

Weaver User's Manual

This document presents all the information needed related to Weaver's usage. This manual is divided into some sections each one explaining different aspects of the program: the first section explains step by step how to run a simulation; the second one describes the configuration file structure; and the third section lists the main output files.

A. Running a simulation step by step

The user has to follow the next steps in order to run a full instance of a simulation:

Step 1 - The executable file *Weaver3D* must be placed in any directory along with the configuration file *run_params*. It is strictly needed that both files are placed in the same directory, so the program *Weaver3D* will begin by searching for the file named *run_params* just inside the same directory from where the execution was called.

In addition, that one same directory must also contain the subdirectory *lib* including the four files provided with the binaries. Thus, the files location must be similar to the one depicted in the figure 1.

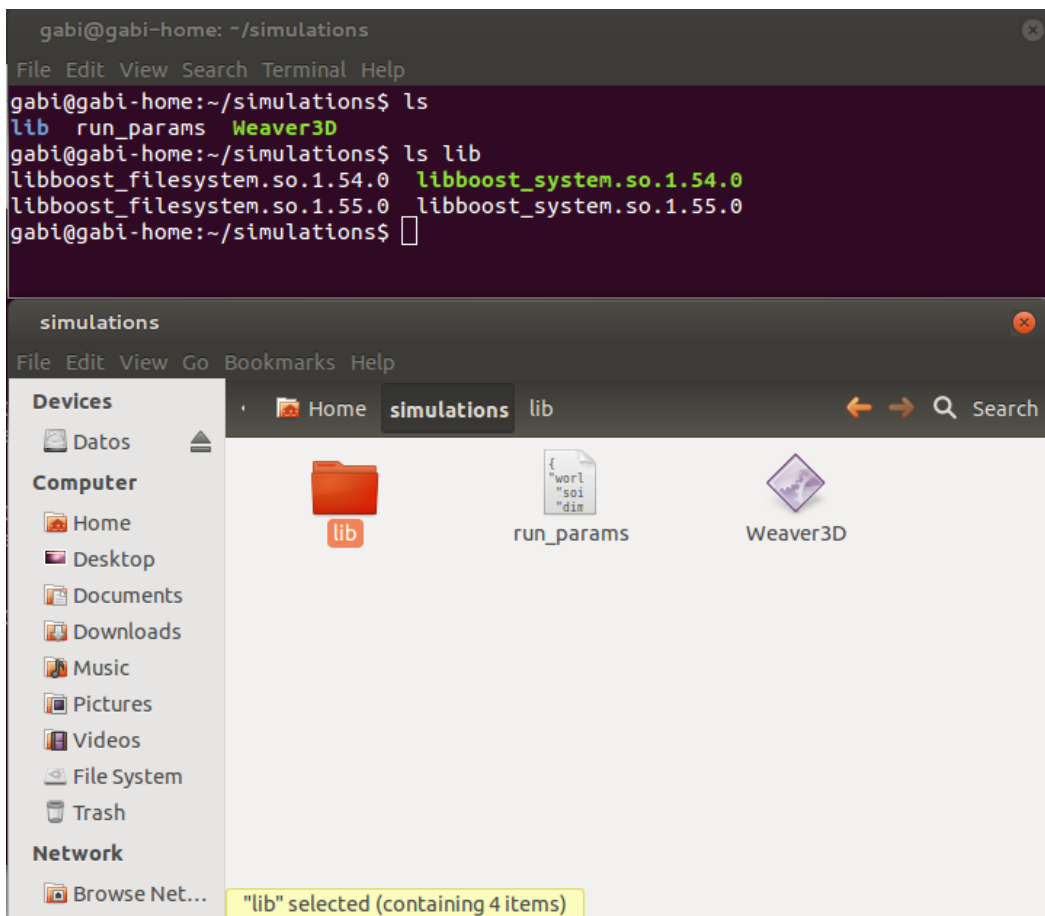


Figure 1 - Files location example

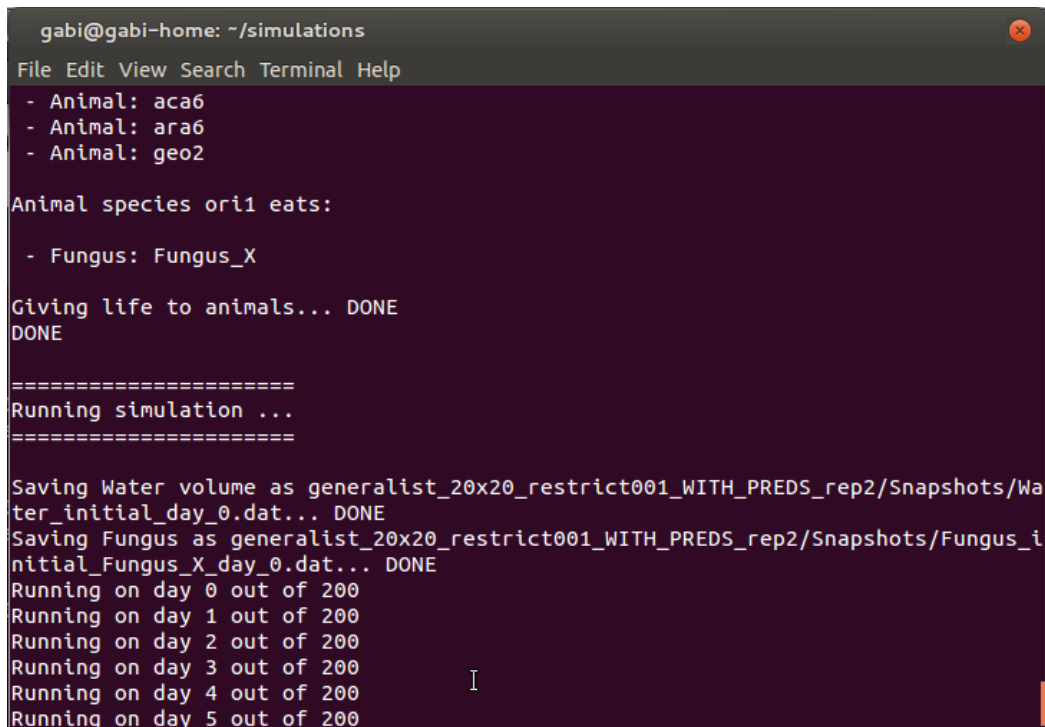
Step 2 - After verifying that we have the two files plus the subdirectory *lib* placed into the same directory, a *Weaver3D* simulation can be run just using the standard execution process: the user has to type in the command `./Weaver3D` without using any added standard input parameter. However, it is recommended to type the following command prior to the execution, so the libraries will be found correctly at running time:

```
export LD_LIBRARY_PATH=/home/<user>/simulations/lib
```

where `<user>` refers to the user name in the computer. This absolute path then corresponds with the *lib* subdirectory mentioned in step 1.

Using this setup, all the information needed is collected at running time from the *run_params* configuration file, which is explained in detail in the section B of this document.

