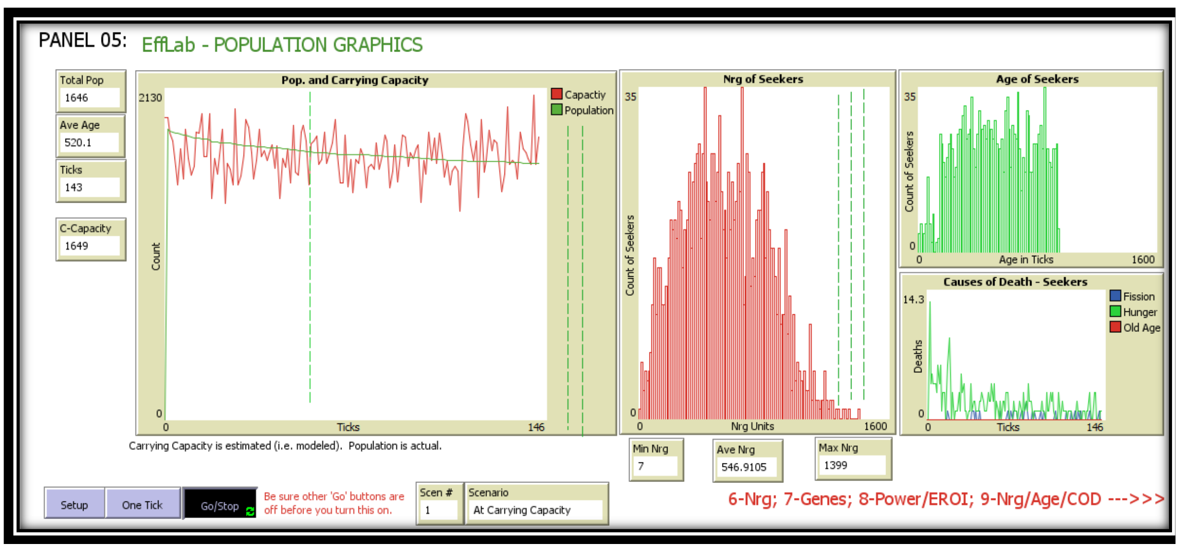
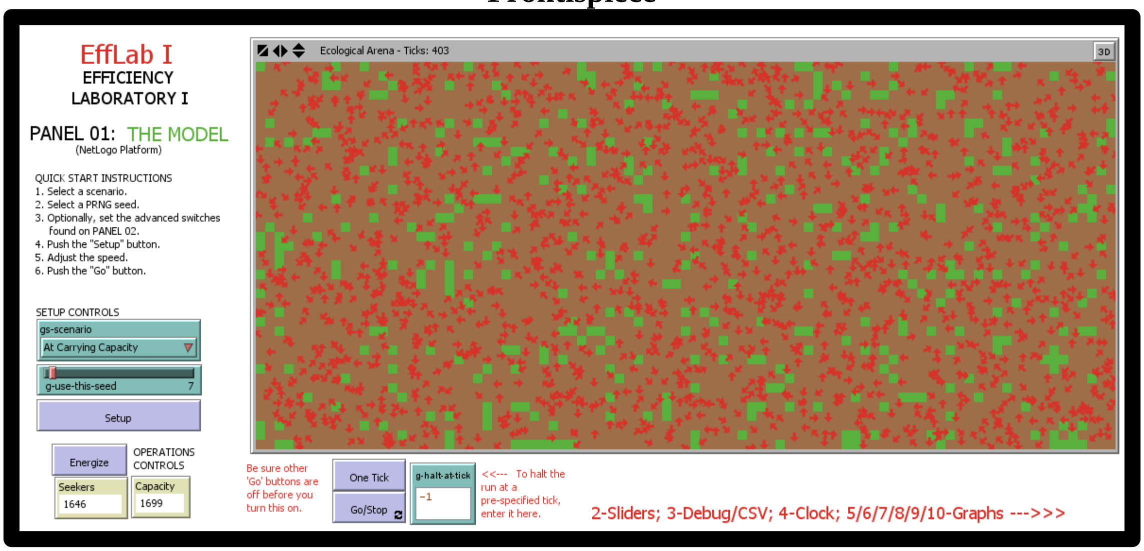
HIGH-LEVEL DESIGN
and
USER DOCUMENTATION
For EffLab
(Efficiency LABORATORY)



Author: Garvin H Boyle
Date: 07 September 2019
orrery-software.webs.com

HIGH-LEVEL DESIGN
and
USER DOCUMENTATION
For EffLab
(EFFICIENCY LABORATORY)

Frontispiece



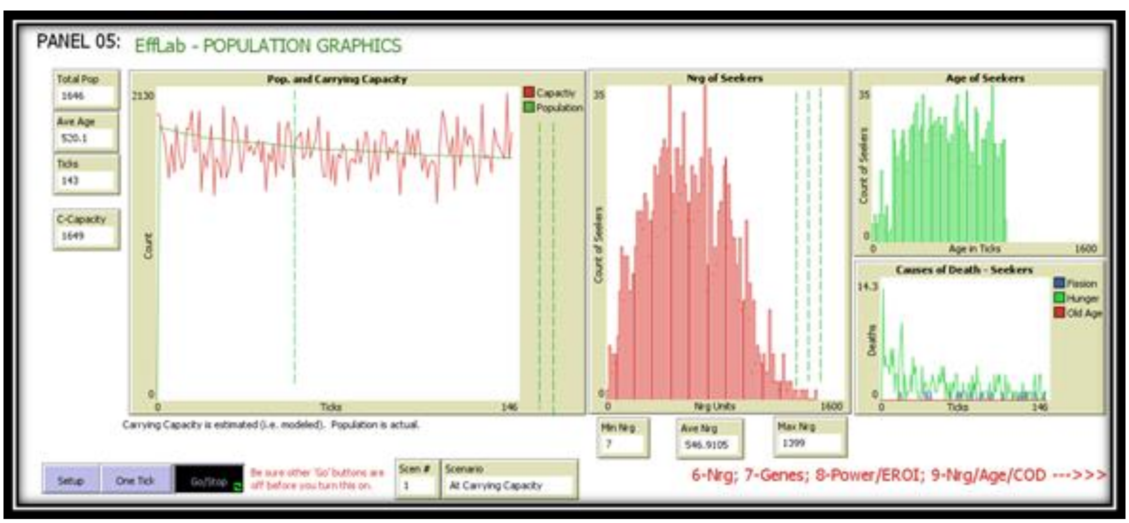


Table of Contents

1 - References

A. G.H. Boyle (2017) EffLab_V5.06.nlogo

Documents about Efficiency Ratios

B. 151224 NTF ICBT and PowEff R8.PDF
C. 170514 NTF On Efficiencies R3.PDF
D. 170518 NTF On Efficiency and Growth R3.PDF
E. 170430 NTF On Efficiency and Energy Grade R6.PDF (***Unfinished***)
F. Boyle_ISBPE_2017 Modeling EROI R9.PPTX
G. 190207 NTF Goldilocks Hypothesis R1.PDF
H. 190506 NTF Telecon - On Conservation in EffLab R1.PDF

Documents about Evolution in ABMs

I. M. Palmiter (1989), Simulated Evolution, Application. Downloadable from http://lifesciassoc.home.pipeline.com/instruct/evolution/.
J. A.K. Dewdney (1989) “Computer Recreations”, “Simulated Evolution – wherein bugs learn to hunt bacteria”, Scientific American, 260(5), 138-141.
K. Boyle, Garvin (2017, April 13). “00 PSoup V1.22 – Primordial Soup” (Version 1.0.0). CoMSES Computational Model Library. Retrieved from: https://www.comses.net/codebases/5594/releases/1.0.0/
L. Boyle, Garvin (2019, October 05). “00b SimEvo_V5.08 NetLogo” (Version 1.0.0). CoMSES Computational Model Library. Retrieved from: https://www.comses.net/codebases/b196ebb8-bae0-47dd-9e49-f2b3c766681e/releases/1.0.0/
M. A.J. Lotka (1922b) 'Natural Selection as a Physical Principle', Proceedings of the National Academy of Sciences of the United States of America, Vol. 8, No. 6 (Jun. 15, 1922), pp. 151-154.

Documents about Entropy in ABMs

N. A. Drăgulescu and V.M. Yakovenko (2000) “Statistical Mechanics of Money”, Eur. Phys. J. B 17, 723-729.
O. G.H. Boyle (2015) ‘Capital Exchange Models – With Investigations of Entropic Index.’ (Version 3). CoMSES Computational Model Library. Retrieved from: https://www.openabm.org/model/3860/version/3
P. 190927 NTF Entropy in a Histogram R4.PDF
Q. 190304 SOP Data For Model I Graphs R1.PDF
R. 180730 NTF About Pup Over Pdown R4.PDF

Documents about the Maximum Power Principle in ABMs

S. A.J. Lotka (1922a) 'Contribution to the Energetics of Evolution', Proceedings of the National Academy of Sciences of the United States of America, Vol. 8, No. 6 (Jun. 15, 1922), pp. 147-151.
T. H.T. Odum and R.C. Pinkerton (1955) 'Time's speed regulator: The optimum efficiency for maximum output in physical and biological systems ', Am. Sci., 43 pp. 331–343.
U. Boyle, Garvin (2017, April 15). “03 MppLab V1.09 – Maximum Power Principle Laboratory” (Version 1.0.0). CoMSES Computational Model Library. Retrieved from: https://www.comses.net/codebases/5596/releases/1.0.0/
V. G.H. Boyle (2015) ‘150907 MPP and Trophic Levels R1.’ Retrieved from: https://www.youtube.com/watch?v=1NsRF4Bi0Zc
W. Boyle, Garvin (2017, April 15). “04 TpLab V2.08 – Teleological Pruning Laboratory” (Version 1.0.0). CoMSES Computational Model Library. Retrieved from: https://www.comses.net/codebases/5601/releases/1.0.0/

Orrery Programming Standards

X. 190503 OrrerySW NetLogo Coding Standards R5.PDF

2 - Background – Conceptual History of EffLab

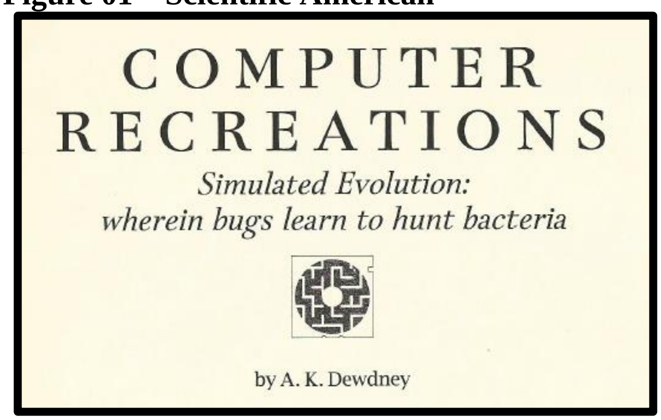
Ref A is the agent based model called **EffLab**, written using NetLogo, which is the topic of this document. The development of this model is part of a personal study of the nature of sustainable economic systems, with a focus on the very basic dynamics of persistent economic processes. At the time of writing of this document some of the above-listed reference documents written by me are very much in need of revision. However, I decided to push through to the end of my logical trajectory to see if all of the stepping stones along the way were identifiable. Having a glimmer of hope, I wanted to see if that glimmer was a will-o-the-wisp, or a real lantern in the distance at the end of a traversable path. A number of the following references are also mentioned in the 'info' tab of the **EffLab** application. **EffLab** may appear, on first glance, to be a simple toy, but I believe that it demonstrates several important dynamics that are both profoundly important, and also obvious but much overlooked.

Refs B through H are a series of diary notes (i.e. notes to file, or NTFs) in which the ideas around EROI and ETA were developed, along with the concept of a Goldilocks loop. The specific goal for Refs C, D and E was to prepare for a presentation at the ISBPE 2017 conference, as shown in the Ref F slide deck.

The Ref I is a DOS-based application that was written by Dr Michael Palmiter in 1985 as a teaching tool for his biology classes. It was made available for download in 1989 by Life Sciences Associates.

Ref J is an article describing Palmiter's application – 'Simulated Evolution'. That was the basis of my several re-implementations of Palmiter's program, including **PSoup** (2002) and **SimEvo** (2019).

Figure 01 – Scientific American



Ref K, **PSoup**, is my most complex re-implementation of Palmiter's ideas. It is written in C++.

Ref L, **SimEvo**, is an extremely simple re-implementation of Palmiter's ideas, used for teaching purposes, and also used as a NetLogo base model for other econodynamic models such as **TpLab** (Ref W) and **Efflab**.

Ref M is a paper by Lotka in which he argues that evolution (and the phenomenon of natural selection) is much more widely applicable than merely in discussion of evolving species, with implications of applications in economics.

Ref N is a paper in which entropy in an ABM is identified. I used this as the basis for my own development of entropy in ABMs.

Ref O is the EiLab (C++) model in which I implemented the eight models described in the Ref N paper, and then expanded on them by building Model I.

Ref P is the culminating note of a series of diary notes in which I determine the best formula to use for calculating entropy in an ABM. Those notes are used in the development of the experimental code in EffLab used to calculate and display the entropic indices for energy distributions and genetic distributions.

Ref Q is a diary note in which I examine some of the data coming out of Model I of EiLab.

Ref R is a diary note in which I examine the connection between entropy in ABMs and Crooks Fluctuation Theorem in thermodynamics.

Refs S through W are a selection of notes about the Maximum Power Principle (MPP).

In 1989 when Dr Michael Palmiter's agent based model called "**Simulated Evolution (SE)**" (Ref I) was described in an issue of "Scientific American", I did the same as many hobby programmers of the day – I wrote my own version of **SE** to better understand how it worked. It was fascinating in its simplicity of design and profundity of insight to be delivered. About twelve years later, in 2002, I reprised that work in the form of **PSoup** (Ref K). In developing **PSoup** I used Palmiter's genes in chromosome #1 (called C1), but also abstracted six of the hidden parameters of Palmiter's **SE** that controlled life functions (metabolic rates, reproductive rates, senescence rates) and placed them into a second chromosome called C2. That proved to be fortuitous for me, as it gave me insight into how to develop many of the other evolutionary and econodynamic models that followed. Variations on the C1 and C2 genes can be found in all of my models that have intergenerational biophysical bases – that is, all but the "capital exchange" models, which do not have a biophysical component.

