How to Build an Ecosystem

I use genetic algorithms to breed an ecosystem full of plants, herbivores, and carnivores living on energy from a nutrient layer. In nature there are many levels of carnivores with competition among various herbivores. I implemented only one species at each level to simplify my work. I use three genetic operators in the simulation: crossover, mutation, and selection. Survival is my sole selection criterion. If an organism has enough energy it survives. If it has enough more it can reproduce. It finds a mate with whom it exchanges genetic material. Copies of two parent chromosomes are combined using the crossover operator with a small amount of mutation. The upper bits from one parent and the lower bits from the other parent form the new chromosome. This process is repeated for all chromosomes in each parent. Each organism has an energy and a marker chromosome. Every plant also has a chromosome to hold its characteristic traits. Herbivores have a chromosome to store their traits, while carnivores have a chromosome to hold theirs. Crossover breeds for the most fit organisms; mutation breeds for change.

The environment lives in a 3500x3500 pixel area which I call The Veldt. The nutrient layer is a grid which forms the framework of the ecosystem. Each organism: plant, herbivore, or carnivore lives at a point on that grid. The plants feed from the nutrient layer at their location. Herbivores and carnivores have other feeding behaviors. The plants absorb the nutrients then reproduce, if given enough energy. The herbivores move around the veldt searching for plants to eat. The carnivores follow herbivores; if they are hungry they feast. Scroll around the environment using the arrow keys, or the scroll bars. PgUp and PgDn, or the mouse wheel can also control up and down movements.

I wrote this simulator to examine how energy flows through the system. The ecosystem starts with the energy contained in each organism and stored in the nutrient layer. Each organism expends energy during a cycle. The rate of metabolism is determined by a gene on the organism's energy chromosome EN. Herbivores and carnivores do not use all the available energy when they eat. The amount they absorb is determined by another gene on the EN chromosome. Any left over plant or herbivore energy flows back into the nutrient layer. Reproduction is similarly messy. Each parent provides one half of the offspring's natal energy. They also return a certain amount of energy back to the nutrient layer. At senescence an organism dies and its energy flows to the nutrient layer.

During each time cycle every plant eats, reproduces (if energetic enough), and dies if its energy drops below zero. Each herbivore and carnivore moves around, eats what it can, reproduces if possible, and dies if it must. The herbivores and carnivores move directed by their direction chromosome DR. The DR chromosome has eight genes which represent possible directions, plus another for speed. The genes on DR are read and a normalized direction vector is calculated.

Plants have spines and poison, determined by genes on the plant chromosome PL. Herbivores can eat some level of poison and shorter spines, both genes are stored on the herbivore chromosome HB. Reproduction age, satiety, and senescence are also genetically determined. Use the two-click spot sample mechanism to track certain genes and see histograms of some traits. Approximately one half of the genes are currently expressed. All of the energy and direction genes are employed, but only some of the plant, herbivore, and carnivore genes are accessed. Those genes describe behavior traits. At the present I am not using the sensor chromosome SN. All five senses could improve the hunting skills of a carnivore, or provide some level of safety to a herbivore.

Energy flow

The nutrient layer begins with a uniform amount of energy in each cell of the grid. Each plant, herbivore, and carnivore is created with a fixed amount of energy. Each organism gains energy by absorbing energy. All organisms lose energy to the environment because of their metabolism. Herbivores and carnivores also lose energy when they move. If an organism reaches the age of senescence it dies, giving its remaining energy back to the environment.

A plant absorbs energy from the nutrient layer. It examines its satiety gene (EN::G3) to determine whether it is hungry or not. If it is hungry then it looks at its absorb gene (EN::G4) to calculate the absorption factor. Each plant gains absorb * E(i,j)->energy. In other words, it absorbs a certain percentage of the nutrients available at E(i,j).