Step 3 - The simulation starts running and the user can read some information about the progress on the standard output, as observed in the figure 2. While the simulation runs some output files are generated in a new subdirectory, which is created inside the very same running directory. This new subdirectory is named as indicated in the configuration file *run_params*, using the parameter *namedoutputDirectory*. A new set of output files (and subdirectories) is then created inside the new subdirectory. They are list in detail in the section C of this document.



```
gabi@gabi-home: ~/simulations
File Edit View Search Terminal Help
- Animal: aca6
- Animal: ara6
- Animal: geo2

Animal species ori1 eats:
- Fungus: Fungus_X

Giving life to animals... DONE
DONE

=====
Running simulation ...
=====

Saving Water volume as generalist_20x20_restrict001_WITH_PREDS_rep2/Snapshots/Wa
ter_initial_day_0.dat... DONE
Saving Fungus as generalist_20x20_restrict001_WITH_PREDS_rep2/Snapshots/Fungus_i
nitial_Fungus_X_day_0.dat... DONE
Running on day 0 out of 200
Running on day 1 out of 200
Running on day 2 out of 200
Running on day 3 out of 200
Running on day 4 out of 200
Running on day 5 out of 200
```

Figure 2 - Running simulation example

NOTE - Simulations can be run in any up-to-date GNU/Linux platform. However, be aware that output files and memory requirements may be quite demanding for your PC depending on the simulation parameters.

B. Configuration file structure

This section explains in detail the usage of every parameter included in the `run_params` configuration file, which is loaded when running each instance of the program. This file is written in an open standard format used to transmit data object: JSON (JavaScript Object Notation). Reading and writing in this format is a simple and intuitive task, and it is also easy to interpret, manipulate and generate the files. JSON syntax and basic data types can be viewed in <http://www.json.org/>.