PSoup demonstrates evolution in a large number of scenarios, but the most basic scenario replicates Palmiter's **SE**, and has been carried forward into the models at Ref L (**SimEvo**), at Ref W (**TpLab**), and at Ref A (**EffLab**). In all of these models, blind organisms stumble about randomly in a field of randomly (re)placed goodies in a fierce competition for survival. The only tool they have that might offer some evolutionary advantage is a heuristic stochastic search pattern that is encoded in their C1 genes. Slight differences in the genome lead to subtle differences of bias in the stochastic process used to effect decisions as they stumble about randomly. Those biases may be adaptive (advantageous) or non-adaptive. In all of these models, with surprising speed, a population of organisms evolve a highly effective search pattern called the "cruiser", and that search pattern also appears as the native "best" search pattern in **EffLab**.

EffLab is a very insightful bench tool, because I can demonstrate a range of phenomena with it, and study their interactions. These include the evolution of search skills, the evolution of metabolic processes, EROI at the scale of the individual seeker, as well as at the scale of the entire system, Jevon's paradox, the maximum power principle, and possibly other related phenomena. The interaction of all of these dynamic phenomena within one model is, I think, rare.

3 - Purpose

The purpose of the Ref A computer application is to explore the nature of a phenomenon referred to as EROI (Energy Returned on Investment).

The purpose of this design document is to present and discuss the design decisions associated with V5.07 of the Ref A computer application – i.e. of the "Efficiency Laboratory". Within this note I will refer to it as **EffLab**.

4 - Discussion of Design Elements

My approach in this document is to write it as a sort of user guide, and a sort of technical record. For the technical part, I will discuss a number of design elements including both technical and conceptual elements. For the user guide part, I will describe each of the panels in the user interface. There is, of course, additional documentation in the **EffLab** model itself, in the info tab and in the code tab. But this document is intended to be the main source of guidance on the use of the model.

4.1 - What is EffLab About?

EffLab was built to support the study of the efficiency of agents in an evolving complex adaptive system. The interim results were presented to an ISBPE conference in 2017 (Ref F). In particular:

- There is a definition of efficiency used in ecology, and an analogous definition widely used in business. In ecological studies it is called EROEI (energy returned on energy invested), or, more briefly, EROI (pronounced E-Roy). In business it is called ROI (dollars returned on dollars invested).
- In addition, there is the more well-known definition of efficiency first invented by Sadi Carnot, and widely used by engineers. It is usually represented by the Greek letter η. In this document it is referred to as ETA.

These two measures of efficiency bear a peculiar relationship to each other, as explored in the diary notes at Refs B through G. Here's a graphic from the Ref B diary note, relating three efficiency measures. I ignore N in this model, because it is so similar to R.

Table 05 – Table of conversions of efficiency formulae.

	η	$\mathcal{R}$	$\mathcal{N}$
η	$\eta = \eta$	$\eta = \frac{R-1}{R}$	$\eta = \frac{N}{N+1}$
$\mathcal{R}$	$R = \frac{1}{1-\eta}$	$R = R$	$R = N+1$
$\mathcal{N}$	$N = \frac{\eta}{1-\eta}$	$N = R-1$	$N = N$

A key assumption, for me, is that both economies and ecosystems are subject to the same stochastic processes when they evolve. These include the effects of conservation of energy (and money, for economies), rising entropy, and maximized power. So this model is purely an ecosystem model, but, as I see it, with strong applicability to economic systems.

PSoup (Ref K) is a purely biophysical ecosystem model, as is its derivative NetLogo version called **SimEvo,** a re-implementation of the 'Basic' scenario of **PSoup**. PSoup is, itself, a re-implementation of a model written by Dr Michael Palmiter in 1985 called '**Simulated Evolution**' (see Refs I and J). **EffLab** is a close copy of **SimEvo**, but with a number of data collection features added.

The key biophysical features of Palmiter's '**Simulated Evolution**', **PSoup**, **SimEvo**, and **EffLab**:

- **Biophysical:** Agents wander through a forest eating fruit. They move about, they eat, they reproduce via fission from time to time, or they die of hunger or old age.
- **Energy Balance:** Agents must collect enough energy to fuel their activities. They collect energy, and they expend it in movement and in reproduction. Energy is conserved, so they cannot generate their own energy – they must find it.
- **Generational:** Agents must reproduce quickly or die. So the rate at which they collect energy is key to their survival. Each generation of agents competes with other agents in its own generation – in its own cohort of agents.
  - **Births:** Agents reproduce via fission, by which they disappear from the model and are replaced by two daughter agents that are almost perfect copies, except for some minor unbiased mutations in their genes. They cannot reproduce until they are old enough and healthy enough. Typically they would need to find 160 patches of fruit within 800 ticks in order to reproduce more quickly than other agents in their cohort.
  - **Deaths:** Agents die if they are hungry or too old. Hunger comes from finding too little fruit to continue metabolic processes. Old age comes from lacking sufficient health to be reproductive. Either way, a low rate of energy collection is the cause.
- **Limited Ability to Sense Fruit:** But, they cannot see the energy-containing fruit until they step into it, so the movement pattern that generates the highest rate of energy returned per tick (highest power of returns) is a key factor in determining which offspring will survive, and which will not. Note that there is a difference between maximum efficiency and maximum effectiveness, as explored in the diary note at Ref G. Those that are maximally effective are most likely to survive into the next generation.
- **Darwinian-style evolution:** Energy comes into the ecosystem at a controlled rate. Every generation can double its population, but there is a de facto limit on how many agents can survive – a carrying capacity. Roughly 50% of each generation will not be able to reproduce due to lack of energy. So-called 'natural selection' will decide who is able to reproduce, and who is not.

The Data Collection Features of **EffLab**:

- Biophysical data is collected and displayed. This includes data about the population size, energy flows, demographic data, genetic data. These are also seen in **PSoup** (Ref K).

- Efficiency-related data is collected and displayed. This is the main purpose of this model. Data about both EROI and ETA are collected. This data provides rather direct insight into several phenomena:
    - Goldilocks curves (Ref G);
    - The Maximum Power Principle (Refs S through W);
    - Jevon's Paradox.
- Entropy-related data is collected and displayed (Refs N through R).

4.2 - Technical Issues

In this section I describe the technical aspects needed to understand the implementation of the concepts. There is a lot more to be said than is found in the "info tab" that could help a modeller understand the way this model is designed, and how it is intended to work.

4.2.1 - Application Development Environment (ADE)

EffLab is implemented in NetLogo version 5.0.5. NetLogo is an interpreted ADE. That is to say, the code is not pre-compiled, but is interpreted line by line as the program runs. This version of NetLogo has now been somewhat surpassed by the developers of the NetLogo platform, which is a fast-moving target. I don't what to get into an endless loop of upgrading my models, so I decided to stick with the application development environment that I first used for the **PMM**, **OamLab**, **MppLab** and **CmLab** models. The '5' in the version number is intended to indicate the NetLogo platform used.

4.2.2 - Models Demonstrating Evolution

Fundamental Evolutionary Dynamic – As described above, **EffLab** is a derivative of a model published many years ago. It was called "**Simulated Evolution**" (**SE**), written by Dr. Michael Palmiter and featured in two Scientific American articles in 1989 and 1990, I think. I wrote my own version of **SE** using Microsoft Basic in the early nineties. Then in 2002, I re-wrote it in C++ and called it **PSoup**. Most of my models written since then borrow heavily from the ideas of Dr Palmiter. For example, the parameters discussed below called DAT, DET, RAT, RET, EPM and EPA directly encode the biological function parameters used in **SE** as described in those Scientific American articles. DAT and DET encode the phenomenon of death in terms of age and energy respectively. RAT and RET encode the phenomenon of reproduction in terms of age and energy. EPM and EPA encode the tendency of agents to expend or store energy, respectively. The general dynamic per tick is then:
(1) move (EPM – energy expended per move);
(2) feed while competing for resources in limited supply (EPA is upper limit on energy storage);
(3) reproduce with variation of capability (RAT and RET determine sufficiency of age and energy); and
(4) die of hunger (DET), old age (DAT), or in reproduction.

The elements of this dynamic all seem to be necessary background constraints to demonstrate real evolution in an Agent Based Model (ABM). ***EffLab demonstrates the evolution of a society of seekers of food.*** It is a multi-generational biophysical model in which the transactional conservation of energy is observed, and the one-use-only nature of free energy is also observed.

4.2.3 - Agent Based Models

EffLab is an agent-based model (ABM). There is one breed of agents called seekers that interact with each other, seeking energy in patches of fruit. They expend energy with every turn (EPM energy units), and they must gather energy quickly enough to grow to mature age (> RAT ticks) size (> RET energy units) and reproduce before they get too old (> DAT Ticks). The energy is limited, however, with insufficient to go around. Those that find more patches of fresh fruit than their contemporaries survive to reproduce via fission, and the others perish of hunger (< DET energy units). The agents are virtually blind and cannot see each other, nor can they see the fruit except for uneaten fruit that coexists in their current cell. The hungry seeker who gets there first gets the food, and nothing is left for others, until that patch is randomly delivered a new batch of fresh fruit. The agents must evolve a heuristic strategy (essentially a stochastic search algorithm) that is more effective than those of their contemporaries.

EffLab is therefore fundamentally an ecological model in which the agents, i.e. seekers, evolve. There is a constant flow of energy through the system with an upper limit on the total amount available per tick, and that upper limit on energy flow determines the carrying capacity of the ecosystem. All seekers reproduce by fission, so once carrying capacity is attained, on average 50% of every generation must die of starvation or old age, and competition for energy begins in earnest. With such fierce competition for a limited resource, any seeker that enjoys the tiniest advantage over other members of its contemporary cohort will have an increased probability of survival. If that advantage can be passed on to its offspring, both will benefit from that advantage, and the number of offspring of that advantaged seeker will grow, at the expense of some other seeker. Such advantage is entirely genetically based, in this model.

But there are two characteristics of seekers in **EffLab** that make it, I think, different from many other evolutionary ecological models:

- First, using the compound gene structure first devised by Dr Michael Palmiter, there is a natural physical correspondence between behaviour of the seeker and its genes. The genes encode a heuristic search pattern in such a way that minor changes in the genes can have exaggerated effects on behaviour. In other words, a minor change in any one gene can convey a substantial advantage or disadvantage to the owner of the gene, and natural selection quickly and effectively selects winners and losers in every generation.
- And third, the sensory capability of a seeker is very limited in range, extending only to the single cell in which it stands. It is able to detect fruit in that cell, and eat it if it is hungry. It is not able to detect fruit in cells right beside it. But it must move in every turn, and expend energy in every turn. Given the fierce competition for finding fruit, every move must be advantageous. On average, the energy returned on energy invested (EROI) must exceed one, or it will perish. Having no useful sensory input, it is virtually blind, and all advantage must come from internal logic – from the heuristic search pattern as encoded in its genes.

4.2.4 - Simulation Vs Demonstration;

A computer model can represent reality in two distinctly different ways:

- **Simulation** – The model is attempting to reproduce, with as much accuracy as possible, a data stream that can be obtained empirically from measurements in the real world. When simulation is sufficiently accurate, then the model may be used to extrapolate beyond the empirical data and predict future events from past events. Weather models, for example, are

of this type. Most economic models are of this type. Usually they are built using macro-level mathematical models involving otherwise intractable differential equations. The immense power of computers to generate an array of look-ahead scenarios lets the modellers investigate 'what-if' possibilities, and possibly assign probabilities of realization of those scenarios using Monte-Carlo or similar techniques. Agent-Based models do not excel at this sort of modelling.
- **Demonstration** – Or the model is demonstrating dynamic behaviour that is implicit in its design, and can be studied as a dynamic system in its own right. While the understandings gained from such studies may be applicable to real-world systems only by analogy, there is no intent to use the model to duplicate empirical data streams obtained from real-world systems. The "Game of Life", or the study of Wolfram's "Finite State Machines" would fall into this category. Whereas few would argue that such models simulate reality, such models are often clearly applicable by analogy, and the things learned in their study has been hugely informative. Agent Based Models excel at this sort of modelling.

EffLab is not a simulation of anything. It is a demonstration of the dynamics that emerge out of the 'natural selection' of logical organisms that reproduce with some small variation. That is all that it is intended to be. ***My design of TpLab is intended to demonstrate EROI in action, in some fashion, and not emulate or simulate it.*** It demonstrates the way EROI works in nature.

4.2.5 - Verification Vs Validation

When a new computer model is constructed there are two ways to test it:
- **Verification** – Is the code free of error – does the software process data as intended?
- **Validation** – Is the model a valid simulation of reality?

The process of "verification" of computer code is the process of finding and eliminating all of the bugs and programming errors. A computer program is considered verified if it has no logical bugs, and runs as intended. This application contains many hundreds of debug calls that are designed to find and eliminate all such bugs. There is, of course, no guarantee, but as far as I know, the economic engine of this application is 100% verified to be running as I intended. All design decisions are documented in this document (at a high level), in the change log (at a medium level), and in the code itself (in detail), and all evidence of bugs of all sorts in the logical engine has been addressed.

The process of "validation" of a model involves comparison of model output with real-world data. **EffLab** is not meant to be validated by comparing numeric output from this model to real-world data. The goal is not to replicate some stream of real-world data accurately. The goal is to generate data that exemplifies the interactions of agents within such biophysical systems, so that the dynamic relationships between them can be better understood.

The kind of validation I seek for **EffLab** is in the validity of the insight it may provide into the workings of a real-world economy. That is, I suppose, a question of interpretation, a matter of degree, and an open question.

4.2.6 - A Model "Run"

EffLab is what one of my students once referred to as a "zero person game". You set some parameters, start it up, and then watch it run until it comes to a conclusion (biophysical collapse?) or until you get tired of watching it (biophysical stationary state?). Each such 'run' is moderated by a pseudo-random number generator (PRNG) and each unique seed for the PRNG will produce a unique run. Each unique run is driven by a unique stream of random numbers, and can be thought of as similar to water flowing in a stream bed. If you place a single leaf into a flowing river, and then track its progress down the river, you might be able to map out the course of the river. But if you do the same with many leaves, one at a time, dropped at different places in the river, and track them all, you may learn about the many bends, channels, islands and back-flows of the river.

A run of the model can be duplicated in every detail if, and only if, all of the setup parameters are the same, including the PRNG seed, and if the run proceeds without disruption of the stream of random numbers delivered by the PRNG. NetLogo tends to call the PRNG implicitly for many actions, so great care is required if a run is stopped. In general, the "One Tick" and "Go" buttons are safe, if used with care. Do not try to press two "Go" buttons at the same time. Do not press "One Tick" if a "Go" button is already on. But, you can stop a run using an already depressed "Go" button, and restart it again, and the run is still repeatable.