Herbivores express seven genes when they feed. In order: speed (DR::G0), satiety (EN::G3), eat (HB::G1), resist (HB::G0), spine (PL::G1), strength (PL::G2), and absorb (EN::G4). On a side note: examine the labels inside the parentheses. The first two characters (EN, DR, HB, and PL) designate which chromosome is used. The final two characters: G0, G1, etc. designate which gene on the chromosome is used. It is a shorthand I use to help me map genes.

Each organism has an energy chromosome, EN. It determines the energy use per cycle, the energy absorption rate, the satiety level, the breeding age, and the age of senescence. After many tests I have found it is best to not allow eternal life. It is also not good to overeat. Thus the two genes: senescence and satiety. If a herbivore is hungry it eats until full. If it is already full it does not eat any more.

Animals are messy during feeding. Not all of their food gets eaten, some of it falls onto the ground, and then back to the nutrient layer. The herbivore absorbs a genetically determined percentage of the plant's energy (EN::G4). The rest returns to the nutrient layer while the plant is removed from the simulation. Carnivores absorb energy by eating herbivores. EN::G4 determines how much energy the carnivore gains. The herbivore's remaining energy returns to the environment at its location.

I expressed a few more genes so plants now have both spines and poison. Reciprocally, each herbivore has two genes which allow them to withstand a certain amount of poison, or a certain length of spines. I find it interesting to see how plants breed toward more spines or higher poison levels, while the herbivores breed for higher resistance to poison and the ability to eat longer spines. Use the two-click spot sampling mechanism; then click the 'Spine War' button to monitor the progress.

When an organism reproduces it must have enough energy, and be old enough. Two plants breed to create an offspring with 1/8 of the energy from both parents. Each parent also loses 1/16 of their energy back to the environment. That means each parent loses 3/16 of their energy due to reproduction. So the energy for each offspring is taken from both parents while some energy falls back into the environment. No energy is created and hopefully none is lost. It would be good to monitor the energy total from nutrient layer up through the carnivore layer. Ideally, the total will be a fixed amount.

It is easy to track the history of energy use in the nutrient layer. A large growth of plants is followed by a dark, depleted nutrient layer. Watch the herbivores eat a dense area of plants. Notice the nutrient layer growing brighter as the plants disappear. When the carnivores start feasting on those herbivores the nutrient layer grows even brighter. All of the waste energy from eating, and reproduction, is fed directly to the nutrient layer. This is recorded in the nutrient layer until the next wave of plant growth passes through.

The nutrient layer, and each organism, starts with a set amount of energy. All of the organisms metabolize; each loses energy and grows one cycle older. The metabolic energy is returned evenly across the nutrient layer. Balancing the energy flow makes the system more stable for longer simulation runs.

Simulation display

The application begins with a window 1000x1000 pixels in size. It is set to the upper right of the 3500x3500 pixel simulation area. Scroll across the area to find points of interest. Expand the window to fit your screen size. A text window also opens. It displays the cycle number; plant, herbivore, and carnivore populations; as well as the highest energy level of the plants, herbivores, and carnivores. I use this display to tune the ecosystem. I also graph the information it stores in dt.dat with Gnuplot so I can see population or energy cycles.

The simulation uses the following colors. Each red pixel represents a single carnivore, each blue pixel is a herbivore, and each green pixel represents a plant. The nutrient layer is displayed a little differently. Each pixel represents the energy level at that location. The darker color (a dusky brown shade) is shown where the energy has been depleted. The lightest colors represent areas of high energy. Notice how rapidly plants reproduce where the color is bright, as compared to areas where the nutrient layer is depleted, designated by darker colors. The nutrient layer maps a 'history' of energy use.

Over time the nutrient layer develops a complex, mosaic structure. This variation is mirrored by the plant, herbivore, and carnivore populations. Noticing this caused me to save mature nutrient layer data for the future. Hit the 'n' key at any time to store a new copy of nut.dat, the entire nutrient layer energy level map.