A fragment from a `run_params` configuration file is shown below. It is relatively easy to notice how the data and components are distributed inside the file. A detailed explanation of each parameter is written after this whole example.

```
{
  "world": {
    "soil": {
      "dimensions": {
        "depth": "10",
        "length": "50",
        "width": "50",
        "cellSize": "1"
      },
      "moisture": {
        "patches": [
          {
            "type": "sphere",
            "radius": "5",
            "xPos": "15",
            "yPos": "15",
            "zPos": "5",
            "value": "87.5"
          },
          (...) (add more patches if needed)
        ]
      },
      "nutrients": {
        "minC": "0",
        "maxC": "23",
        "minN": "0",
        "maxN": "23",
        "minP": "0",
        "maxP": "23"
      },
      "temperature": "18",
      "maxK": "9.6",
      "minK": "0.1",
      "timelapseForChemostatEffect": "1",
      "thresholdForChemostatEffect": "0.05",
      "increaseForChemostatEffect": "1.0"
    }
  }
}
```

```

},
"life": {
  "animals": [
    {
      "name": "aca1",
      "huntingMode": "active_hunting",
      "genetics" : {
        "NumberOfLoci": "20",
        "NumberOfAlleles": "10",
        "NumberOfTraits": "13",
        "traitsPerModule": "3"
      },
      "edibleAnimalSpecies":
["geo1","lit1","opi1","ara1","ara2","ara3","ara4","aca1","aca2","aca3","aca4","col1",
"col2","col3","enc1","enc2","enc3","ori1","ori2","ori3"],
      "edibleFungusSpecies" : [],
      "initialPopulation": "414",
      "restrict": "0.01",
      "correlationCoefficientRHO": "0.1",
      "A": "0.053",
      "B": "2.494",
      "forDensitiesGrowth": "1.4",
      "forDensitiesEggSize": "0.002",
      "ecosystemSize": "20000",
      "minCondition": "0.01",
      "maxVoracityT": "0.100",
      "keepForSurv": "3.0",
      "maxReproductionEvents": "5",
      "assignedForReproduction": "0.9",
      "forControllingDryBodyMass": "1.0",
      "probabilityDeathFromBackground": "0.0",

      "deadlyTank": "0.1",
      "assignedForGrowth": "0.9",
      "numberOfInstars": "4",
      "percentOfTimeSinceLastInstar": "0.9",
      "alphaForPredation": "1.0",
      "minRandomForEncounters": "0.337",
      "maxRandomForEncounters": "0.340",
      "minRandomForPredation": "0.606",
      "maxRandomForPredation": "0.617",

      "maxEncountersT": "10",
      "maxSearchAreaT": "10",

      "Q10phenology": "0.35",
      "Q10digestion": "0.25",

      "meanSizeHunted": "20.08",

```

```

"sdSizeHunted": "7.5",
"meanVorHunted": "13.96",
"sdVorHunted": "4.9",
"meanSpdHunted": "40.65",
"sdSpdHunted": "15",

"meanSizeHunter": "20.08",
"sdSizeHunter": "7.5",
"meanVorHunter": "13.96",
"sdVorHunter": "4.9",
"meanSpdHunter": "40.65",
"sdSpdHunter": "15",
"meanSearchAreaHunter": "0.0",
"sdSearchAreaHunter": "0.0",

"meanSizeXSize": "806.4",
"sdSizeXSize": "275",
"meanVorXVor": "389.6",
"sdVorXVor": "140",
"meanSpdRatio": "25.5",
"sdSpdRatio": "9",
"meanSizeRatio": "397.2",
"sdSizeRatio": "140",

"minTraitsRanges": {
  "energy_tank": "0.25",
  "growth": "1.35",
  "pheno": "3",
  "body_size": "0.001",
  "assim": "0.7",
  "voracity": "0.55",
  "speed": "0.1",
  "search_area": "0.1",
  "met_rate": "0.6",
  "vorQ10": "3",
  "spdQ10": "1.5",
  "srchQ10": "2.0",
  "actE_met": "0.33"
},

"maxTraitsRanges": {
  "energy_tank": "0.5",
  "growth": "1.45",
  "pheno": "11",
  "body_size": "0.003",
  "assim": "0.9",
  "voracity": "0.75",
  "speed": "0.3",
  "search_area": "0.4",