4.3 - Design Features

4.3.1 - User Interface Panels

NetLogo does not allow a programmer to temporarily "paint" the computer screen with a type of output placed into a "window". Instead, most graphic output mechanisms (monitors and graphs) must be placed on a single area of logical real estate, and the user can use scroll bars to look at different parts of that real estate. These output mechanisms have static size and location, and only the content varies. I have organized them all, along with the input parameters and controls, into eleven "panels" each of which has a general purpose. Screen grabs of each of the eleven panels are in Annex B of this NTF, and there is a detailed description of each in section 5. The eleven panels are:

- PANEL 01 – Main panel – The Model;
- PANEL 02 – Advanced System Setup Parameters;
- PANEL 03 – Debug and Data Collection Tools;
- PANEL 04 – Clock Face – Averages, 8 Compound Genes;
- PANEL 05 – Population Graphics;
- PANEL 06 – Energy Graphics;
- PANEL 07 – Average Values – Compound Genes, By Type of Component, By Gene #;
- PANEL 08 – Time Lines – Individual EROI and ETA;
- PANEL 09 – Life-Time Efficiency (LTE) data, with breakout by type of death – reproduction or hunger.
- PANEL 10 – Life-Time Efficiency (LTE) data – second panel.
- PANEL 11 – Two Kinds of Entropy – of energy, and of genetic material.

4.3.2 - The Code

The code is readily accessible in any NetLogo model under the "Code" tab. This application has 2,730 lines of code (V5.06) under that tab, which sounds like a lot, but I use a lot of white space, and the code is very heavily commented. The distribution of types of lines of code breaks down like this:

TYPE	No Of Lines	%
Native code	1,413 Lines	45%
Comments	1,101 Lines	35%
Debug calls	77 Lines	3%
White space	534 Lines	17%
Total	3,125 lines	100%

In many instances in the in-code commentary I discuss theory and options and implement one or another of the options discussed. That is roughly equivalent to 20 pages of documentation right in the code. So, I think that some of the very best documentation of this application is in the code itself. I have done it this way to enable other interested modellers to modify or expand upon the model easily.

4.4 - Logical Concepts

The various output panels in the user interface of **EffLab** use some jargon that is unavoidable. These are based on the concepts developed in notes in the reference list, but I expand or refine those concepts as needed. I plan to go back and revise those Refs based on what I am learning in this modeling exercise. Those logical concepts are as follows.

4.4.1 - Movement Within The "Ecological Arena"

A standard feature of every NetLogo agent based model is a spatial representation of the model that I call the arena. The arena is a rectangle tiled by small squares called patches. A patch may be empty (coloured brown), or it may contain fruit (coloured green). In addition, a patch may contain one or more seekers that are represented as little arrows. If we anthropomorphize the little seekers, a decision to move from one patch to the next has three stages of operation:

- **Stage 1** – it consults its set of phenotypic characters ([P] values) to determine how many degrees to the right it should turn in order to maximize its energy returns for the necessary expenditures associated with this step – i.e. to maximize its EROI. The relative [S]trength of each compound gene provides phenotypic character which becomes the probability that a turn of that size will be made.
- **Stage 2** – Turn that many degrees to the right. Such turns are always in multiples of 45 degrees, so the next patch pointed to is always one of the eight patches that immediately surround the seeker (note that the edges of the arena are usually wrapped, so there are usually eight options). The little arrow will turn appropriately as the heading is changed.

Stage 3 – Step forward into the indicated patch, if possible, and attempt to feed. If there is fruit there to be eaten, that is the immediate biophysical payoff.

4.4.2 - Attributes of a Seeker

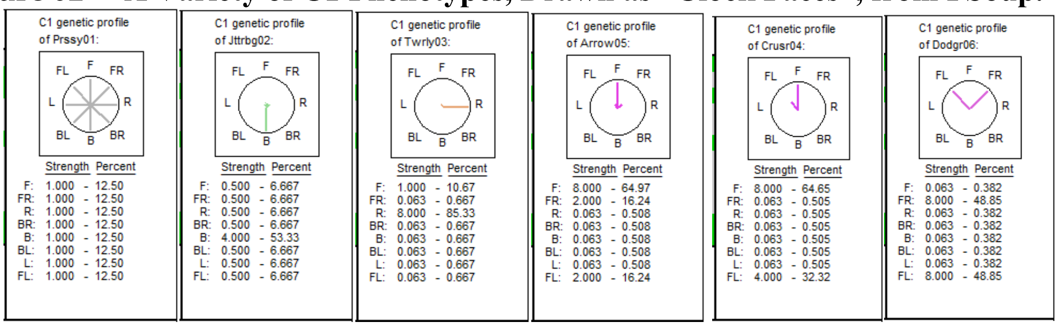
In NetLogo a breed of agents has variables that exist within each agent, called the agent's own attributes. The following code is copied from the code tab of **EffLab**:

```
;;---------------------------------------------------------------------------|
;; Attributes of seekers
seekers-own
[
  ;; BUILT-IN ATTRIBUTES
  ;; who         ;; fixed id number
  ;; breed       ;; to which breed this turtle belongs [seeker]
  ;; heading     ;; 0 <= heading < 360, 0 = north
  ;; xcor        ;; min-pxcor <= xcor < max-pxcor
  ;; ycor        ;; min-pxcor <= xcor < max-pxcor
  ;; size        ;; size relative to a patch, default is 1
  ;; shape       ;; a shape chosen from the shape library
  ;; color       ;; color of this turtle ( 0 <= color < 140 )
  ;; pen-mode    ;; "up" or "down"
  ;; pen-size    ;; in pixels
  ;; hidden?     ;; true or false
  ;; label       ;; label of this turtle
  ;; label-color ;; color of this turtle's label ( 0 <= label-color < 140 )

  ;; USER-DETERMINED ATTRIBUTES
  ;; The chromosome 1 (c1) genes are used to distinguish behaviours.
  c1-bases    ;; c1 - list of 8 [B]ases for genes (S=B^E)
  c1-expos    ;; c1 - list of 8 heading delta [E]xponents
  c1-stren    ;; c1 - list of 8 [S]trengths
  c1-pheno    ;; c1 - list of 8 [P]henotypic characters (Pi=Si/sum(Sj))

  ;; The chromosome 2 (C2) genes are static in this model.
  DAT         ;; Death Age Threshold.
  DET         ;; Death Energy Threshold.
  RAT         ;; Reproductive Age Threshold.
  RET         ;; Reproductive Energy Threshold.
  EPM         ;; Energy Per Move.
  EPA         ;; Maximum Energy Per Agent.

  ;; Other variable characteristics.
  mas-who               ;; serial number of parent agent.
  age                   ;; age of the agent in ticks
  nrg                   ;; nrg in this agent
  cause-of-death        ;; for statistical purposes
  b-is-ready-to-move       ;; 0 = no; 1 = ready to move
  b-is-ready-to-reproduce ;; mature (in age) and healthy (in nrg)
  b-is-ready-to-die        ;; old (in age) or starved (in nrg)

  ;; Variables for calculating individual EROI and ETA.
  ind-nrg-returned      ;; Numerator of EROI - an aggregate = Income of (I/C)
  ind-nrg-invested      ;; Denominator of EROI - an aggregate = Costs of (I/C)
  ind-nrg-benefits      ;; Numerator of ETA - an aggregate = Benefits of (B/I)
  ind-eroi              ;; Gross nrg returned on nrg invested = (I/C)
  ind-eta               ;; Net nrg efficiency = (B/I)
  l-ind-er              ;; A list of delta ERs of length delta T
  l-ind-ei              ;; A list of delta EIs of length delta T
  ind-eroi-tick-counter ;; For tracking time up to delta T.

  ;; Life-time efficiency (lte) variables.
  lte-er                ;; Energy returned (harvested) per lifetime
  lte-ei                ;; Energy invested (spent) per lifetime
  lte-eta               ;; Life-time efficiency - eta
  lte-eroi              ;; Life-time efficiency - eroi
]
```

Descriptions of the purpose of each of these attributes appears elsewhere in this document.

4.4.3 - About Compound Gene Structure

Figure 02 – A Variety of C1 Phenotypes, Drawn as "Clock Faces", from PSoup.

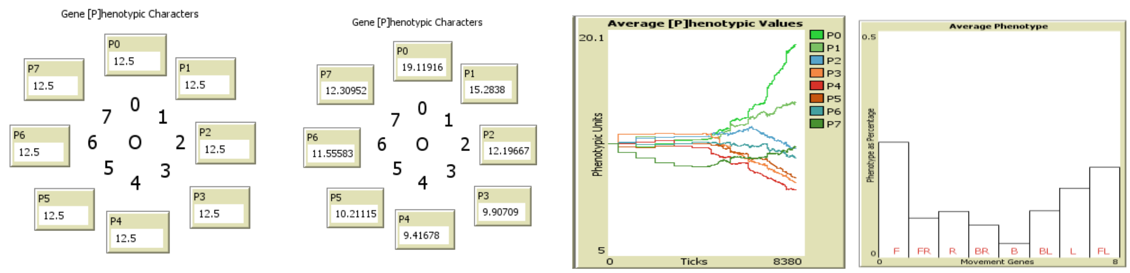


Pristine ***Jitterbug*** ***Twirly*** ***Arrow*** ***Cruiser*** ***Dodger***

The design of the genes in chromosome #1 (C1) was taken directly from the work of Dr Michael Palmiter. I developed that concept further when I expanded upon his **Simulated Evolution** and wrote **PSoup**. In the **PSoup** application I presented these C1 Palmiter genes as a clock face. In Figure 02 you see a selection of such clock faces representing a variety of phenotypes that exhibit as distinctive heuristic search patterns.

I can only approximate this graphic in NetLogo, as seen in Figure 03.

Figure 02 – C1 Phenotypes, Drawn as Clock Faces, From TpLab



Pristine ***Emerging Cruiser*** ***History of Phenotypes*** ***Cruiser***
At 0 Ticks ***<----------------------- At 8,000 Ticks ---------------->*** ***At 33,000 Ticks***

In **EffLab**, all seekers start in pristine form, and the other forms may appear as the populations diversify and evolve. But, as the population becomes very mature, most of the variant forms prove to be less than maximally adaptive, and disappear, leaving a few very effective search patterns (phenotypes) in competition with each other. That is, only those phenotypes which maximize the rate of collection of energy, and the rate of preservation of useful energy (for growth and reproduction) persist. It also happens that these are the phenotypes that cause maximum collection and degradation of energy at a system level. The two contrary effects explain Jevon's paradox, and also demonstrate the two opposing components of the (revised) Maximum Power Principle:

- **PoMDE** – (Principle of the Maximum Degradation of Energy) – Those system-wide energy pathways compete for energy and evolve to garner and degrade free energy at ever higher rates, so long as there is suitable matter to enable the capture and degradation.
- **PoMPE** – (Principle of the Maximum Preservation of Energy) – Those organism-to-organism energy transfer processes (such as digestion and gestation) evolve to garner and preserve free energy at ever higher rates, so long as there are suitable sources of energy that can be metabolised.

4.4.4 - Breeds of Agents, and Explicit and Implied Trophic Relations

In NetLogo, which is an application development environment (or ADE) explicitly intended for agent based models (ABMs), the agents come in flavours called "breeds". In this model I have implemented only one such breed and call them seekers. They seek fruit, and therefore might be considered analogous to herbivores. There are, in fact, representations of two other kinds of organisms. We therefore have three trophic levels:

- The fruit trees get their energy directly from the Sun. These are analogous to primary energy producers at trophic level 1.
- The seekers eat fruit, and are therefore herbivores. These would be at trophic level 2.
- The EPM gene (energy per move) determines how much energy is degraded and discarded as waste heat by every organism. However, some seekers die with unexpended energy, which, in nature, would be consumed by decomposers and detritivores (D&D). This results in the model having a small remnant of non-degraded energy that is released into a special sink, added to track that small amount. There is a switch which activates the D&D and degrades this small remnant, allowing me to calculate the system-wide EROI, including the activities of the D&D. D&D belong to no numbered trophic level.

4.4.5 - Seeker Details and Formulae for EROI and ETA

Details – Each seeker is characterized by a few variables. These include:

The genes in chromosome #1 control the movement of the seekers. These encode the heuristic search pattern of each individual seeker. These are the primary determinants of the rate of collection of energy. That controls the rate of “energy returned” – the ER part of ER/EI.

```
;; The chromosome 1 (c1) genes are used to distinguish behaviours.
c1-bases    ;; c1 - list of 8 [B]ases for genes (S=B^E)
c1-expos    ;; c1 - list of 8 heading delta [E]xponents
c1-stren    ;; c1 - list of 8 [S]trengths
c1-pheno    ;; c1 - list of 8 [P]henotypic characters (Pi=Si/sum(Sj))
```

The genes in chromosome #2 control the metabolic processes of the seekers. These are the primary determinants of the rate of degradation of energy per move. “EPM” controls the “energy degraded per move”. EPA controls the total “energy per agent” that can be stored for future use. This is analogous to controlling growth to adult size, and no larger. These two genes control the rate of “energy invested” – the EI part of ER/EI.

```
;; The chromosome 2 (C2) genes are static in this model.
DAT         ;; Death Age Threshold.
DET         ;; Death Energy Threshold.
RAT         ;; Reproductive Age Threshold.
RET         ;; Reproductive Energy Threshold.
EPM         ;; Energy Per Move.
EPA         ;; Maximum Energy Per Agent.
```

The other four genes of chromosome #2 (i.e. DAT, DET, RAT and RET) do not mutate at birth, and so do not evolve, in this model. Therefore they play no dynamic role in the evolving values of the various calculations of EROI.

Each seeker has a “who number” – that’s a serial number. Each time a seeker experiences fission, it’s who number is retired, and each of the two daughters are given fresh who numbers. The who number of the mother is contained in the variable mas-who (Mother’s who). Age is the age of this seeker in ticks. Nrg is the number of units of energy stored in this seeker. It must between 0 and EPA (maximum energy per agent). Note that EPA can mutate on reproduction, and so evolves as the population and the eco-system matures.

```
;; Other variable characteristics.
mas-who              ;; serial number of parent agent.
age                  ;; age of the agent in ticks
nrg                  ;; nrg in this agent
cause-of-death       ;; for statistical purposes
b-is-ready-to-move      ;; 0 = no; 1 = ready to move
b-is-ready-to-reproduce ;; mature (in age) and healthy (in nrg)
b-is-ready-to-die       ;; old (in age) or starved (in nrg)
```

Within one tick, the system goes through several processes at the sub-tick time level. These are:

- Pre-tick system maintenance.
- Energize energy from the Sun enters the system and is stored in random empty patches.
- Move all marked seekers select a C1 movement gene, turn, and step forward, if possible.
- Feed all seekers eat food in the current patch, if possible.
- Die all marked seekers die, and any remaining energy is stored for consumption by decomposers and detritivores.
- Reproduce all marked seekers reproduce via fission, if possible.
- Post-tick system maintenance.

