

Reproducibility of Results for *The effect of mutation rate on host range expansions & shifts*

Katherine Skocelas¹, Ira Woodrig¹, Julia Zhang¹

¹ Computer Science & Engineering, Michigan State University, East Lansing, Michigan, USA

1 Software

All experiments were produced with the Modular Agent Based Evolution (MABE) software package [1]. The MABE framework allows researchers to programmatically construct custom components for organisms, genomes, brains, worlds and other entities (figure 1). MABE is coded in the C++ language.

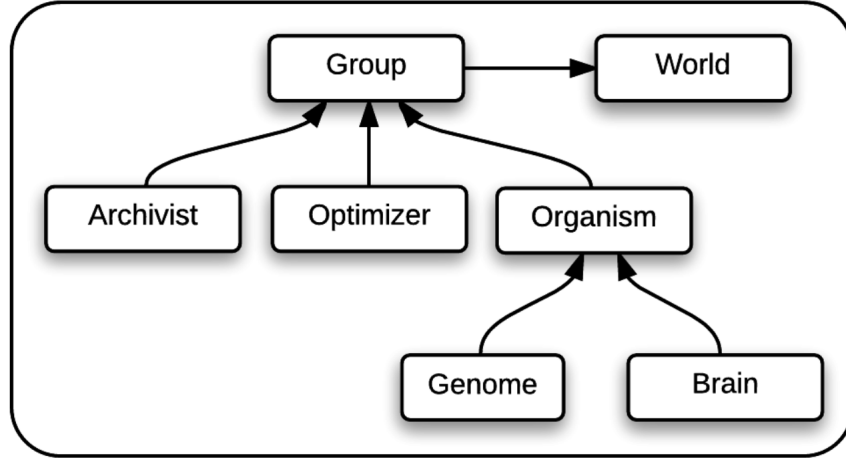


Fig 1. MABE components

Normal MABE operation allows for the design of a custom Genome and Brain for Organisms that are then placed in a customized World. However, project specifications for this particular set of experiments dictated a different approach, due to the following reasons:

- the organisms did not require brains
- the genome for the organisms were simple bit strings

Therefore, to increase efficiency and decrease overhead, a custom World was created that housed definitions for simple Host and Parasite organisms. The authors called this World "HHPWorld" for Host-Host-Parasite World. As this project required multiple simple hosts, a tag was applied to each Host instance to indicate if the Host were of type one or type two (type "Cat" or type "Dog" in our experiments).

This custom world is governed by a large number of parameters, a description of which are included in Table 1. All source code for HHPWorld can be found at <https://github.com/irawoodring/MABE>.

2 Building the Code

Obtaining and working with the code for this project requires use of the `git` version control tool ¹. A detailed use of how to use this tool is beyond the scope of this document, but the authors will provide the most basic commands for acquiring and building the code.

Providing `git` has been installed the repository may be cloned to the users local directory with the command

```
git clone https://github.com/irawoodring/MABE.git
```

Users should then change directory to the MABE directory. The code for HHPWorld is housed in the development branch of this repository, so users will then need to checkout that branch with the command

```
git checkout development
```

Source code for HHPWorld can then be found in the directory titled `World/HHPWorld`.

MABE provides a fairly sophisticated build system comprised of python and autotools scripts. To customize the MABE build, a user modifies the `buildOptions.txt` file to include the modules desired. The `buildOptions.txt` file included with this repository is preset with values that work with HHPWorld.

To invoke the build system users may execute the `pythonTools/mbuild.py` script. This script provides a number of parameters for building, such as which compiler to use, whether to cleanup unnecessary files after the build, parallel building, etc. To build the system with just default options, users may invoke the command

```
python pythonTools/mbuild.py
```

Note that MABE requires use of the Python programming language, version 3 ².

3 Running the Code

After a successful build users may run the simulation by invoking the resulting `./mabe` or `mabe.exe` executable (depending on Unix or Windows based system). As noted previously in this document, HHPWorld is controlled through a parameterization system. Parameters are passed to MABE via three files:

- `settings.cfg` - Global MABE settings.
- `settings_organism.cfg` - Unused for HHPWorld, as the organisms are part of the world.
- `settings_world.cfg` - Settings specific to the world, such as amount of resources, size, etc. For HHPWorld also houses Host and Parasite settings.

MABE also provides a mechanism for creating and submitting jobs to a SLURM workload management system ³. This is essential for completing a large number of runs without direct user intervention. MABE provides the `pythonTools mq.py` script for submitting SLURM jobs. A detailed usage of SLURM is beyond the scope of this document.