```

```

        "met_rate": "0.8",
        "vorQ10": "4",
        "spdQ10": "2.5",
        "srchQ10": "2.5",
        "actE_met": "0.42"
    },
    "minTraitLimits": {
        "energy_tank": "0.01",
        "growth": "1.01",
        "pheno": "1",
        "body_size": "0.00001",
        "assim": "0.1",
        "voracity": "0.4",
        "speed": "0.05",
        "search_area": "0.05",
        "met_rate": "0.4",
        "vorQ10": "1",
        "spdQ10": "1",
        "srchQ10": "1",
        "actE_met": "0.2"
    },
    "maxTraitLimits": {
        "energy_tank": "1",
        "growth": "2.0",
        "pheno": "100",
        "body_size": "394.7",
        "assim": "1",
        "voracity": "0.8",
        "speed": "0.35",
        "search_area": "0.5",
        "met_rate": "0.9",
        "vorQ10": "6",
        "spdQ10": "3",
        "srchQ10": "3",
        "actE_met": "0.9"
    }
},
(...) (add more species if needed)
],
"fungi": [
    {
        "name": "Fungus_X",
        "sporeMass": "4",
        "minimumFungus": "0",
        "ACTIVATION_ENERGY": "0.68",
        "NORMALIZATION_B": "25.98",
        "minHR": "85",
        "maxHR": "90",

```

```

        "maxRScale": "0.5",
        "zeroFungi": "8",
        "patches": [
            {
                "type": "sphere",
                "radius": "5",
                "xPos": "15",
                "yPos": "15",
                "zPos": "5",
                "value": "1.0"
            },
            (...) (add more patches if needed)
        ]
    }
}
},
"simulation": {
    "runDays" : "200",
    "outputDirectory": "spheres_chemo1_distant_HYPERGENERALIST",
    "saveIntermediateVolumes": "false",
    "saveIntermediateVolumesPeriodicity": "100000",
    "encountersMatrixFilename": "encountersMatrix",
    "predationsMatrixFilename": "predationsMatrix",
    "nodesMatrixFilename": "nodesMatrix",
    "predationEventsOnOtherSpeciesFilename": "predationOnSpecies"
}
}
}

```

Here we explain in detail what is the usage of each parameter:

Dimensions of the world

```

{
  "world": {
    "soil": {
      "dimensions": {
        "depth": "10",
        "length": "50",
        "width": "50",
        "cellSize": "1"
      },
    },
  },
}

```

The world is defined by cells, so far of arbitrary units. "cellSize" refers to one arbitrary unit. For now it should be kept at a fixed value of 1.

"depth", "length" and "width" refers therefore to the 3D dimensions of the simulated ecosystem.

Dimensions of the water-pocket islands

```
"moisture": {
  "patches": [
    {
      "type": "sphere",
      "radius": "5",
      "xPos": "15",
      "yPos": "15",
      "zPos": "5",
      "value": "87.5"
    },
    {
      "type": "sphere",
      "radius": "5",
      "xPos": "35",
      "yPos": "15",
      "zPos": "5",
      "value": "87.5"
    },
    {
      "type": "sphere",
      "radius": "5",
      "xPos": "35",
      "yPos": "35",
      "zPos": "5",
      "value": "87.5"
    },
    {
      "type": "sphere",
      "radius": "5",
      "xPos": "15",
      "yPos": "35",
      "zPos": "5",
      "value": "87.5"
    }
  ]
},
```

Water pockets have also 3D dimensions and may have different shapes and gradients. There exist different ways of defining water in our scenery, "homogeneous", "spherical", "gaussian" and "random_gaussian". All these patches can overlap and there is no virtual limit for the number of them. Just take into account that if two or more patches overlap in space, the maximum value for all of them will be assigned to the affected voxel. Here is an explanation about each type of patch:

- **Homogeneous:** All voxels in the soil will be assigned this same value. Next example shows indicates that all voxels will be initialized to a 25% of water content.

```
"moisture": {
  "patches": [
    {
      "type": "homogeneous",
      "value": "25"
    },...
  ]
}
```

- **Spherical:** All voxels within the defined sphere are assigned the same value. Next example defines a spherical water patch centered on voxel with coordinates x,y,z=(100,100,100) a radius of 10 voxels and a moisture of 25%.

```
"moisture": {
  "patches": [
    {
      "type": "spherical",
      "radius": "10",
      "xPos": "100",
      "yPos": "100",
      "zPos": "100",
      "value": "25"
    },...
  ]
}
```