All of the above agent characteristics (i.e. internal variables) are common to the three models (**SimEvo**, **EffLab**, and **TpLab**). The next set of characteristics are peculiar to **EffLab**. The following variables are peculiar to EffLab, and are used to calculate **EROI** and **ETA** in various ways.

The following characteristics of each agent are used to store data and calculate the EROI values, and the ETA values. I need to translate the lexicon of the Ref B note (about ICBT curve sets), into the notation used in this model. We have:

- I = total income = energy returned or dollars returned = ER = gross returns.
- C = costs = energy invested or dollars invested = EI.
- B = benefits = profits = (I – C) = ER – EI = net returns.

Then the equations for EROI and ETA are:

Hall's EROI $$R = \frac{ER}{EI} = \frac{I}{C}$$	Equ 01
Carnot's ETA $$\eta = \frac{ER - EI}{EI} = \frac{B}{I} = \frac{I - C}{C}$$	Equ 02

Recall that ER and EI are power terms, where the time change is incorporated. In the model, I calculate an "Instantaneous" value, and life-time value, and a system-level value, for both EROI and ETA. Since on average a seeker collects 40 units of energy per patch of fruit, and burns 4 units searching for that patch, then the overall success rate is about one patch every ten ticks, so the "instantaneous" rates are zero most of the time (9 ticks out of 10) for the average seeker. To avoid pointless collection of data, I implement a "close to instantaneous" value. I collect a list of ER and EI values that is ΔT ticks long, add up the ER and EI values, and divide by ΔT, to find a moving average instantaneous rate.

Hall's EROI – Individual – Instantaneous Efficiency $${}^{Instantaneous}_{Individual}R = \frac{\sum(ER)/\Delta T}{\sum(EI)/\Delta T}$$	Equ 03
Carnot's ETA – Individual – Instantaneous Efficiency $${}^{Instantaneous}_{Individual}\eta = \frac{\sum(ER - EI)/\Delta T}{\sum(EI)/\Delta T}$$	Equ 04

```
;; Variables for calculating individual EROI and ETA.
ind-nrg-returned      ;; Numerator of EROI - an aggregate = Income of (I/C)
ind-nrg-invested      ;; Denominator of EROI - an aggregate = Costs of (I/C)
ind-nrg-benefits      ;; Numerator of ETA - an aggregate = Benefits of (B/I)
ind-eroi              ;; Gross nrg returned on nrg invested = (I/C)
ind-eta               ;; Net nrg efficiency = (B/I)
l-ind-er              ;; A list of delta ERs of length delta T
l-ind-ei              ;; A list of delta EIs of length delta T
ind-eroi-tick-counter ;; For tracking time up to delta T.
```

When calculating the life-time values, then ΔT is different for each agent, and is dependent on the age of the seeker when it dies (during fission, or by old age, or by hunger). For those values, I store up the ER and EI sums over the life of the seeker (between 0 and 1500 ticks) and calculate the two ratios of interest when the seeker dies. This data is stored, for each agent, in its lte variables.

Hall's EROI – Individual – Life Time Efficiency (LTE) $${}^{LTE}_{Individual}R = \frac{\sum(ER)/LT}{\sum(EI)/LT}$$	Equ 05
Carnot's ETA – Individual – Life Time Efficiency (LTE) $${}^{LTE}_{Individual}\eta = \frac{\sum(ER - EI)/LT}{\sum(EI)/LT} = \frac{B}{I}$$	Equ 06

Where 'LT' stands for a ΔT equal to the Life-Time of this seeker. The characteristics of the seeker used to manage this data collection are as follows:

```
;; Life-time efficiency (lte) variables.
lte-er               ;; Energy returned (harvested) per lifetime
lte-ei               ;; Energy invested (spent) per lifetime
lte-eta              ;; Life-time efficiency - eta
lte-eroi             ;; Life-time efficiency - eroi
```

This level of detail is never visible in the various monitors and graphs in the display panels. Neither is there currently any means to dump this detailed data to CSV files, other than through the export of line graphs.

I also calculate the instantaneous EROI and ETA at the system level. There is almost always some seeker eating fruit somewhere in every tick, especially if there are sufficient numbers of seekers (>> 10), so I can simply add the ER and EI of this tick, across all seekers, and calculate a value. This data is collected at the system level, so no characteristics are needed within each seeker to hold the sum. This is an exactly 'instantaneous' value, and not a moving average.

Hall's EROI – Instantaneous – System Wide $$ {}_{\substack{Instantaneous \\ System-wide}}R = \frac{\sum(ER)/1}{\sum(EI)/1} $$	Equ 07
Carnot's ETA – Instantaneous – System Wide $$ {}_{\substack{instantaneous \\ System-Wide}}\eta = \frac{\sum(ER - EI)/1}{\sum(EI)/1} $$	Equ 08

5 - Descriptions of the Panels

NetLogo is unable to redraw the screen, so items on the screen must be laid out in a static pattern, and any items that do not fit onto a single screen must be accessed via scroll bars. I have therefore organized all the graphs, monitors, sliders and other objects into eleven panels that are accessible via horizontal scroll bars. Screen shots of the eleven panels are in Annex A, but each is described in detail below. For all programming I follow the coding conventions and variable naming conventions established at Ref X.

I remind you that this model is intended to be a bench tool, so all is not perfectly clear from the graphs. Sometimes I adjust the code, or delete certain graphs, to be sure that other graphs collect enough data. Then I dump the data to CSV files (data files with Comma-Separated Values), load it into MS Excel 2010, and then do further analysis.

5.1 - Panel 01 – The Model

This is the main panel, the panel that you see when you first start up the **EffLab** application. On this panel you will find:

- Instructions for a "Quick Start".
- The "Ecological Arena" in which the fruit trees show as green patches, and the seekers show as arrows.
- Two categories of controls labelled "Setup controls" and "Operations Controls". Setup controls are to be used once at the beginning of a specific run. Once the run has begun, they should not be used, or they will invalidate the run, and possibly cause errors. The operational controls, on the other hand, can be altered at any time during a run. It is best if the run is paused before the control is adjusted, but that is not a necessity. However, the model does not keep track of such changes in control settings, and such runs are not repeatable.

Panel 01 has the following setup controls:

- **gs-scenario** – This is a "chooser" that lets a user select one of two scenarios.
- **g-use-this-seed** – This slider lets the user choose a seed for the pseudo-random number generator (PRNG). A different seed results in a different run. The same seed, if used with care, results in a repeatable run. The PRNG is invoked implicitly by NetLogo for many commands, so, if you stop a run, issue a few commands from the prompt line, and then resume a run, it may not be repeatable. If the only controls used to pause, step or resume are the "Go/Stop" and "One Tick" buttons, then the run is repeatable.
- **Setup** – This button invokes a routine that initializes all seekers and all variables. It should be pressed once only at the beginning of a run. If pressed (during a pause) in the middle of a run, it will reset everything according to the controls, and restart the model at tick 0.

Panel 01 has the following operational controls:

- **Go/Stop** – This button invokes the same sub-routine as the "One Tick" button (described below), but it continues to run that sub-routine over and over again until the "Go/Stop" button is pressed one more time. The "Go/Stop" button turns black when pressed, and turns grey-blue again when un-pressed. Note that there are in fact several copies of the "Go/Stop" button in the various panels. Only one should be pressed at once. Each launches an additional thread of execution, and the results of multiple threads will be unpredictable, and probably not good.

- **[g-halt-at-tick]** – This light-green coloured box is an input box. Any value entered here will be stored in the **g-halt-at-tick** variable. It will be used to terminate execution of a run at a pre-determined tick. Essentially, it un-presses the Go button when that tick arrives. This is useful for debugging, or for the purposes of exploration of events at special ticks.
- **One Tick** – This button invokes a routine that makes the model execute a single tick. That is, the model moves forward in time by one tick. Many sub-routines are called during a tick, in serial order, as follows:
  - ***Do-pre-tick*** – this sub-routine attends to a small number of behind-the-scenes technical matters associated with data collection and display or model performance, that must be addressed prior to the initiation of the other sub-routines.
  - ***Do-energize*** – this sub-routine manages the placement of nrg into the arena. All of the patches are canvassed in random order, and a list of empty patches is established. Then the PRNG is consulted, and each empty patch, with probability [**g-prob-of-deposit**], has an amount [**g-nrg-per-deposit**] deposited into it. This nrg is placed into a variable in the patch called "fruit", and is available to the next hungry seeker that steps into the patch. The actual flow of energy is determined by the number of empty cells, which is determined by the number of seekers and their efficiency at removing fruit.
  - ***Do-move*** – this sub-routine handles the movement of agents within the "Ecological Arena" found in Panel 01. The seekers are addressed in random order. They consult their [P]henotypic values in chromosome C1 to decide in which direction to turn, then turn, and step forward. In the process, they expend nrg in the amount of [**g-c2-epm-parm**]. EPM stands for energy per move, and is based on the work of Dr Michael Palmiter in his model "**Simulated Evolution**". (See Refs I and J.)
  - ***Do-feed*** – In this sub-routine seekers check if they are full of nrg, or still hungry (i.e. their nrg level exceeds or is less than [**g-c2-epa-parm**]). EPA stands for maximum energy per agent. If hungry, they check whether there is fruit available to eat in the patch that they currently occupy. If they are hungry and there is food available, they eat it all, leaving the patch empty until the next execution of the Do-energize routine.
  - ***Do-reproduce*** – In this subroutine each seeker will reproduce via fission, but only if it is old enough and energetic enough. The seeker compares its age to the parameter [**g-c2-rat-parm**]. RAT stands for reproductive age threshold. It also compares its nrg to the parameter [**g-c2-ret-parm**]. RET stands for reproductive energy threshold. Both of these parameters are derived from the work of Dr Michael Palmiter. If old enough and healthy enough, it undergoes fission, producing two daughters each having half of the nrg of the mother, and having age 0. The daughters inherit the compound genes, but with possible mutations. With a probability of [**g-prob-of-mutation**], one of the eight compound genes in chromosome C1 is selected to be mutated, and either the [B]ase value (c1-bases) or the [E]xponent value (c1-expos) is varied slightly.
  - ***Do-die*** – Each seeker checks to see if it should die, due to hunger or old age. If its age is greater than [**g-c21-dat-parm**] it should die of old age. If its nrg level is less than [**g-c2-det-parm**], it should die of hunger. Both of these parameters are derived from the work of Dr Michael Palmiter. If it dies, the remaining nrg in the seeker is sent to a sink and recorded. This may or may not be consumed by decomposers and detritivores (DnD), depending on the value of the switch [**gb-include-dnd**] in Panel 08.

- ***Do-post-tick*** – In this subroutine several activities associated with maintenance of the model are undertaken. This includes updating aggregate variables and calculating EROI, ETA and entropies.

5.2 - Panel 02 – Advanced System Setup Parameters

There are three sets of controls in this panel, and one stand-alone button – the "Reset Defaults" button. None of these sliders need to be adjusted to enjoy insight from the running of the model, but I placed them here initially to tune the model, and have left them here for the amusement of those who have an interest.

Reset Defaults – This button invokes a routine that resets the default values of all of the setup and operational controls to a profile of values that are known to be interesting. When the sliders in this panel and in Panel 01 are changed, and the model is saved to file, the new values are persistent across such a save operation. Many combinations of control settings are of little interest as they are unstable, and possibly dysfunctional. This is a bench tool and not a game. It can be easily broken and made dysfunctional. This button enables the user to return all controls to default values, at need. It is a bit like a factory reset.

Various Advanced Controls – This group of sliders includes a variety of parameters associated with this specific model:

- ***[g-seekers-at-start]*** – used in the setup routine to decide how many seekers to be created. In all scenarios, the number of seekers created will be this number times 8. However, in scenarios 0-2 they will all be placed into a single tribe. In scenario 3, there will be this number of distinct tribes, and 8 seekers per tribe.
- ***[g-sun-nrg-per-tick]*** – (Disabled, then removed) – in the Do-energize routine, this is the maximum amount of nrg made available for deposit into empty patches. Raise this number to raise the carrying capacity. Carrying capacity is now handled differently.
- ***[g-nrg-per-deposit]*** – when nrg is deposited into an empty patch, this is the amount deposited. Compare this with [**g-c2-epm-parm**]. Standard values are EPM=4 and [**g-nrg-per-deposit**]=40. This means a seeker must find a fruit-filled patch, on average, every tenth step, or EROI drops below 1 and the seeker will die of hunger. Change these two parameters together. Raise this number to raise the carrying capacity.
- ***[g-prob-of-deposit]*** – when nrg is deposited into an empty patch, this is the probability that the empty patch will in fact be filled on any given turn. This puts a throttle on the rate at which nrg flows into the system. Raise this number to raise the carrying capacity.
- ***[g-prob-of-mutation]*** – when a seeker reproduces by fission, each daughter has this probability that a random compound gene will mutate. Either the [B]ase value or the [E]xponent value might change upwards or downwards by a small amount. [B]ase values are =2 at setup. [E]xponent values are =0 at setup.
- ***[g-max-for-lte-stats]*** – collecting data for "Life-Time EROI" or "Life-Time ETA" statistics requires a special run. This is a control that sets the number of seekers that must die (and have their statistics collected) before an "LTE" run can terminate.
- ***[g-dt-for-ind-stats]*** – the EROI graphs are extremely volatile, to the point of being almost unreadable. So I apply a moving average to the data. This slider sets the interval over which the moving average is calculated.

