# AgModel

## An *Agent-Based Model* of the forager-farmer transition

[![DOI](data:image/svg+xml; charset=utf-8;base64,)](https://doi.org/10.5281/zenodo.7909271)

AgModel is an agent-based model of the forager-farmer transition. The model consists of a single software agent that, conceptually, can be thought of as a single hunter-gather community (i.e., a co-residential group that shares in subsistence activities and decision making). The agent has several characteristics, including a population of human foragers, intrinsic birth and death rates, an annual total energy need, and an available amount of foraging labor. The model assumes a central-place foraging strategy in a fixed territory for a two-resource economy: cereal grains and prey animals. The territory has a fixed number of patches, and a starting number of prey. While the model is not spatially explicit, it does assume some spatiality of resources by including search times.

Demographic and environmental components of the simulation occur and are updated at an annual temporal resolution, but foraging decisions are “event” based so that many such decisions will be made in each year. Thus, each new year, the foraging agent must undertake a series of optimal foraging decisions based on its current knowledge of the availability of cereals and prey animals. Other resources are not accounted for in the model directly, but can be assumed for by adjusting the total number of required annual energy intake that the foraging agent uses to calculate its cereal and prey animal foraging decisions. The agent proceeds to balance the net benefits of the chance of finding, processing, and consuming a prey animal, versus that of finding a cereal patch, and processing and consuming that cereal. These decisions continue until the annual kcal target is reached (balanced on the current human population). If the agent consumes all available resources in a given year, it may “starve”. Starvation will affect birth and death rates, as will foraging success, and so the population will increase or decrease according to a probabilistic function (perturbed by some stochasticity) and the agent’s foraging success or failure. The agent is also constrained by labor caps, set by the modeler at model initialization. If the agent expends its yearly budget of person-hours for hunting or foraging, then the agent can no longer do those activities that year, and it may starve.

Foragers choose to either expend their annual labor budget either hunting prey animals or harvesting cereal patches. If the agent chooses to harvest prey animals, they will expend energy searching for and processing prey animals. prey animals search times are density dependent, and the number of prey animals per encounter and handling times can be altered in the model parameterization (e.g. to increase the payoff per encounter). Prey animal populations are also subject to intrinsic birth and death rates with the addition of additional deaths caused by human predation. A small amount of prey animals may “migrate” into the territory each year. This prevents prey animals populations from complete decimation, but also may be used to model increased distances of logistic mobility (or, perhaps, even residential mobility within a larger territory).

If the agent chooses to consume cereals, then they extend time searching for a cereal patch and processing the cereals grains in that patch. Human selection preferences will drive evolutionary processes within the overall cereal population, and the selection rate be altered as a parameterization of the simulation. Two characteristics are selected for: plant morphology (which affects both grain size and ease of harvesting), and patch density (reseeding density). These are generalized into two intrinsic populations of cereals: “wild type,” and “domestic type.” Human harvesting will increase the overall ratio of domestic type cereal plants in utilized patches over time due to artificial selection in patches that are under consistent use. The ratio will begin to decrease in patches that are not used according to a diffusion rate, and the proportion of wild to domesticated cereals in all patches. Morphologically wild cereals produce fewer kcal per seed, has longer handling times, and produces fewer seeds per patch. Thus, as cereal is more frequently used, its desirable traits are enhanced (via selection), and it becomes more profitable. At the same time, however, there is a constant “diffusion” of wild-type characteristics back to patches so that if cereal is consumed less frequently, the proportion of wild type cereals will begin to increase in patches that are not used. Currently, cereal patches are accessed in the same order every year, so humans are assumed to consistently choose the “best” patches first, if/when they decide to consume cereals in a given year. The “size” of individual cereal patches is set by the maximum wild/domesticated cereal patch density values (default values assume patch size = 1 ha).

I have written AgModel in pure Python with the hope of better integration to scientific Python (e.g., pandas, matplotlib) and the open-science movement. The model has a simple graphical interface, or can be run “headless.” Optionally, experiments using parameter sweeps with repetition can be set up to run in parallel on as many available processors as desired. Careful parameterization of AgModel can test scenarios where complex hunter-gatherers engaged in foraging decisions that interacted with plant evolutionary dynamics to artificially increase the prevalence of domestic type plans in the local ecosystem. When the domestic phenotype is most prevalent, we can consider the species to be “domesticated.” The temporal dynamics of the process of domestication, however, are not linear, and the timeline and longevity of the domestication event may differ under changing environmental conditions.

## Installation and Running

You will first need to install Python 3 to run it. Depending on your operating system, you may need to use something like Anaconda to do this. On Linux you can use your package manager.

You will then need to install the following Python modules:

* NumPy
* Pandas
* MatPlotLib
* seaborn
* EasyGUI

If you use the [PIP Python installer](github.com/pypa/pip), which is my preferred method of installation, you can try the following from the terminal:

pip3 install -U numpy pandas matplotlib seaborn easygui

Once these are all installed, you just place the script in a folder of your choosing, open a terminal window in that same directory (you can often do this from the “right click” pop-up menu), and type python3 AgModel-xx.py (where xx is the current version number). The first window that will pop up will ask for a configuration file. I include a sample config file in this github repo that will parametrize the model with reasonable default values. These default values that will also populate the fields if you choose to create a new config file. Then, there will be two windows showing you the variables, and allowing you to change them. Anything you change will be saved to the config file you chose (so you can load them up again that way later). Once you’ve adjusted the parameters, the plotting canvas window will pop up, and it will ask you to if and how you want to start the simulation. If you are just figuring out how to use the model, you may want to let some of the “realtime” text or plot updates occur so that you will be able to see what’s going on in the simulation. These can slow the execution time by quite a lot, however, so you may wish to eventually run it without any realtime output. Once it’s finished, you get the option of saving some output stats files as well as the plot. You can save any, all, or none of these: make sure to select all of the ones you want.