- **Gaussian:** Similar to the spherical patch but with water values decreasing with radius following a 1D, 2D or 3D gaussian distribution. Therefore, the center of the patch will have a maximum water quantity. The amplitude value tells how much water will exist in the middle of the patch while the sigma value defines how fast this value decreases with radius.

```
"moisture": {
  "patches": [
    {
      "type": "gasussian",
      "radius": "10",
      "xPos": "100",
      "yPos": "100",
      "zPos": "100",
      "amplitude": "25",
      "sigma": "3"
    },...
  ]
}
```

- **Random Gaussian:** Here, we let the program decide the physical distribution of "numberOfPatches" water patches following randomized Gaussian distributions. We can decide whether the sigma values are randomized (through "useRandomSigma" set to "true" or "false") but with a maximum of "maxSigma". Amplitude behaves the same way with "useRandomAmplitude" and a maximum of "maxAmplitude". Location and patch radius will be always randomized.

```
"moisture": {  
  "patches": [  
    {  
      "type": "radomGaussian",  
      "numberOfPatches": "10",  
      "useRandomSigma": "true",  
      "useRandomAmplitude": "true",  
      "maxSigma": "10",  
      "maxAmplitude": "100",  
      "patchesRadius": "35"  
    },...  
  ]  
}
```

In Moya-Laraño et al. 2014 water pockets have a spherical shape ("sphere") with 5 cells of radius. The "xPos", "yPos", "zPos" refer to the position (coordinates) of the center of the sphere in the world. "value" is the relative humidity of the water pocket, which in this version is fixed through the entire simulation. These islands with water available hold the only cells in which fungi can grow. Everywhere else in the world there are no basal resources growing.

Nutrient availability

```
"nutrients": {  
  "minC": "0",  
  "maxC": "23",  
  "minN": "0",  
  "maxN": "23",  
  "minP": "0",  
  "maxP": "23"  
},
```

Not yet in use.

Temperature at which the simulation occurs

```
"temperature": "18",
```

Temperature in Celsius. In this version temperature is constant through the entire simulation.

Carrying capacity of each cell

```
"maxK": "9.6",  
"minK": "0.1",
```

Maximum and minimum fungus carrying capacity per cell. In Moya-Laraño et al. (2014) each cell was set at $0.99 \cdot \text{maxK}$ at initialization.

Chemostat parameters

```
"timelapseForChemostatEffect": "1",
```

How many days elapse between renewal pulses of basal resources (fungi). In Moya-Laraño et al. (2014) this was every day (1).

```
"thresholdForChemostatEffect": "0.05",
```

This last parameter works if and only if "timelapseForChemostatEffect" is set to 0, and then renewal occurs when basal resources are below "thresholdForChemostatEffect" times the initial biomass across the entire world, which is the sum of the $0.99 \cdot \text{maxK}$ across all cells in the world.

```
    "increaseForChemostatEffect": "1.0"  
  },
```

This parameter determines which proportion of the initial basal resources (i.e., sum of the $0.99 \cdot \text{maxK}$ across all cells in the world) are added to the world each time resources are renewed (i.e., at each pulse).

Description of animal parameters

As an example we explain the details of "aca1" (predatory mite species number 1) for the simulation with maximum connectance (0.55) and maximum genetic variability ("restrict" of 0.01) in Moya-Laraño et al. (2014).

```
"life": {  
  "animals": [  
    {  
      "name": "aca1",  
      "huntingMode": "active_hunting",
```

"hunting mode" includes three types of hunting "active hunting". Predators include animals that actively search for their prey or "sit-and-wait" and animals that sit and wait for their prey, such as web building or burrowing spiders. This last feature was used in Moya-Laraño et al. (2013) using mini-Akira (the former R version of Weaver) but is currently not in use. Fungivores are included in the category "does_not_hunt", which forces animals to feed on basal resources only. Omnivory *sensu stricto*; i.e., animals

feeding on both other animals as well as fungi, is not yet implemented. However, trophic level omnivory (animals feeding on more than one trophic level) is allowed within the vector "edibleAnimalSpecies" (see below).

```
"genetics" : {
    "NumberOfLoci": "20",
    "NumberOfAlleles": "10",
    "NumberOfTraits": "13",
    "traitsPerModule": "3"
},
```