C2 Controls – Biological Functions – This group of sliders includes six parameters that are derived directly from the work of Dr Michael Palmiter, the man who invented the schema for the c1 genes. (See Refs I and J.) I parameterized these values when I designed **PSoup** (see Ref K), and have used them since in many evolutionary models. Together they address limits on size of organisms, age of organisms, energy consumption rates, and characteristics of reproduction and death:

- ***[g-c2-dat-parm]*** – Death Age Threshold – maximum age after which a seeker dies of old age;
- ***[g-c2-det-parm]*** – Death Energy Threshold – minimum level of nrg below which a seeker dies of hunger;
- ***[g-c2-rat-parm]*** – Reproductive Age Threshold – minimum age at which a seeker can reproduce;
- ***[g-c2-ret-parm]*** – Reproductive Energy Threshold – minimum energy at which a seeker can reproduce;
- ***[g-c2-epm-parm]*** – Energy Per Move – mandatory expenditure of each seeker per turn;
- ***[g-c2-epa-parm]*** – Energy Per Agent – maximum energy an agent may hold, preventing further feeding when exceeded.

The values set in these sliders are the values at the start of a run. The default values are derived from Dr Palmiter's program. These values are transferred into the C2 chromosome at setup. Two of them can mutate and evolve, if the [**gb-mutate-epm**] and [**gb-mutate-epa**] switches on Panel 06 are activated.

Reset Defaults – A button – This sets all sliders to their default values, like a "factory reset" button.

Active Scenario – A monitor – This monitor displays the value of the string [**gs-scenario**], which is set when the chooser is set on Panel 01.

5.3 - Panel 03 – Debug and Data Collection Tools

Panel 03 has two sub-panels. All of the controls and monitors in this panel were built during the verification process. I decided to leave them as active controls for two reasons:

- **Sub-Panel 03a** – Others may wish to change the model in various ways, and these tools may prove useful when the code is being re-verified by other programmers.
- **Sub-Panel 03b** – Even if you are not a coder, these debug tools may give you a view inside the operations of the model that is not available via the various monitors and displays in the user interface.

5.3.1 - Sub-Panel 03a – Debug Tools

These controls are organized into four groups as follows:

- **Group 1 – "Toggle Debug" button, and associated "Debug" monitor.** The code contains many calls to a routine called "LOG-TO-FILE" which creates a stream of data for a log file. Let's call it the log data stream. This button turns on the collection and presentation of the log data stream. This button MUST be pushed to turn on logging **before** any other debug actions are taken. If not, unpredictable things may happen, and the program may show errors relating to file accessibility or similar things. The monitor indicates whether a log file is open or not. Use this toggle to enable a T-junction by which the data stream is sent both to the log file on your hard drive, but also to the NetLogo standard "Command Centre". The command centre is special window available via a small pair of very small caret triangles in the bottom left corner. Use the expand caret (pointing upwards) to open the command centre window, then use the two-headed arrow at the bottom right to expand it to fill half of the screen. Then the log data stream will be visible in the command centre. See the graphic for Panel 01 or Panel 03 to see the location of these tiny carets (Annex A). This button can produce thousands of pages of output, so be careful.

- **Group 2 – [gs-debug-step-chooser]** – Use this chooser to select which sub-routine is to produce a log data stream. You can set it to "All", or to a specific step.
- **Group 3 – Setup, One Tick and Go buttons** – These are copies of three buttons described previously under Panel 01. They occur in several panels.
- **Group 4 – Single step buttons** – It is possible to step through a tick one sub-routine at a time. Use this, together with the chooser in Group 2, to focus on a single sub-routine at a time.

5.3.2 - Sub-Panel 03b – CSV Data Collection Tools

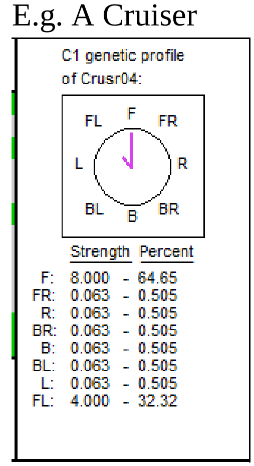
Most of this sub-panel is taken up with generic instructions on how to use the built-in NetLogo exportation tools. There are two groups of additional controls:

- **[db-plot-data]** – this is a Boolean switch that toggles the data plotting function for all graphs on and off. If you want to see the graphs develop and/or you want to dump the data to CSV files later, this must be on. If you want to skip through the transient behaviour of the model, this can be off. It is recommended it be left on, as the best most interesting behaviour is when the model is in its transient phase.
- **Record Selected Plots** – This button is used to send the data contained in the four plots of Panel 07 to CSV files, to be read into MS Excel later for analysis.

5.4 - Panel 04 – Clock Face – Averages, 8 Compound Genes

On this panel there are five key groups of monitors, a standard monitor, and a standard set of operational buttons; seven groups in all. The contents of the five attributes within each compound c1 gene are presented as averaged values. For example, the B0 monitor displays the average across all seekers (all belief affiliations, all tribal affiliations) of the [B]ase value of the compound gene pointing in direction 0. The descriptions of the groups are as follows:

E.g. A Cruiser



- **Group 1** – [B]ase values – Eight averages, one for each of the eight compound genes.
- **Group 2** – [E]xponent values – Eight averages, one for each of the eight compound genes.
- **Group 3** – [S]trength values – Eight averages, one for each of the eight compound genes.
- **Group 4** – [P]henotypic values – Eight averages, one for each of the eight compound genes. These are arranged in the form of a clock face.
- **Group 5** – A monitor – "Ticks" display the current tick.
- **Group 6** – Push buttons – The three standard control buttons as described under Panel 01: Setup, One Tick, and Go.

The example shown in Panel 04 (See Annex A) is a "cruiser". Another example of the "clock face" of a cruiser, taken from the PSoup model (Ref K), is shown here.

5.5 - Panel 05 – EffLab – Population Graphics

This panel is focused on population-related aggregate statistics, and has seven groups of controls, as follows:

- **Group 1** – four monitors at the far left display (1) the total population of all seekers currently alive; (2) the average age of all seekers; (3) the number of ticks that have been executed so far in this particular run; and (4) the currently estimated carrying capacity of the model.
- **Group 2** – to the immediate right of that, there is a set of four graphs:
  - **Pop. and Carrying Capacity –** this is a historical graph of the population and carrying capacity. Since the carrying capacity is estimated, the estimate is rough, as you can see. This tends to be the way it is in a real ecosystem. The estimate seems to be biased to be high, when compared to the actual maximum that exists. Note that the carrying capacity, however, evolves to climb higher as time passes. This is because the EPM C2 gene is evolving, taking on an ever lower value (see Panel 06). This plays a role in Jevon's paradox, and also plays a role in the two-part maximum power principle.
  - **Nrg of Seekers –** this histogram shows the distribution of energy is a "maximal entropy" version of a "Maxwell-Boltzmann" distribution of energy. That is, the distribution of energy resembles the distribution of energy in a hypothesized bottle of liquid or gas of idealized atoms in equilibrium state. Imagine one of the green vertical lines to be the level of energy needed for evaporation of a liquid to occur. Only a few atoms have sufficient energy to escape the surface tension. If these seekers were competing for money instead of energy, this would be the distribution of net worth in the economy (as seen in my model ModEco, available from **OpenABM**).
  - **Age of Seekers –** this histogram shows the current distribution of ages. Those that are over 800 ticks in age could reproduce, if healthy enough, but we see that many are unhealthy. Many (about a third of the population) are destined to die of old age, being consistently unable to reproduce due to lack of health. I explore causes of death in other Panels.
  - **Causes of Death – Seekers** – This is a line graph with historical information about death by fission (controlled by RAT and RET), by hunger (DET), or by old age (DAT).
- **Group 3** – immediately below the histogram "Nrg of Seekers" is a set of three monitors that indicate the minimum, average and maximum nrg of the population of seekers.
- **Group 4** – is a set of two monitors that indicate which scenario is selected.
- **Group 5** – is the standard set of three control buttons – Setup, One Tick, and Go/Stop.

5.6 - Panel 06 – Energy Graphics

The focus of this model is on energy collection and consumption. A steady flow of energy from the Sun drives the entire ecosystem, and the throttles on that flow of energy set the carrying capacity, and cause conflict over access to energy. So, even though the energy data closely parallels the population data, I have established this panel to present the energy view of the history of a run of the model. Any of these graphs can be exported to CSV files at any time. The line graphs contain a history of values, but the histograms only contain the data for this tick of the model. This panel has seven groups of controls:

- **Group 1** – On the far left, there are five monitors:
    - **Fruit** – the total energy currently in the patches of fruit.
    - **Seekers** – the total energy currently in the seekers.
    - **Total** – the sum of the energy in the fruit and seekers, i.e. the total in the ecosystem.
- **Group 2** – Three graphs:
    - **Nrg Levels** – the three lines are: (1) total energy in all patches of fruit (green); (2) total energy in all seekers (red); and (3) total energy in the system (black).
    - **Sun Takeup Rates** – the actual takeup rate per tick of the model (aggregated in Do-energize step). This tells us how quickly the Sun's energy is being absorbed by the ecosystem. Note that this varies, depending on the number of empty patches, but is fairly stable.
    - **Average Nrg Per Seeker** – I note that it starts at zero, and rises. That is an artifact of the setup routine. All seekers start with a reasonably healthy level of nrg.
- **Group 3 – EPM statistics** – these are only interesting if mutation of the EPM C2 gene is activated:
    - **Ave EPM** – this is a line graph of the average value of the C2 EPM gene, across the entire population of seekers. You will note that there is evolutionary pressure on the EPM (energy per move per seeker) to decline in value. This means that the seekers evolve to be more efficient foragers, consuming less energy per tick as they wander the forest seeking fruit. Because they evolve to become more efficient, the carrying capacity of the system increases, and the number of seekers increases, and the rate of capture of energy from the Sun also increases. This is the root of Jevon's paradox. It is also a key point in the 2-part maximum power principle (MPP).
    - **Distribution of EPM** – this is a histogram of the values of the C2 EPM gene across the current population of seekers. I note that it is like a Gaussian distribution, having the appearance of the results of a sum of many random walks, but it is biased on the left. Those seekers that have a lower EPM value are more numerous than those with a higher EPM value. A low EPM value confers on the seeker a greater efficiency, and a greater chance of gathering enough energy to become reproductive.
    - **Three monitors** – Minimum EPM, Ave EPM and Max EPM. These numbers are associated with the above two graphs.
    - **One [gb-mutate-epm] switch** – this switch toggles the ability of seekers to experience a mutation of the C2 EPM gene during reproduction. When turned off, the EPM gene does not evolve, and the demonstration of Jevon's paradox and of the Maximum Power Principle does not happen. The default value is on.
- **Group 4 – EPA statistics** – these are only interesting if mutation of the EPA C2 gene is activated:
    - **Ave EPA** – this is a line graph of the average value of the C2 EPA gene, across the entire population of seekers. You will note that there is only very slight evolutionary pressure on the EPA (maximum energy per seeker) to rise in value. This means that the seekers evolve to be slightly larger in size, holding more energy (and more biomass).
    - **Distribution of EPA** – this is a histogram of the values of the C2 EPA gene across the current population of seekers. I note that it is like a Gaussian distribution, having the appearance of the results of a sum of many random walks, but I presume that it is ever-

so-slightly biased on the right. Those seekers that have a higher EPA value are more numerous than those with a lower EPA value. A high EPA value confers on the seeker a greater store of energy, and a greater chance of surviving to reproductive age (RAT) as a healthy seeker.

  - **Three monitors** – Minimum EPA, Ave EPA and Max EPA. These numbers are associated with the above two graphs.
  - **One [gb-mutate-epa] switch** – this switch toggles the ability of seekers to experience a mutation of the C2 EPA gene during reproduction. When turned off, the EPA gene does not evolve. I believe that this also plays some minor role in the demonstration of Jevon's paradox and of the Maximum Power Principle. The default value is on.
- **Group 5** – is a set of three standard control buttons as described in Panel 01: Setup, One Tick and Go/Stop.
- **Group 6** – Two monitors are placed at the bottom of the panel to help in the interpretation of the panel: scenario number, and scenario name.

5.7 - Panel 07 – Average Values – C1 Genes – By Component, By Gene

This panel contains four line graphs and two groups of controls, for six groups in total. This is a very informative panel, and the right-most graph called "[P]henotype Values By Gene" is the most informative among the four graphs. Each of the graphs contains eight lines, one for each of the eight C1 genes. These four graphs are those that are automatically exported to CSV files using the CSV tools of Panel 02. The seven groups of controls are:

- **Graph 1** – [B]ase Values By Gene # – These numbers all start with a value of 2, and are altered with probability [**g-prob-of-mutation**] during reproduction. The value plotted is the average across all seekers.
- **Graph 2** – [E]xponent Values By Gene # – These numbers all start with a value of 0, and are altered with a probability [**g-prob-of-mutation**] during reproduction. The value plotted is the average across all seekers.
- **Graph 3** – [S]trengths By Gene # – These numbers are calculated as [Si]=[Bi]^[Ei]. The value plotted is the average across all seekers.
- **Graph 4** – [P]henotype Values By Gene # – These numbers are calculated as [Pi]=[Si]/SUM([Sj]). The value plotted is the average across all seekers.
- **Group 5** – is a set of three standard control buttons as described in Panel 01: Setup, One Tick and Go/Stop.
- **Group 6** – four monitors are placed at the bottom of the panel to help in the interpretation of the panel: scenario number, and scenario name, ticks, and PRNG seed.

Compare the graph of phenotype values with the clock face of Panel 04. The clock face is the current set of values, but this line graph is the historic set of clock face values. Evolutionary pressure in directly on the phenotypes and strengths, and only indirectly on the exponents, and very indirectly on the bases. The color coding of the line graphs gives visibility to this. Note that the backwards-turning genes (the reddish lines) are diminished in power to control movement, and the forwards-turning genes (the greenish lines) are elevated in power to control movement.