The application searches for two files when the simulation begins. One of them is the nutrient layer data, nut.dat. The other is the organism sample file, sample9GM.dat. This is a sample taken every 1000 cycles of the surviving organisms. The organisms become better adapted to the system as the simulation matures. Storing both the nutrient layer and the organism samples, lets me save the genetic information and the nutrient mosaic. If both of the files are present the ecosystem begins with a varied mosaic of nutrients, and time tested chromosomes. Because these are only sample files, new, random organisms are added to achieve the initial population levels. Thus mature genetics is represented by the sampled organisms, while random genetics is provided to the remaining population. The simulation begins with a uniform nutrient layer if the nutrient data is not available.

Genes from bits to behavior

Organisms derive traits from genes mapped on chromosomes. All genes are bit based. Trait parameters are calculated from those binary genes. Each organism has multiple chromosomes, each of them holds multiple genes. Every chromosome is 32 bits long. I employ a few simple techniques to convert genes into trait parameters. I will use the direction gene DR as an example. DR has 9 genes mapped onto it. 3 bits for each cardinal direction (N,NE,...,W,NW) and 3 more for a speed gene. I sum the 8 raw direction values, then divide each value by that sum for a direction probability. One herbivore's direction vector gives me: E 0.125, SW 0.250, S 0.125, SE 0.500. This herbivore is able to move in four directions: E, SW, S, and SE. It moves southeast one half of the time, southwest one quarter of the time, and it moves east or south one eighth of the time respectively. This herbivore cannot move in any of the other four directions.

I have written a number of ecology simulations. In those earlier simulations I expressed genes as whole numbers. Whole numbers work for a few cases but not for all of them. My first modification was adding an offset in the code. The gene's raw value is added to the offset to calculate the result. This method worked for more cases. However, the real world does not use whole numbers; it uses a continuous range of values. I needed fractional numbers to track energy more accurately. I began by dividing the gene's bit value by the maximum possible value.

This gave me a range from 0.0 to 1.0. This range must be limited. If your absorption gene tells you to use 0% of the available energy you will quickly die. My first response was to add 1 to the raw gene value. Then I added 1 to the maximum possible value. Instead of 0/7, I have 1/8 as the minimum value, and 8/8 as the maximum value. If this gene tells you to use all of the available energy the simulation will become unbalanced. I needed to limit the upper end of the range too.

My next modification was to increase the maximum value. I do not want any plant to be able to absorb more than half of what is available. I let the absorption factor range from 0.067 to 0.533 {1/15 to 8/15}. Fractions simplify things. Subtracting a fixed amount from a parameter can create negative values. But, multiplying a small number by a fraction always gives you something. It may be small, but it is not zero, or a negative number.

Plants contain three chromosomes: energy characteristics are on EN, plant traits on PL, and the genetic marker GM. Herbivores have five chromosomes: energy EN, herbivore traits HB, direction DR, sensor SN, plus genetic marker GM. Carnivores have a similar set of five chromosomes: energy EN, carnivore traits CN, direction DR, sensor SN, and the genetic marker GM. The GM chromosome is special; it is passed directly from the mother to the offspring without using the crossover function. This allows me to track new chromosomes breeding in a population.

PAoffset, HAoffset, and CAoffset are user input values of plant, herbivore, and carnivore age of sexual maturity. Here is the pseudo code to calculate a few traits from the EN energy chromosome:

maturity = EN::G2 + HAoffset

satiety = EN::G3 + CSoffset

absorb = (EN::G4 + 1.0) / 15.0 1/15 to 8/15, or 0.07 to 0.53

Here is pseudo code to express the absorb energy trait:

Energy added to plant at (i,j) = absorb * nutrient energy available in cell E(i,j).

Or energy available to herbivore = absorb * energy held in plant.