"NumberOfLoci" is the number of loci involved in each trait. To calculate the value of each trait for one individual, the values of all these loci are summed up assuming perfect codominance. "NumberOfAlleles" refers to the total number of different alleles present per locus in the entire population. "NumberOfTraits" refers to the total number of traits with quantitative genetic basis for each particular species. "traitsPerModule" is the number of traits involved in each module, for which one can play with the degree of genetic correlation (from pleiotropic effects) among the traits ("correlationCoefficientRHO"). "NumberOfTraits" and "traitsPerModule" must not be modified for now, as this version only supports computation for the current traits. Further details can be seen in Moya-Laraño et al. (2012, 2014).

```
"edibleAnimalSpecies" :
["geo1","lit1","opi1","ara1","ara2","ara3","ara4","aca1","aca2","aca3","aca4","col1","col2","col3","enc1","enc2","enc3","ori1","ori2","ori3"],
```

This is a vector which establishes the directed links of each predator to its prey. Since connectance is maximal in this example, this predator is linked to all other species in the food web.

```
"edibleFungusSpecies" : [],
```

This is a vector which establishes which fungus species each fungivore can feed upon. Since "aca1" is a predator, this vector is empty.

```
"initialPopulation": "414",
```

This is not in use in the current version. Instead, each population is established by allometric equations constrained by body size and instar (Moya-Laraño et al. 2014) and only the size of the entire ecosystem is fixed at initialization ("ecosystemSize").

```
"restrict": "0.01",
```

This is the Phi (ϕ) parameter, which varies between 0 and 1 and establishes the degree of genetic variability in traits. A low value means high genetic variability (Moya-Laraño et al. 2012, 2014). In this version one cannot establish different genetic variability for each of the traits.

```
"correlationCoefficientRHO": "0.1",
```

This parameter varies between 0 and 1 and establishes the degree of genetic correlation among traits within a module. In this version one cannot establish different genetic correlations for different modules. Module structure (positive or negative correlations among traits) is identical as in Moya-Laraño et al. 2012. All simulations in Moya-Laraño et al. 2014 were run with low genetic correlations (0.1).

"A": "0.053",
"B": "2.494",

These are the constant and scaling coefficients for the Mass-Length equation A8 in Moya-Laraño et al. 2014, which relates shape to body mass.

"forDensitiesGrowth": "1.4",
"forDensitiesEggSize": "0.002",

These two last parameters are necessary to calculate how many animals of each instar will be present at initialization following abundance-body size allometric equations (Moya-Laraño et al. 2014). "forDensitiesGrowth" is the midpoint between "minTraitsRanges" "growth" and "maxTraitsRanges" "growth" (see below). "forDensitiesEggSize" is the midpoint between "minTraitsRanges" "body_size" and "maxTraitsRanges" "body_size" (see below).

"ecosystemSize": "20000",

Total number of individuals in the simulation at initialization. How this number is scattered across species and instars is established by allometric abundance-size constraints (Moya-Laraño et al. 2014).

"minCondition": "0.01",

Lower limit for how body condition affects flexible traits in simulations (upper limit is always 1). The body condition state variable (the energy tank ϵ divided by the structural body size B , Moya-Laraño et al. 2012) is interpolated from the maximum range (min=deadlyTank, max=up_tank), where deadlyTank is the energy threshold below which an animal dies ($\epsilon/B = 0.1$) and where up_tank is the upper tank threshold (max ϵ/B) calculated from the mass increase necessary to molt to the next instar, and this changed into the (1, minCondition) scale. Note how this allows animals in better condition to have lower c coefficients in equations A4-A6 (Moya-Laraño et al. 2014).

"maxVoracityT": "0.100",

V_{\max} in equation A7 (Moya-Laraño et al. 2014)

"keepForSurv": "3.0",

coefficient by which the minimum energy available for survival is multiplied and the result subtracted from the energy budget ϵ in order to decide the total budget available for reproduction.

"maxReproductionEvents": "5",

maximum number of egg batches laid per one individual in its lifetime.

"assignedForReproduction": "0.9",

when the animal reaches its maximum instar, this parameter defines the amount of energy needed in order to reproduce. If the animal has enough energy, its life state changes to "reproducing".

"forControllingDryBodyMass": "1.0",

this parameter changes depending on whether the mass-length equation (A8 in Moya-Laraño 2014) has been applied based on wet (1) or dry mass (0.3, assuming a 70% water content).