5.8 - Panel 08 – Time Lines – Power and EROI (Moving Averages)

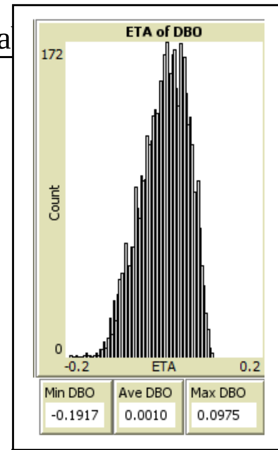
There are four graphs and some standard controls in this panel – six groups in all:

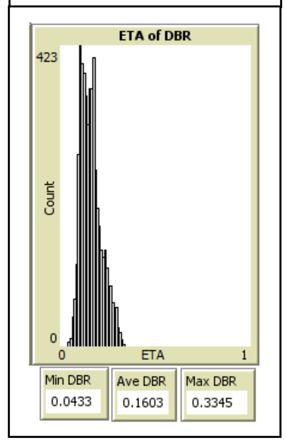
- **Group 1** – EROI – IND (MAX, AVE, MIN) AND SYSTEM-WIDE – This graph shows the instantaneous individual EROI of the population of seekers, and also the system-wide EROI. The data for the individual EROI in this graph is an average over the population of individual EROI numbers calculated as moving averages over the past [**g-dt-for-ind-stats**] number of ticks. The data for the system-wide EROI is calculated on a tick-by-tick basis.
- **Group 2** – ETA – IND (MAX, AVE, MIN) AND SYSTEM-WIDE – This graph shows the instantaneous individual ETA of the population of seekers, and also the system-wide ETA. The data for the individual ETA in this graph is an average over the population of individual ETA numbers calculated as moving averages over the past [**g-dt-for-ind-stats**] number of ticks. The data for the system-wide ETA is calculated on a tick-by-tick basis.
- **Group 3** – Individual ETA – This histogram with its accompanying three monitors shows the distribution of the instantaneous individual ETA of the population of seekers, calculated as moving averages over the past [**g-dt-for-ind-stats**] number of ticks. Note that the Carnot-style efficiency clusters at a value just less than one, in accordance with Odum's predictions (Ref T) about "optimum efficiency" rather than "maximum efficiency". His argument was that individual energy transfer processes within an ecosystem will evolve to function at maximum (preservative) power, and not at maximum efficiency. This demonstrates the PoMPE part of the phenomenon that he called the maximum power principle (or MPP).
- **Group 4** – Individual EROI – This histogram with its accompanying three monitors shows the distribution of the instantaneous individual EROI of the population of seekers, calculated as moving averages over the past [**g-dt-for-ind-stats**] number of ticks. Note that the Hall-style efficiency clusters at a value around 1, but with a maximum number of seekers just above 1, as indicated by the manually-placed vertical red line.
- **Group 5** – this group consists of:
  - **A button -** "Clear these plots", which clears the data from the two line graphs, but leaves the histograms as is. I use this when I am no interested in this data and it is causing the model to slow down.
  - **A switch –** [**gb-include-dnd**] – this switch causes the effects of the decomposers and detritivores to be taken into account. When it is turned on, the system-wide EROI drops to exactly 1. In Panel 08, note how the system-wide EROI drops to 1 about 33,500 ticks (marked by a manually-placed vertical line). That is when I turned on the dnd switch.
  - **A monitor -** "Ticks", which tells me what tick of the model this is.
- **Group 6 – Standard Buttons** – there is a set of three standard control buttons as described in Panel 01: Setup, One Tick and Go/Stop.

5.9 - Panel 09 – LTE Data, With Breakout By Type Of Death (#1)

This panel displays two groups of three histograms each, each histogram accompanied by three monitors. LTE stands for Life-Time Efficiency. This data is captured at time of death of a seeker, and is tabulated over and extended length of time – over many generations. It is tabulated by type of death: hunger (DBH), old age (DBO) and reproduction (DBR):

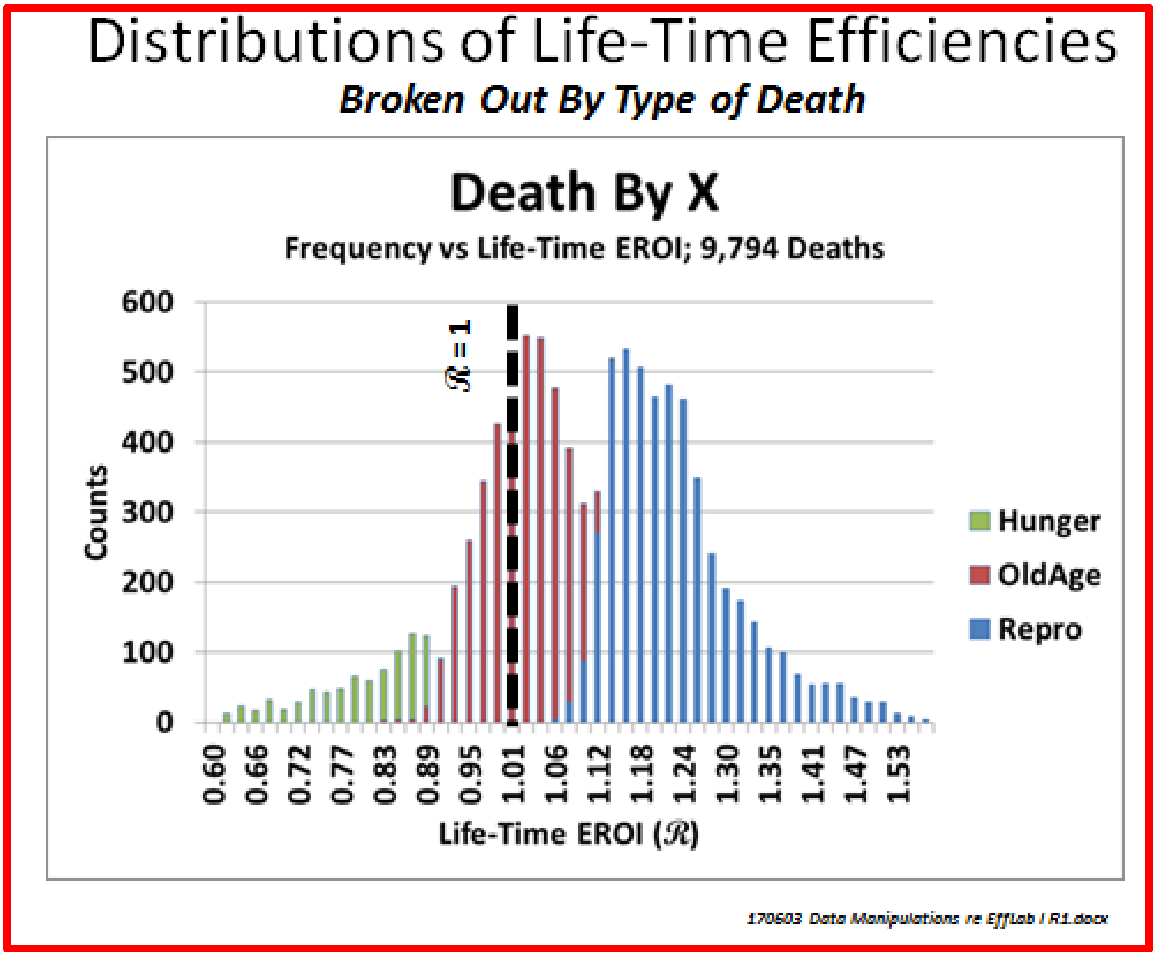
- **Group 1** – Life time efficiency – ETA – By Type of Death:
  - **ETA of DBH** – ETA can be negative, because a seeker who starts with an endowment can lose it very quickly. It lies somewhere in the range [-∞, 1]. I understand that negative efficiencies are not what most engineers consider to be within a reasonable range of values for ETA, since they would never make a machine that is so inefficient. Nature does. I explore that issue in my diary notes at Refs C, D and E. Those that die by





hunger are so inefficient that they consume energy at a higher rate than they gather it, and so run through their initial endowment, run out of energy, and die of hunger.
  - **ETA of DBO** – ETA lies in the range [-0.2, 0.2]. These seekers manage to "hang in there" until old age, and then die. Some have a little more energy than they started with. Some have a little less. None are "wealthy" (economic interpretation) or "healthy" (biophysical interpretation).
  - **ETA of DBR** – ETA lies in the range [0, 0.4]. These are the best of the best, managing to collect enough energy to attain reproductive status. Nevertheless, their life time efficiency runs at about 10% to 20%. Very interesting!
  - The nine monitors at the base of these three histograms show the minimum value, average value, and maximum value, for each graph.
- **Group 2** – Life time efficiency – EROI – By Type of Death:
  - **EROI of DBH** – EROI can be in the range [0, ∞]. As I said above, those that die by hunger are so inefficient that they consume energy at a higher rate than they gather it, and so run through their initial endowment, run out of energy, and die of hunger. These have an LTE EROI of less than 1.
  - **EROI of DBO** –These seekers manage to "hang in there" until old age, and then die. Some have a little more energy than they started with. Some have a little less. None are "wealthy" (economic interpretation) or "healthy" (biophysical interpretation). Their EROI is high enough to keep them alive, but just.
  - **EROI of DBR** – The EROI of these seekers lies in the range [1, 1.6]. These are the best of the best, managing to collect enough energy to attain reproductive status. Nevertheless, their life time EROI efficiency rarely exceeds 1.6.
  - The nine monitors at the base of these three histograms show the minimum value, average value, and maximum value, for each graph.
- **Group 3** – there is a set of three standard control buttons as described in Panel 01: Setup, One Tick and Go/Stop.
- **Group 4** – there is a button to clear the LTE data. LTE data is collected for a number of agents up to the number set by [**g-max-for-lte-stats**] in Panel 02. This means it collects that many data points for each mode of death: DBH, DBO and DBR. As each seeker dies, the data is added to the correct data list until all three data lists are filled. This button empties those data lists, and causes the collection of such data to start over. Since often the first several generations of seekers are biased, it is best to run the model until there is clear competition for energy, and then clear the LTE data, and only then, collect the statistics.
- **Group 5** – There are six monitors to be used to guide the collection of LTE data.

The LTE data can be processed using MS Excel 2010 to produce a combined histogram, as follows, which was presented during the ISBPE conference of 2017.



5.10 - Panel 10 – LTE Data, With Breakout By Type Of Death (#2)

This panel displays five groups controls, continuing to display the LTE data from Panel 09.

- **Group 1** – Life time efficiency – ETA – All Types of Death:
  - **ETA – All Types** – This histogram does not break out by type of death, but, rather, shows the aggregate data.
  - The three monitors at the base of the histogram shows the minimum value, average value, and maximum value.
- **Group 2** – Life time efficiency – EROI – All Types of Death:
  - **EROI – All Types** – This histogram does not break out by type of death, but, rather, shows the aggregate data.
  - The three monitors at the base of the histogram shows the minimum value, average value, and maximum value.
- **Group 3** – Mean ETAs – this is a line graph of historical values of the ETA, broken out by type of death.
- **Group 4** – Mean EROIs – this is a line graph of historical values of the EROI, broken out by type of death.
- **Group 5** – A button “Clear These Plots” that will empty the two line graphs of their data.

- **Group 6** – there is a set of three standard control buttons as described in Panel 01: Setup, One Tick and Go/Stop.

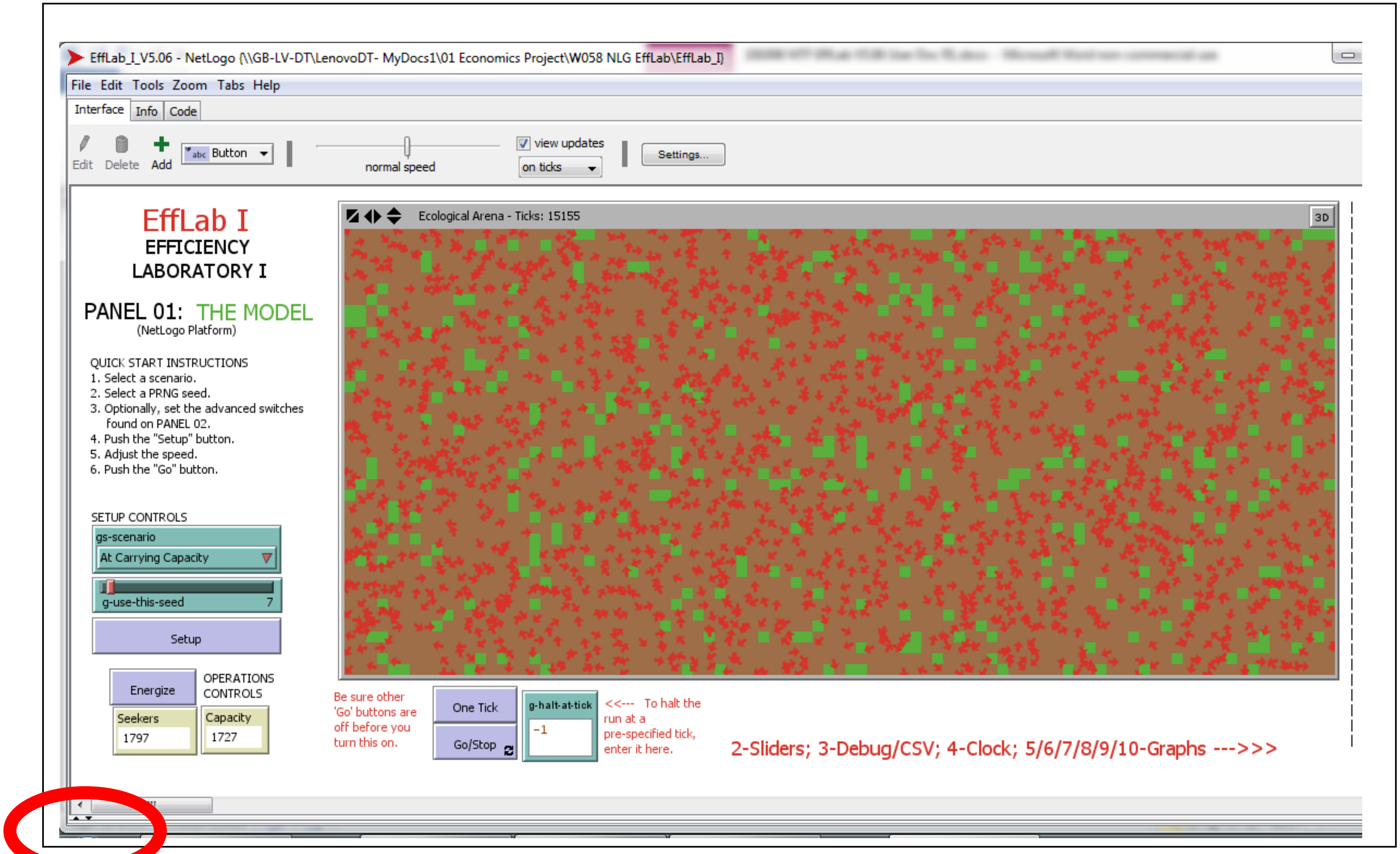
5.11 - Panel 11 – Entropy of Energy and of Genetic Material

This panel displays three groups:

- **Group 1** – Entropic Index of Energy Distribution:
    - **Energy Distribution** – This is the distribution of the nrg in the seekers. As noted before for Panel 05, this distribution is very similar to the distribution of energy in a gas of ideal atoms, and demonstrates the Maxwell-Boltzmann energy distribution.
    - **Energy Entropic Index** – This is an experimental graph, using the techniques developed in my model EiLab, and discussed in my diary notes at Refs N through R. All seekers start with the same energy, so the initial distribution is uniform. That has entropic index of exactly 1. However, as steady state distribution is approached, the entropy falls to a sustainable value as approximately 0.93. That is the entropic index of a typical Maxwell-Boltzmann distribution.
- **Group 2** – Entropic Index of the Phenotype Distribution:
    - **Experimental Study** – This is a new experiment that I have not yet studied, or formed a final opinion on. I am thinking that there is a kind of "information entropy" that can be associated with shapes and structures and organized patterns that appear in nature. When a population of organisms evolves, non-random patterns appear. The Maxwell-Boltzmann is an example of a non-random pattern. Certainly, randomness is necessarily involved in the emergence of such patterns, but other driving factors and constraints seem to determine the nature of the pattern. It is not just a purely random pattern. In many cases uniform distributions would be most typical of pure unconstrained randomness. The Maxwell-Boltzmann arises when the conservation of energy applies a constraint on the distribution of energy, it seems. When a particularly efficient search pattern is encoded in genes, and when that phenotype is passed on from generation to generation in a largely intact form, then a pattern associated with an effective phenotype emerges, and it has an entropic index.
    - **Average Phenotype** – This graph is a graph of a histogram showing the relative sizes of the phenotypic values (fractional – non-percentage – form), averaged across the entire population of currently living seekers. Each of the eight genes in the C1 chromosome are labelled as Forward (F), Forward Right (FR), Right (R), Back Right (BR), Back (B), Back Left (BL), Left (L), and Forward Left (FL). This is another representation of the average "clock face". Compare this graph with the "clock faces" of Panels 04 and 07.
    - **Phenotype Entropic Index** – This is an experimental graph, using the techniques developed in my model EiLab, and discussed in my diary notes at Refs N through R. All seekers start with the same phenotype.
- **Group 3** – there is a set of three standard control buttons as described in Panel 01: Setup, One Tick and Go/Stop.

