

Default GMSE data structures

GMSE: an R package for generalised management strategy evaluation (Supporting Information 7)

A. Bradley Duthie^{1,3}, Jeremy J. Cusack¹, Isabel L. Jones¹, Jeroen Minderman¹, Erlend B. Nilsen², Rocío A. Pozo¹, O. Sarobidy Rakotonarivo¹, Bram Van Moorter², and Nils Bunnefeld¹

[1] Biological and Environmental Sciences, University of Stirling, Stirling, UK [2] Norwegian Institute for Nature Research, Trondheim, Norway [3] alexander.duthie@stir.ac.uk

The most important (default) GMSE data structures

The default submodels of GMSE (`resource`, `observation`, `manager`, and `user`) use a small number of default data structures to hold the information needed in simulations. While these default submodels do not necessarily need to be used in every run in GMSE (see use of `use of gmse_apply`), they will be used in any run of `gmse`, and in any call of `gmse_apply` that does not run with entirely custom submodels. Simulation and model inference does not require an understanding of the default data structures, but such an understanding can be especially useful when running `gmse_apply` if there is a need to extract uncommonly used information, change key simulated values (e.g., landscape properties, agent budgets, or resource movement rules as in [Supporting Information 4](#)), or build custom individual-based submodels. Here we provide a brief explanation of the following key data structures (each name below is listed as named in the output `gmse_apply` when `get_res = "Full"`).

1. `AGENTS`
2. `resource_array` (or `RESOURCES`)
3. `observation_array` (or `OBSERVATION`)
4. `manager_array` (or `COST`)
5. `user_array` (or `ACTION`)
6. `LAND`

Note that these are not the only data structures used in GMSE, but they are the only ones that can potentially be usefully modified in GMSE v0.4.0.3 (see, e.g., [Supporting Information 4](#)), so they are the ones that we focus on here. Additionally, any custom subfunction that returns an array rather than a single value should adhere to the same structure as these defaults if any default GMSE functions are to be used in `gmse_apply`. We can investigate each data structure by running a single simulation of `gmse_apply`.

```
sim <- gmse_apply(get_res = "Full");
```

The full list output of `sim` holds each structure by name (in the case where two names are used, e.g., `resource_array` and `RESOURCES`, both are identical, but the lower case `resource_array` takes precedence in case of a change). Each data structure can be examined, changed, and incorporated into a new simulation (e.g., `new_sim <- gmse_apply(old_list = sim)`).

1. AGENTS

The `AGENTS` data structure is a two dimensional array with a fixed number of 17 columns and a number of rows that is always equal to the total number of manager and users (each row is an individual agent).

```
print(sim$AGENTS);
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,]    1    0    0    0   78   82   50    0   10    0   43    0    0
## [2,]    2    1    0    0   25    3   50    0   10    0    0    0    0
## [3,]    3    1    0    0   76   14   50    0   10    0    0    0    0
## [4,]    4    1    0    0   80   88   50    0   10    0    0    0    0
## [5,]    5    1    0    0   21   53   50    0   10    0    0    0    0
##      [,14] [,15]      [,16] [,17]
## [1,]      0      0 9521.469 1000
## [2,]      0      0   0.000 1000
## [3,]      0      0   0.000 1000
## [4,]      0      0   0.000 1000
## [5,]      0      0   0.000 1000
```

In the default case above, there are five agents (one manager and four user), each represented by a unique row. Columns in the array represent the agent traits listed below.

1. ID (each agent gets a unique number)
2. Type 1 (0 indicates the manager; 1 indicates users)
3. Type 2 (currently unused)
4. Type 3 (currently unused)
5. x-location on the landscape (typically ignored)
6. y-location on the landscape (typically ignored)
7. Movement distance (typically ignored)
8. Time parameter (typically ignored)
9. Distance of vision (currently used only for managers)
10. Error parameter (currently unused)
11. Resource marking parameter (currently used only for managers)
12. Resource tally parameter (currently used only for managers)
13. Unused column 1
14. Unused column 2
15. Unused column 3
16. Yield from owned land (zero for users when default `land_ownership = FALSE`)
17. Budget

It is obvious from the above list that most columns represent traits that are either typically ignored or currently not in use. This is intended to allow for easier future development of default model options and potential customisation of submodels in `gmse_apply`. We anticipate that future versions of GMSE will contain multiple user types with unique traits and among-user interactions.