"probabilityDeathFromBackground": "0.0",

background mortality for reasons others than starvation or predation

"deadlyTank": "0.1",

ϵ/B threshold below which death from starvation occurs

"assignedForGrowth": "0.9",

proportion of accumulated energy (ϵ) which is invested in molting

"numberOfInstars": "4",

Number of instars from egg to adult

"percentOfTimeSinceLastInstar": "0.9",

time threshold beyond which if the target energy for molting has not been reached, the animal molts anyway. It is a fraction of the time spent in the former instar (pp 131-132 in Moya-Laraño et al. 2014).

"minRandomForEncounters": "0.337",

"maxRandomForEncounters": "0.340",

"minRandomForPredation": "0.606",

"maxRandomForPredation": "0.617",

P-values to decide encounters and predation (p 134 in Moya-Laraño et al. 2014).

"maxEncountersT": "10",

Upper limit of encounters with predators in one day. This is used to establish energy and assimilation efficiency losses, as well as tuning activity (anti-predatory behaviour).

Above this number of encounters the effects apply according to the limits of the linear interpolation (see Moya-Laraño 2012, 2014 for details).

"maxSearchAreaT": "10",

A_{\max} in equation A7 (Moya-Laraño et al. 2014)

"Q10phenology": "0.35",

Developmental time and therefore birth date (phenology) is affected by this parameter depending on temperature using linear interpolation (see Moya-Laraño 2012 for further details).

"Q10digestion": "0.25",

Currently not in use

"meanSizeHunted": "20.08",
"sdSizeHunted": "7.5",
"meanVorHunted": "13.96",
"sdVorHunted": "4.9",
"meanSpdHunted": "40.65",
"sdSpdHunted": "15",

"meanSizeHunter": "20.08",
"sdSizeHunter": "7.5",
"meanVorHunter": "13.96",
"sdVorHunter": "4.9",
"meanSpdHunter": "40.65",
"sdSpdHunter": "15",
"meanSearchAreaHunter": "0.0",
"sdSearchAreaHunter": "0.0",

"meanSizeXSize": "806.4",
"sdSizeXSize": "275",
"meanVorXVor": "389.6",
"sdVorXVor": "140",
"meanSpdRatio": "25.5",
"sdSpdRatio": "9",
"meanSizeRatio": "397.2",
"sdSizeRatio": "140",

means and standard deviations to normalize traits and trait products to include them in equations A9 and A10 (Moya-Laraño et al. 2014)

"minTraitsRanges": {
 "energy_tank": "0.25",
 "growth": "1.35",

```

    "pheno": "3",
    "body_size": "0.001",
    "assim": "0.7",
    "voracity": "0.55",
    "speed": "0.1",
    "search_area": "0.1",
    "met_rate": "0.6",
    "vorQ10": "3",
    "spdQ10": "1.5",
    "srchQ10": "2.0",
    "actE_met": "0.33"
  },
  "maxTraitsRanges": {
    "energy_tank": "0.5",
    "growth": "1.45",
    "pheno": "11",
    "body_size": "0.003",
    "assim": "0.9",
    "voracity": "0.75",
    "speed": "0.3",
    "search_area": "0.4",
    "met_rate": "0.8",
    "vorQ10": "4",
    "spdQ10": "2.5",
    "srchQ10": "2.5",
    "actE_met": "0.42"
  },

```

Standard trait ranges (L_x , U_x) from which actual trait ranges are obtained in combination with the parameter "restrict" (ϕ) above (Moya-Laraño et al. 2014 p 124).

```

  "minTraitLimits": {
    "energy_tank": "0.01",
    "growth": "1.01",
    "pheno": "1",
    "body_size": "0.00001",
    "assim": "0.1",
    "voracity": "0.4",
    "speed": "0.05",
    "search_area": "0.05",
    "met_rate": "0.4",
    "vorQ10": "1",
    "spdQ10": "1",
    "srchQ10": "1",
    "actE_met": "0.2"
  },
  "maxTraitLimits": {

```



```

        "energy_tank": "1",
        "growth": "2.0",
        "pheno": "100",
        "body_size": "394.7",
        "assim": "1",
        "voracity": "0.8",
        "speed": "0.35",
        "search_area": "0.5",
        "met_rate": "0.9",
        "vorQ10": "6",
        "spdQ10": "3",
        "srchQ10": "3",
        "actE_met": "0.9"
    },
    },

```

Evolutionary trait limits (Π_x , K_x), beyond which trait values cannot evolve (Moya-Laraño et al. 2014 p 124). Any individual surpassing this threshold for any trait is locked at the limit value for the trait.