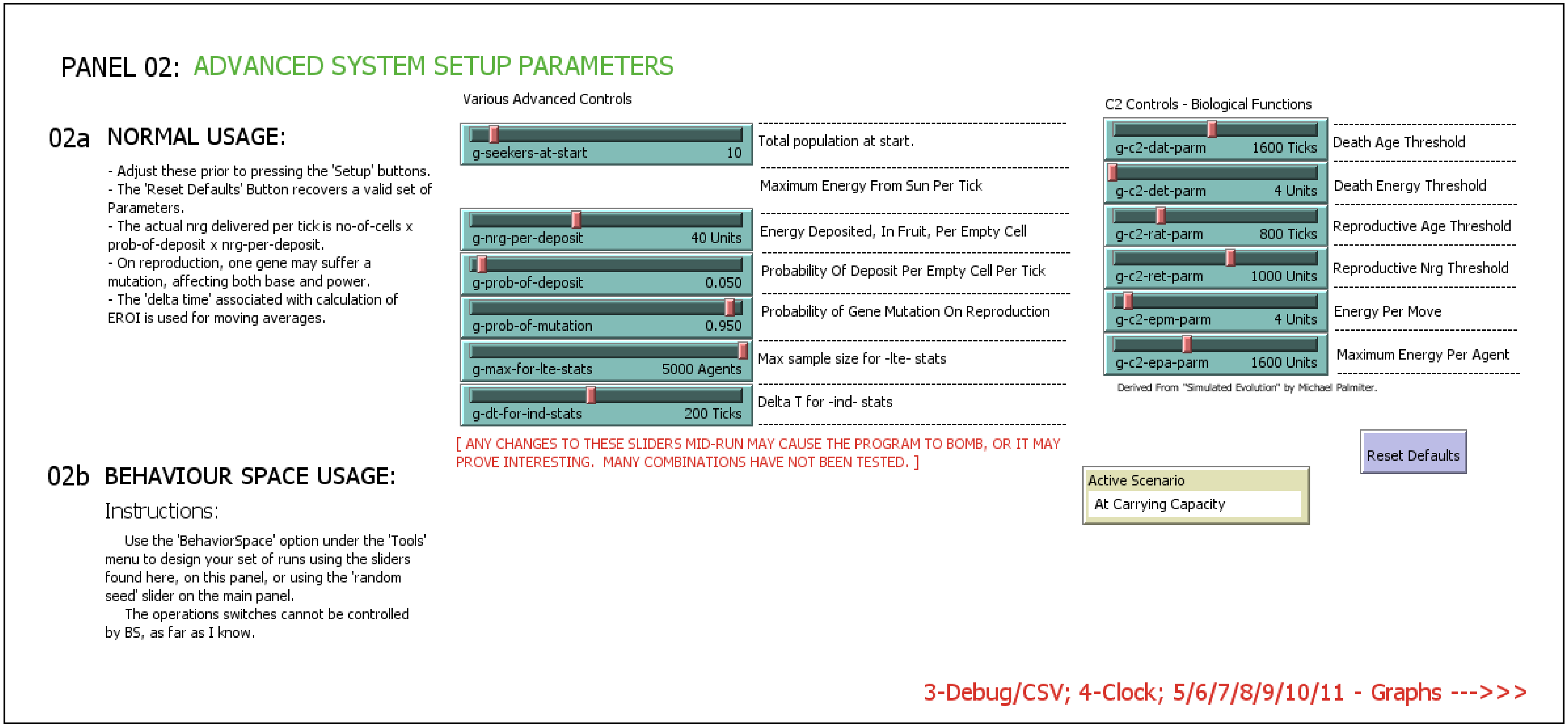
6 - Annex A – Examples of User Interface Panels

6.1 - Panel 01 – The Model



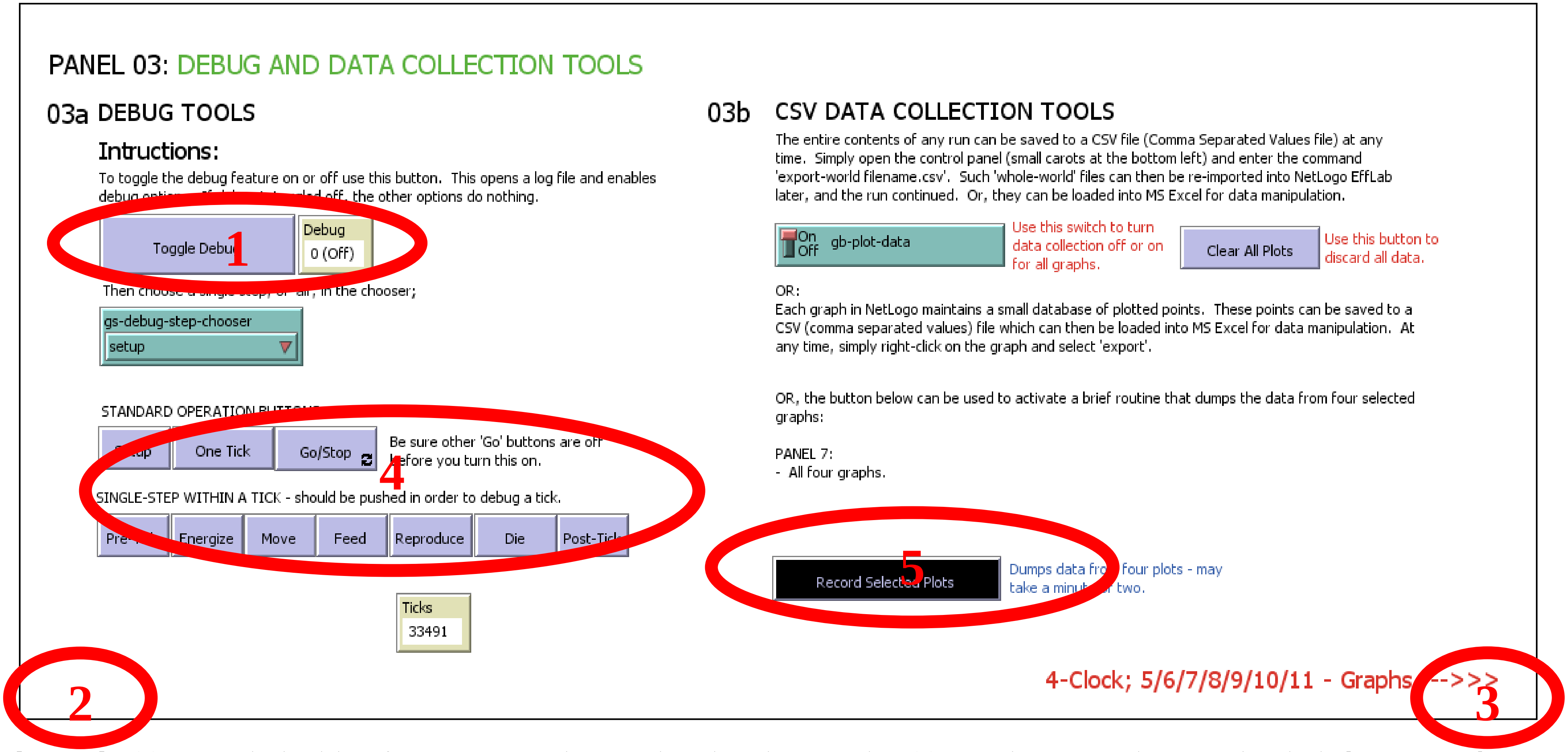
Note the small carets (upward and downward arrow heads) used to access the NetLogo "Command Centre" (circled in red).

6.2 - Panel 02 – Advanced System Setup Parameters



The control for "Maximum Energy From Sun Per Tick" has been removed.

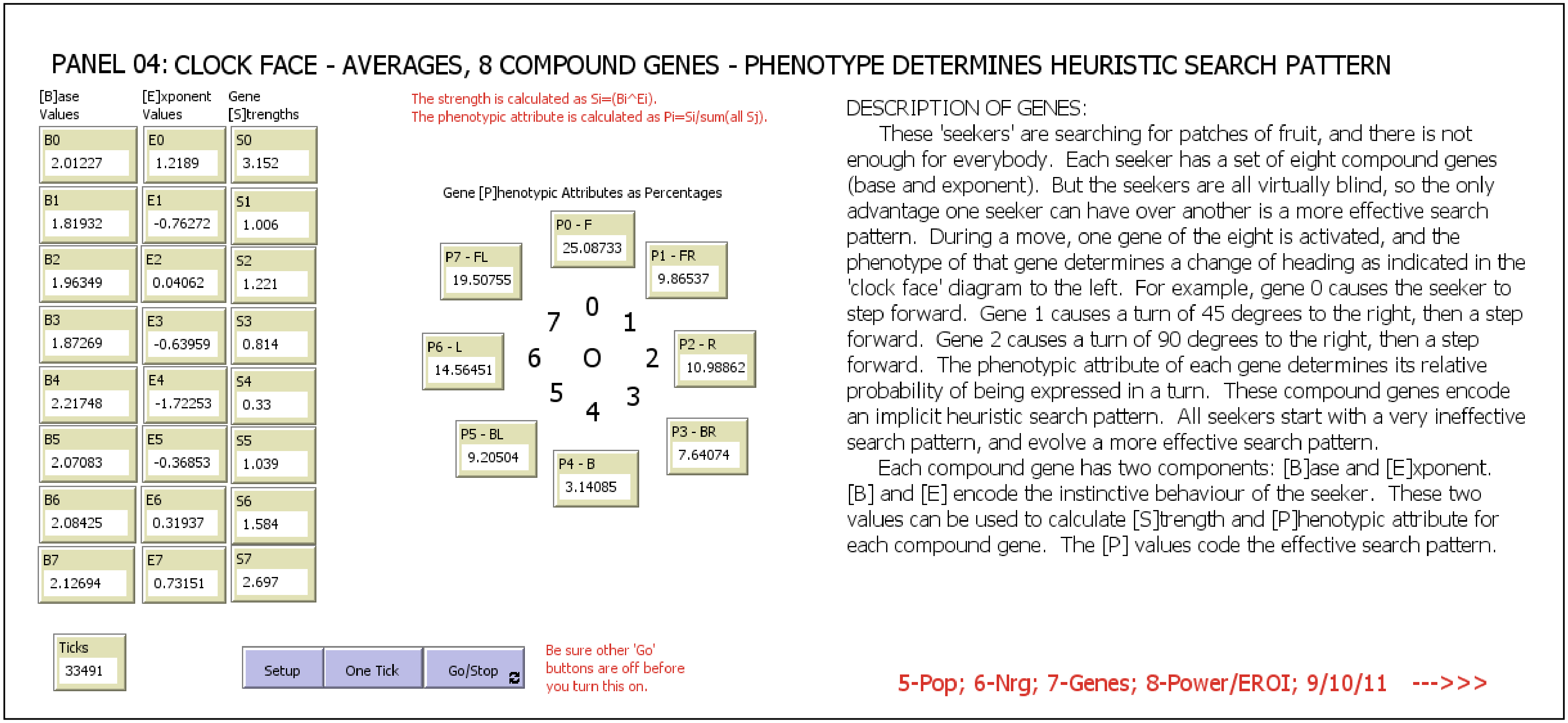
6.3 - Panel 03 – Debug and Data Collection Tools



Debug tools: (1) To toggle the debug feature on, press the 'Toggle Debug' button. Then (2) open the command centre using the little carots on the left (not shown). Then (3) make the command centre vertical using the angled arrows on the right. Then (4) use one of these buttons to run a part of the model.

