136 → combine all the dataframes inside the list into one single dataframe (again)

The previous lines of code are repeated and applied to the 105 dataframe.

Lines 140-142 → split the combined 105 dataframe into a list of multiple dataframes, by the variable that was changed

Lines 144-153 → create a function to name the columns to be the original column name plus the name of the changed parameter and the amount it varied (here 095 = -5%)

Lines 155-158 → apply function

Lines 160-166 → restore dataframes names (inside the split list) and remove unnecessary variables

Lines 167-168 → specify columns by which we can join all the dataframe inside the split list

Lines 170-172 → combine all the dataframes inside the list into one single dataframe (again)

Lines 174-177 → **combine the control, 095 and 105 runs** (with correct column names) into one single dataframe

Lines 179-186 → convert integer64 columns into numeric format (to avoid error messages in the next steps)

Step 3 – Calculate proportions and create sensitivity boxplot

Lines 188-255 → **calculate the necessary proportions** for all variables for the sensitivity analysis plot

Lines 256-257 → write the calculated proportions into a .tsv file (after this step clean the environment, run gc() command and re import the new file. If not Rstudio will show a fatal flaw error message)

Lines 260-262 → re-import the data

Lines 264-265 → subset needed columns for the sensitivity analysis boxplot

Lines 266-268 → transform the data from the wide format to the long format using melt function (pivot_longer does not work with this volume of data)

Lines 270-279 → arrange the labels for the boxplots correctly

Lines 282-303 → produce and save the **Sensitivity analysis plot**