2. resource__array

The `resource_array` (also accessible as `RESOURCES`) is a two dimensional array with a fixed number of 20 columns and a number of rows that is always equal to the total number of resources (each row is an individual resource). In the above simulation, `sim$resource_array` includes 1114 rows, so we only print out the first eight for illustration.

```
print(sim$resource_array[1:8,]);
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,]    1    1    0    0   23   70   20    1    0   0.3    2    2    0
## [2,]    2    1    0    0   11   90   20    1    0   0.3    0    4    0
## [3,]    3    1    0    0   30    8   20    1    0   0.3    0    2    0
## [4,]    4    1    0    0   20   28   20    1    0   0.3    0    2    0
## [5,]    5    1    0    0   51   21   20    1    0   0.3    1    4    0
## [6,]    6    1    0    0   65   91   20    1    0   0.3    0    3    0
```

```

## [7,]      7      1      0      0     13     77     20      1      0     0.3      0      4      0
## [8,]      8      1      0      0     62      1     20      1      0     0.3      0      2      0
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20]
## [1,]      0     0.5      0      0      0      0      0
## [2,]      0     0.5      0      0      0      0      0
## [3,]      0     0.5      0      0      0      0      0
## [4,]      0     0.5      0      0      0      0      0
## [5,]      0     0.5      0      0      0      0      0
## [6,]      0     0.5      0      0      0      0      0
## [7,]      0     0.5      0      4      0      0      0
## [8,]      0     0.5      0      0      0      0      0

```

Columns in the resource array represent the individual resource traits listed below.

1. ID (each resource gets a unique number)
2. Type 1 (currently all resources are of type 1)
3. Type 2 (currently unused)
4. Type 3 (currently unused)
5. x-location on the landscape
6. y-location on the landscape
7. Movement distance
8. Time parameter (typically ignored)
9. Removal (i.e., death) probability
10. Growth (i.e., birth) probability
11. Offspring produced
12. Age (initial resources are given a random age between 1 and the maximum age sampled from a uniform distribution; offspring always start at age zero in their time step of birth)
13. Marking indicator (used in the observation function)
14. Tallying indicator (used in the observation function)
15. Proportion of a landscape cell the resource consumes in a time step
16. Has the resource been scared by an agent?
17. Has the resource been culled by an agent?
18. Has the resource been castrated by an agent?
19. Has the resource's growth rate been increased by an agent?
20. Has the resource's offspring production been increased by an agent?

In the case of columns 16-20, the value is either zero (if no action has occurred), or some positive integer that matches the ID of the agent that has performed the act (e.g., if column 17 equals 3, then that means that the agent with ID 3 culled the resource in the corresponding row; where more than one agent's action is possible per time step – as in scaring – the integer reflects the most recently acting agent). We anticipate that future versions of gmse will contain multiple resource types, and might add rows to include additional resource traits.

2. observation_array

The **observation_array** (also accessible as **OBSERVATION**) is a two dimensional array, the number of rows and columns of which depend on the type of observation being made (i.e., **observe_type**, which can take integer values from 0-3; see the [GMSE reference manual](#) for more information). For the simulation run above using default parameter values, **observation_array** has 22 columns, two more than **resource_array**. The number of rows is equal to or less than that of **resource_array**; each resource that is observed at least once is placed into a unique row, while unobserved resources are not included as rows in the **observation_array**. In **sim**, there are 43 rows, meaning that 1071 were not observed at all in this time step. Below we print out the first eight rows of the observation array.

```
print(sim$observation_array[1:8,]);
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,]   79    1    0    0   76   83   20    1    0    0.3    0    5    1
## [2,]   90    1    0    0   78   77   20    1    0    0.3    0    3    1
## [3,]  107    1    0    0   68   74   20    1    0    0.3    1    3    1
## [4,]  124    1    0    0   82   85   20    1    0    0.3    0    5    1
## [5,]  133    1    0    0   68   90   20    1    0    0.3    0    4    1
## [6,]  147    1    0    0   71   78   20    1    0    0.3    0    5    1
## [7,]  179    1    0    0   83   83   20    1    0    0.3    0    4    1
## [8,]  197    1    0    0   81   81   20    1    0    0.3    0    4    1
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22]
## [1,]    0  0.5    0    0    0    0    0    0    1
## [2,]    0  0.5    0    0    0    0    0    0    1
## [3,]    0  0.5    0    0    0    0    0    0    1
## [4,]    0  0.5    0    0    0    0    0    0    1
## [5,]    0  0.5    0    0    0    0    0    0    1
## [6,]    0  0.5    0    0    0    0    0    0    1
## [7,]    0  0.5    0    0    0    0    0    0    1
## [8,]    0  0.5    0    0    0    0    0    0    1
```