User interface   Press 'h' to get this help file

'd' hides/shows the population data shown in the upper right of the display

'l' saves a sample to disk to sample9GM.dat   choose 1000 each of P, H, C from the population

'n' logs the nutrient layer to nut.dat   save energy level at each 3500x3500 E(i,j)

'a' triggers the adjust age dialog box to modify maturity gene offset levels

'm' triggers the adjust metabolic rate dialog box to manage the offset to the energy gene

'r' triggers the adjust reproduction energy requirement dialog to change the floating point factor

's' calls the senescent age dialog box to modify an offset to the senescence gene

't' to adjust the organisms' satiety levels by an offset to the satiety gene

'q' quit the application

I had forgotten these keys only work while the help list is NOT displayed. Notice also that the word PAUSED is displayed below the population data in the upper right when: a, m, r, s, or t dialog boxes are triggered. You cannot adjust these properties if the simulation is running. If you did the program would crash. Trust me, I know :)

Two-click sample mechanism

Scroll to an 'interesting' population feature; an area with many plants and few herbivores, or an area of many carnivores in a herd of herbivores. Once your feature is centered click once to the upper left and again to the lower right. A dialog box will appear with a number of choices to display histograms of genetics, or population counts, to genetic markers, and energy levels. Follow genetic changes using Compare or Spine War. They both compare genes in the sampled populations. Once you are done with the sample discard it by hitting the 'I Am Done with This Sample' button. Notice, the simulation does not stop during this sampling technique. Many more features will be added here.

Gnuplot from dt.dat

Gnuplot is an application which graphs data. Freeware versions are available for the Linux, Macintosh, and Windows operating systems. I use the dt.dat file to store time stamped population data. The following scripts plot three of the columns of dt.dat as lines on a graph. Here are the steps used under Linux, from a terminal window.

Type gnuplot to start the application. Then type the following (or just cut and paste it):

```
plot 'dt.dat' using 1:4 title 'Carnivore' with line, 'dt.dat' using 1:3 title 'Herbivore' with line, 'dt.dat' using 1:2 title 'Plant' with line
```

or the same script with a different ordering:

```
plot 'dt.dat' using 1:4 title 'Carnivore' with line, 'dt.dat' using 1:2 title 'Plant' with line, 'dt.dat' using 1:3 title 'Herbivore' with line
```

I will parse the last line for you: plot calls the 2D graphing system of Gnuplot. It reads dt.dat as its data source. Using 1:4 tells the 2D system to use column 1 as the X coordinate, and column 4 as the Y coordinate. Title names what you want on the graph's legend; in this case 'Carnivore'. With line draws a line from one data pair to the next. This pattern is repeated for the herbivores and plants. I use the second ordering now because it displays plants as green, herbivores as blue, and carnivores magenta, colors better fitting their respective natures.

Plot displays a linear graph. If you want log, or log, log scales try these commands:

set logscale y leaves the X coordinate alone while it maps the Y coordinate logarithmically.

Type set logscale for a log, log chart where both X and Y are displayed logarithmically. Depending on your data set characteristics, each setting can be useful.

To return to a linear plot type unset logscale.

The Chromosome Map

EN::{ senescence, maturity, satiety, absorb, energy }. Senescence limits organisms to a finite life time, maturity allows time to gain energy and it limits the birth rate, satiety prevents an organism from eating more than it needs, absorb is the amount of energy an organism receives from its food source, and energy is the metabolic rate.

PL::{ strength, spine, pollen } Strength determines the amount of poison, spine controls the length of spines, and pollen determines the pollination rate.

HB::{ camo, eat, resist } Camo controls the efficiency of camouflage, eat allows consumption of some level of plant poison, resist limits the length of spines you can eat.

DR::{ speed, SE, S, SW, W, NW, N, NE, E } Speed limits how far an animal can travel in one time cycle, the direction genes determine the direction probabilities.

SN::{ sight, hearing, smell, taste, touch } Six bits each for all five senses. Still a work in progress.

CN::{ attack, range, sleep, camo, pursuit } I have not implemented any of these behaviors yet.

This gene map is subject to change.

Origin stories

The Veldt by Ray Bradbury
https://www.libraryofshortstories.com/onlinereader/the-veldt
Microcosmic God by Theodore Sturgeon
https://www.scribd.com/doc/44330769/Theodore-Sturgeon-Microcosmic-God