¹<https://git-scm.com/>

²<https://www.python.org/download/releases/3.0/>

³<https://slurm.schedmd.com/documentation.html>

Table 1. HHPWorld Parameters

Parameter	Description
ParasiteRelationship	Controls whether parasites remove all resources gained from host, if they are commensalist, or if they are mutualist.
resourceInflow	The amount of resource at each location in the world per update. Note that this is per host type, so if the value is 10, there will be 10 resources for Cats and 10 for Dogs at each location.
{cat,dog}FleaFeedAmt	A percentage that controls how much one parasite will take from a host with a perfect bitstring match.
{cat,dog,flea}LifeMin	Minimum host lifespan.
{cat,dog,flea}LifeMax	Maximum host lifespan.
{cat,dog}MaxToShareWithFleas	The total percentage of resources that can be taken by fleas on a host.
reproductionThreshold{Cat,Dog,Flea}	The required resources before a host can reproduce.
init{Cat,Dog}Lock	Controls initialization of host bitstring lock. May be all_0, all_1, or random.
init{Cat,Dog}LockLength	Initial length of bitstring genome.
initFleaKey	Controls initialization of flea bitstring key. May be all_0, all_1, or random.
initFleaKeyLength	Initial length of bitstring genome.
ParasitesStartOnCats	The percentage of parasites that start on the Cat host type.
parasiteInheritance{Cat,Dog}	The percentage of parasites that migrate to a Host's offspring.
rateOfContactAcrossSpecies	The rate of contact between two hosts of arbitrary type in a single location of the world.
rateOfContactSpecies	The rate of contact between two hosts of the same type in a single location of the world.
rateOfTransfer	Upon contact, the percentage of parasites transferred.
mutationPoint{Cat,Dog,Flea}	The point flip mutation rate per host.
mutationSize{Cat,Dog,Flea}	The insertion/deletion mutation rate per host.
moveRate{Cat,Dog}	How often a host moves.
turnRate{Cat,Dog}	How often a host switches direction.
worldWraps	Controls world wrapping.
world{X,Y}	Size of the world.
cullParasitesTo	Controls the maximum number of parasites allowed on a host before parasites on that host are no longer allowed to reproduce.
mutationErateRate	The deletion rate of insertion/deletion mutations.
scoreList	Controls what percentage of resources a parasite acquires based on a percentage match of its key bitstring with its host lock.

4 Result Files

Upon completion of a simulation, HHPWorld outputs a number of files. These files are of the types:

- HHP_Report.csv - details overall statistics of the world for every 100 updates (details in Appendix B).
- HHP_cats,dogs,fleas_XXXX.csv - details of the Cat, Dog, or Parasite population at update XXXX. This is a complete list of each Host or Parasite's ID, the length of the genome, and the host's genome at that update.
- cat,dog,flea_snapshot_XXXX.csv - time of birth and ancestry information for each Host of Parasite at timestamp XXXX.

Table 2 provides details of the columns from the HHP_Report.csv file. The other file types are described above, and no further detail is required.

MABE provides a rudimentary graphing tool, `pythonTools\mgraph.py` should users want to view "quick" results. Details of its usage may be found by running the tool with the `--help` flag.

References

1. Bohm C, G NC, Hintze A. MABE (Modular Agent Based Evolver): A framework for digital evolution research. Artificial Life Conference Proceedings. 2017;29:76–83. doi:10.1162/isal_a016.

Table 2. Description of the columns in the HHP_Report.csv files.

Column Name	Description
update	The current update timestamp of the simulation.
{cats,dogs,fleas}	The current population per each host type.
FpC,FpD	An average of the fleas per Cat or fleas per Dog
{cat,dog}LockAveSize, fleaKeyAveSize	The current average length of the bitstring genome for all cats, dogs, or fleas
fleaKeyAveSizeOn{Cats,Dogs}	The current average length of a parasites bitstring genome, per host type
{cat,dog,flea}Births	The number of new hosts per type since the last update
{cat,dog,flea}Deahts	The number of hosts lost (per type) since the last update
fleaJumps	The number of fleas that have migrated to a new host.
fleaJumpsAcross	The number of fleas that have migrated to a new host type.
fleasInherited	The number of fleas that have been passed down from a parent to a child host.
{cat,dog}CollectedResource	The amount of resources acquired by that host type since the last update.
fleaOn{Cat,Dog}CollectedResource	The amount of resources acquired by fleas per type of host.
bitMatch{Cat,Dog}	An average of how well a flea key matches a host lock, by percentage correct match.
{cat,dog}Theft	The percentage of resources taken by parasites from the host type.
infected{Cats,Dogs}Rate	The percentage of the hosts of each type that have been infected by a parasite.