Output choices

```

"runDays" : "200",
  "outputDirectory": "spheres_chemo1_distant_HYPERGENERALIST",
  "saveIntermediateVolumes": "false",
  "saveIntermediateVolumesPeriodicity": "100000",
  "encountersMatrixFilename": "encountersMatrix",
  "predationsMatrixFilename": "predationsMatrix",
  "nodesMatrixFilename": "nodesMatrix",
  "predationEventsOnOtherSpeciesFilename": "predationOnSpecies"

```

C. Output files list

Each time a simulation is run, a bunch of output files can be generated. The user decides where to store files and which ones to keep or discard through JSON directives.

Next there is a description of the most important files that can be obtained from a Weaver simulation:

- ***animal_constitutive_traits.txt*** - The information related to each animal traits that is born is kept in this file. This is used to study genetic variability of populations and its evolution. Currently, 13 traits are stored for each animal: **energy**, **growth**, **pheno**, **body**, **assim**, **vor**, **speed**, **search**, **met**, **vorQ10**, **spdQ10**, **srchQ10**, **e_met**. Other information is also present in this file, including its unique identifier, (**id**), the species the animal belongs to (**species**), generation based on parents ones (**g_numb_prt1** and **g_numb_prt2**), and parents unique identifiers (**ID_prt1** and **ID_prt2**).

- **extendedDailySummary.txt** - A daily summary is kept detailing, for each species, the number of individuals that fall within a given phenological state. Basal resources are also detailed (i.e. fungus biomass). The fields for this data file are: simulation day (**day**), fungus biomass, as **scientificName_biomass**, and the number of individuals at any phenological state as **scientificName_X**; where **X** represents the state (0-to be born, 1-active, 2-starved to death, 3-predated, 4-reproducing, 5-background death, 6-natural death).
- **animals_each_day_end** - This is the most complete set of information regarding each individual. A file is kept for each day containing detailed information for them. Its structure is as **animals_day_X.txt** where **X** represents a given day. Each file is written after a simulation day is finished and contains a summary of each animal currently participating in the simulation: unique individual id (**id**), its species (**species**), spatial location (**x, y, z**), phenological state (**state**), instar for reproduction purposes (**instar**), the initial phenology value (**pheno_ini**), date of creation (**date_egg**), first reproduction date (**age_first_rep**), number of reproduction episodes (**rep_count**), number of offspring to date (**fecundity**), death date (**date_death**), its generation number based on parents ones (**g_numb_prt1** and **g_numb_prt2**), parents unique ids (**ID_prt1** and **ID_prt2**), the number of encounters with predators this day (**encounters_pred**), the accumulated number of encounters with predators along its life (**global_pred_encs**), time to finish digestion (**days_digest**) and all variables regarding genetic traits that can change values along the simulation (**energy, growth, pheno, body, assim, vor, speed, search, met, vorQ10, spdQ10, srchQ10, e_met**).
- **Matrices** - This file is generated upon completion of the simulation and its called **predationOnSpecies**. It stores a NxN matrix where N represent the number of different animal species. This matrix stores an all-to-all predation index with predators forming columns and preys forming rows.

References:

Moya-Laraño, J., Verdeny-Vilalta, O., Rowntree, J., Melguizo-Ruiz, N., Montserrat, M., Laiolo, P. (2012). **Climate Change and Eco-Evolutionary Dynamics in Food Webs**. *Advances in Ecological Research*, 47, 1.

Moya-Laraño, J.; Foellmer, M.W.; Pekár, S.; Arnedo, M.A.; Bilde, T.; Lubin, Y. (2013). **Evolutionary ecology: linking traits, selective pressures and ecological functions**. in: Spider research in the XXI century: trends and perspectives. ed.: D. Penney. Siri Scientific Press. Manchester, UK; pp 122-153.

Moya-Laraño, J., Bilbao-Castro, J. R., Barrionuevo, G., Ruiz-Lupi3n, D., Casado, L. G., Montserrat, M., Meli3n, C. J., Magalh3es, S. (2014). **Eco-evolutionary spatial dynamics: rapid evolution and isolation explain food web persistence**. *Advances in Ecological Research*, 50, 75-144.