### Notes

* This software is released under the [GPL license](http://www.gnu.org/copyleft/gpl.html).
* Current release number is v0.6, which is a fully functional beta.
* Please cite as: **“Isaac I. Ullah, 2023. AgModel, version 0.6. http://dx.doi.org/10.5281/zenodo.17551”** if you publish anything related to this software.

### Changelog:

Proposed Changes for next version \* Introduce SeedToYieldRatio, and use it to replace the variables for initial and maximum Kernel yields per patch to be initial and maximum SeedToYieldRatio \* Tie the SelectionCoefficient to also increase the SeedToYieldRatio over time. \* Add a WildSeedingRate and CultivationSeedingRate (number of kernels/ha) and remove CerealCultivationDensity \* Allow for surplus and storage \* Allow birth rate and death rate to be positively affected by surplus \* Increase diversity of alternative food resources (more than just one prey or plant resource)

v0.5 to v0.6 \* Removed HunterGathererRatio (allows this to be an emergent property) \* Added Gaussian filter to diet breadth logic \* Added randomization of prey encounter number \* Added Gaussian filter to coevolutionary selection and diffusion rates

v0.4 to v0.5 \* Updated variable names for readability

v0.3 to v0.4 \* upgraded to Python 3 \* changes to GUI and interaction \* selection is now balanced against diffusion in utilized patches. \* diffusion is now density dependent, so that it reduces as the percentage of domestic phenotype in the ecosystem increases. \* added docstrings to classes and functions (no functional changes)

### Publications and Presentations

Ullah, I. (n.d.) “An Agent-Based Modeling approach to the forager-farmer transition.” Article submitted to American Antiquity (May, 2023).

Ullah, I. (2023) “AgModel: An Agent-Based Model of the forager-farmer transition.” Symposium: “Simulations for the past, simulations for the future.” 2023 Computer Applications in Archaeology Conference. April 3 – 6, Amsterdam, NL.

Barton, L., and Ullah, I. (2016) “Computer simulation and the origins of agriculture in East Asia,” Paper presented at the 7th Worldwide Conference of the Society for East Asian Archaeology, Boston, MA, June, 2016.

## Model Variables

|  |  |  |
| --- | --- | --- |
| **Variable** | \*\*Default Value | \*\* | **Description** |
| People | 50 | Initial number of people in the forager band |
| MaximumPeople | 3000 | Maximum human population |
| HumanBirthRate | 0.032 | Human birth rate (per annum) |
| HumanDeathRate | 0.03 | Human death rate (per annum) |
| HumanBirthDeathFilter | 0.005 | Width of the Gaussian randomizing filter for human birth and death rates |
| StarvationThreshold | 0.8 | Human starvation threshold (percentage of per annum kcal need) |
| HumanKcal | 912500 | Yearly per capita kcal need |
| ForagingHours | 4380 | Per capita hours of work available per annum |
| ForagingUncertainty | 0.1 | Width of the Gaussian randomizing filter for foraging returns |
| Prey | 200 | Initial number of prey |
| MaxPrey | 500 | Maximum number of prey |
| PreyMigrants | 0 | Number of prey that migrate into the territory per annum |
| PreyBirthRate | 0.06 | Prey birth rate |
| PreyDeathRate | 0.04 | Intrinsic prey death rate (without human hunting) |
| PreyBirthDeathFilter | 0.005 | Width of the Gaussian randomizing filter for prey birth and death rates |
| PreyReturns | 200000 | Kcal return from an average prey |
| PreySearchCost | 72.0 | Hours search time to find a prey |
| PreyDensity | 1000 | Density for which the search time is known |
| MaxPreyEncountered | 4 | Maximum number of prey per encounter |
| MinPreyEncountered | 1 | Minimum number of prey per encounter |
| PreyHandlingCost | 16 | Hours of handling time per encounter |
| Cereal | 100 | Number of cereal patches (assume 1ha/patch) |
| WildCerealReturns | 0.05 | Return rate (kcal) per wild-type cereal seed |
| DomesticatedCereal  Returns | 0.1 | Return rate (kcal) per domestic-type cereal seed |
| WildToDomesticated  Proportion | 0.98 | Initial proportion of wild to domestic types |
| CerealSelectionRate | 0.03 | Selection coefficient (percentage per annum) |
| CerealDiffusionRate | 0.02 | Diffusion coefficient (percentage per annum) |
| SelectionDiffusionFilter | 0.001 | Width of the Gaussian filter applied to selection and diffusion rate |
| CerealSearchCosts | 1.0 | Hours of search time to find a cereal patch |
| CerealDensity | 10000000 | Initial (and minimum) number of individuals per patch |
| MaxCerealDensity | 100000000 | Maximum number of individuals per patch, after human influence |
| CerealCultivation  Density | 1000000 | Rate at which cultivation increases the number of individuals in a patch per annum |
| WildCerealHandling  Cost | 0.0001 | Handling time (hours) per wild cereal seed |
| DomesticatedCereal  HandlingCost | 0.00001 | Handling time (hours) per domestic cereal seed |
| Years | 3000 | The number of years for which to run the simulation |