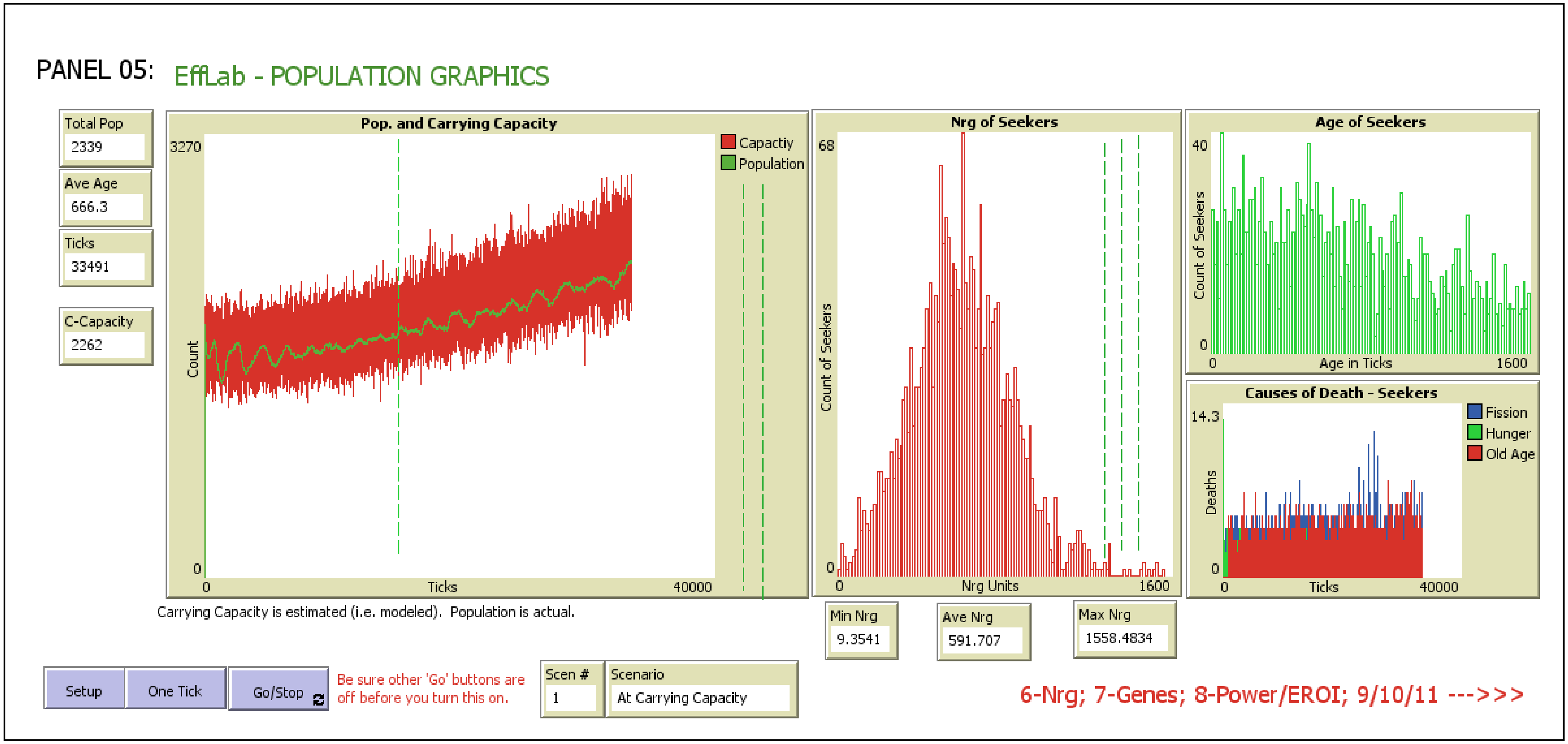
CSV tools: OR (5) Press the button to dump data to CSV files from the four selected graphs.

6.4 - Panel 04 – Clock Face – Averages, 8 Compound Genes



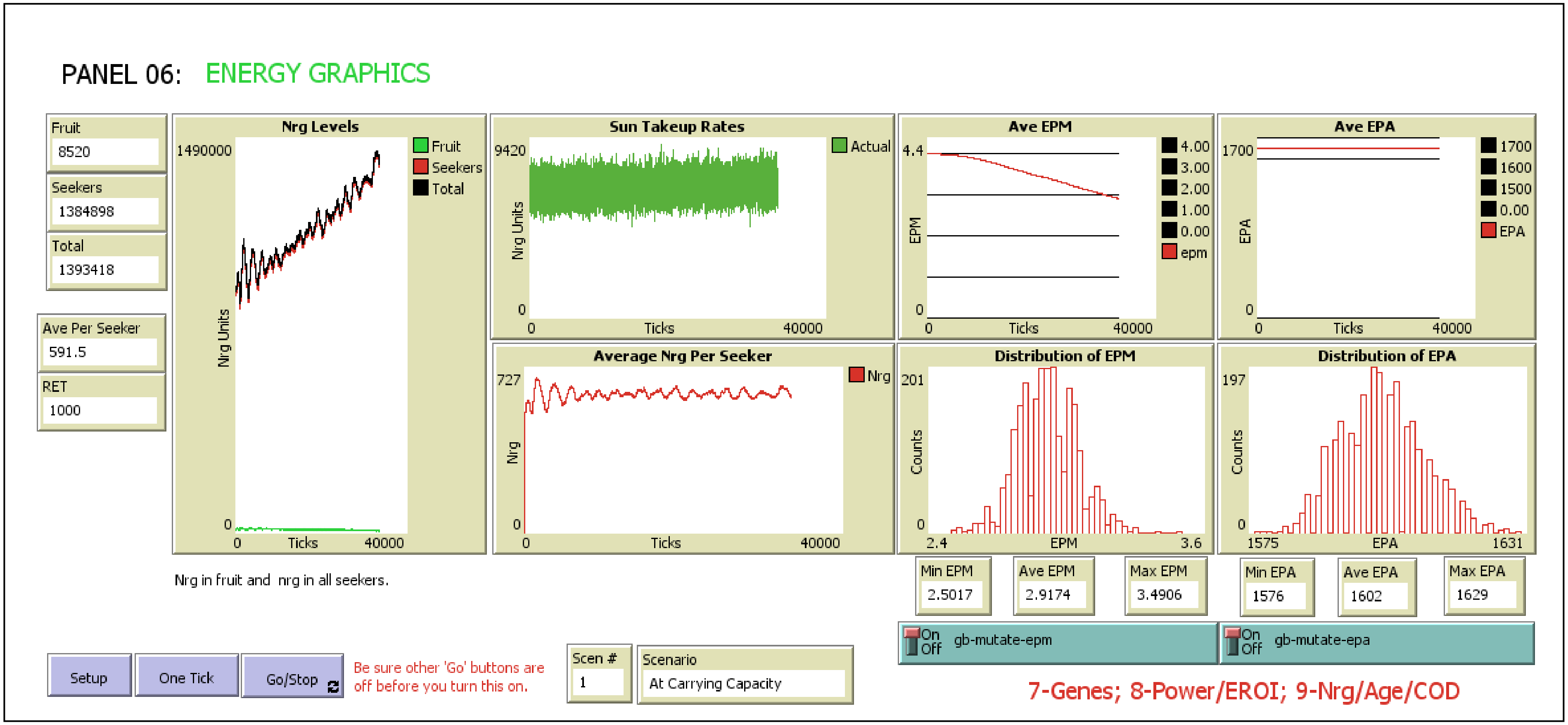
The clock face shows that a cruiser is emerging (large F and FL) as the dominant phenotype. It may eventually evolve to be a bent arrow. The C1 phenotype is referred to (by me) as a clock face. It can be represented in different ways. It appears in Panel 07 as “Phenotype value by genes”, and in Panel 11 as “Average phenotype”.

6.5 - Panel 05 – Population and Energy Graphics



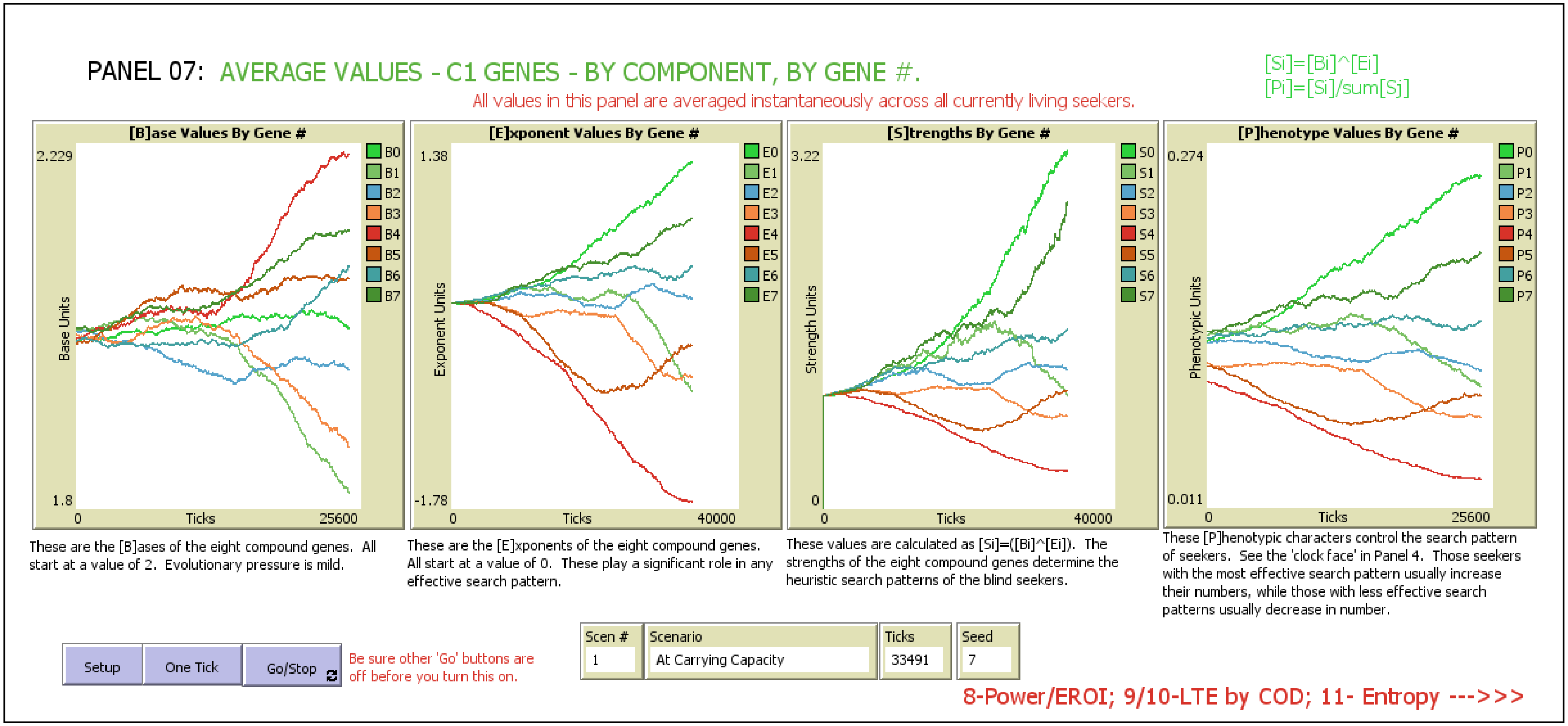
Note that the carrying capacity increases. This is an effect that contributes to Jevon's paradox, and also to the 2-part effects of the maximum power principle.

6.6 - Panel 06 – Energy Graphics



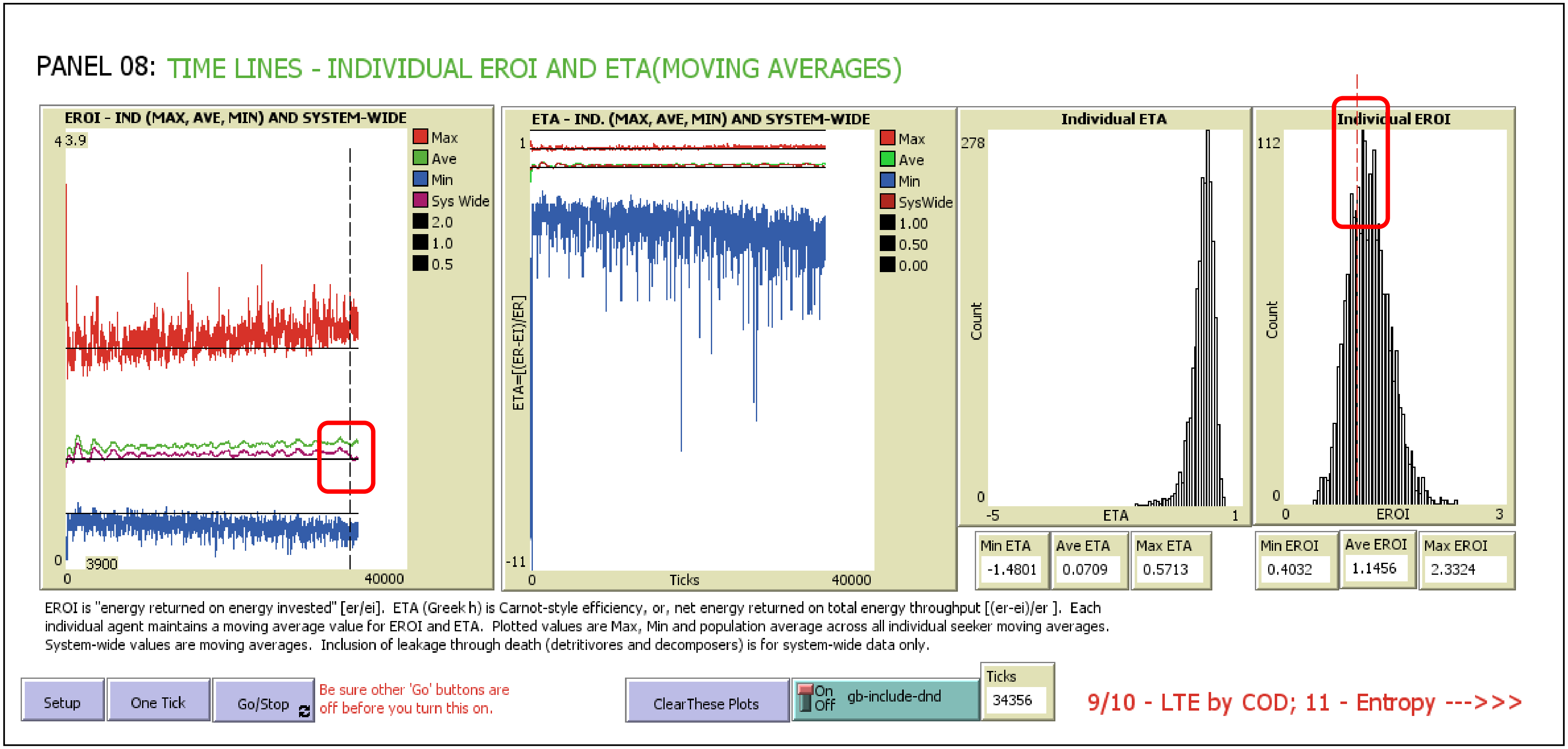
Nrg Levels of the seekers displays a damped oscillation about a longer-term steady-state value. Power (energy consumption) in this model is directly related to population size and the average C2 gene called EPM.

6.7 - Panel 07 – Average Values – C1 Genes – By Component, By Gene



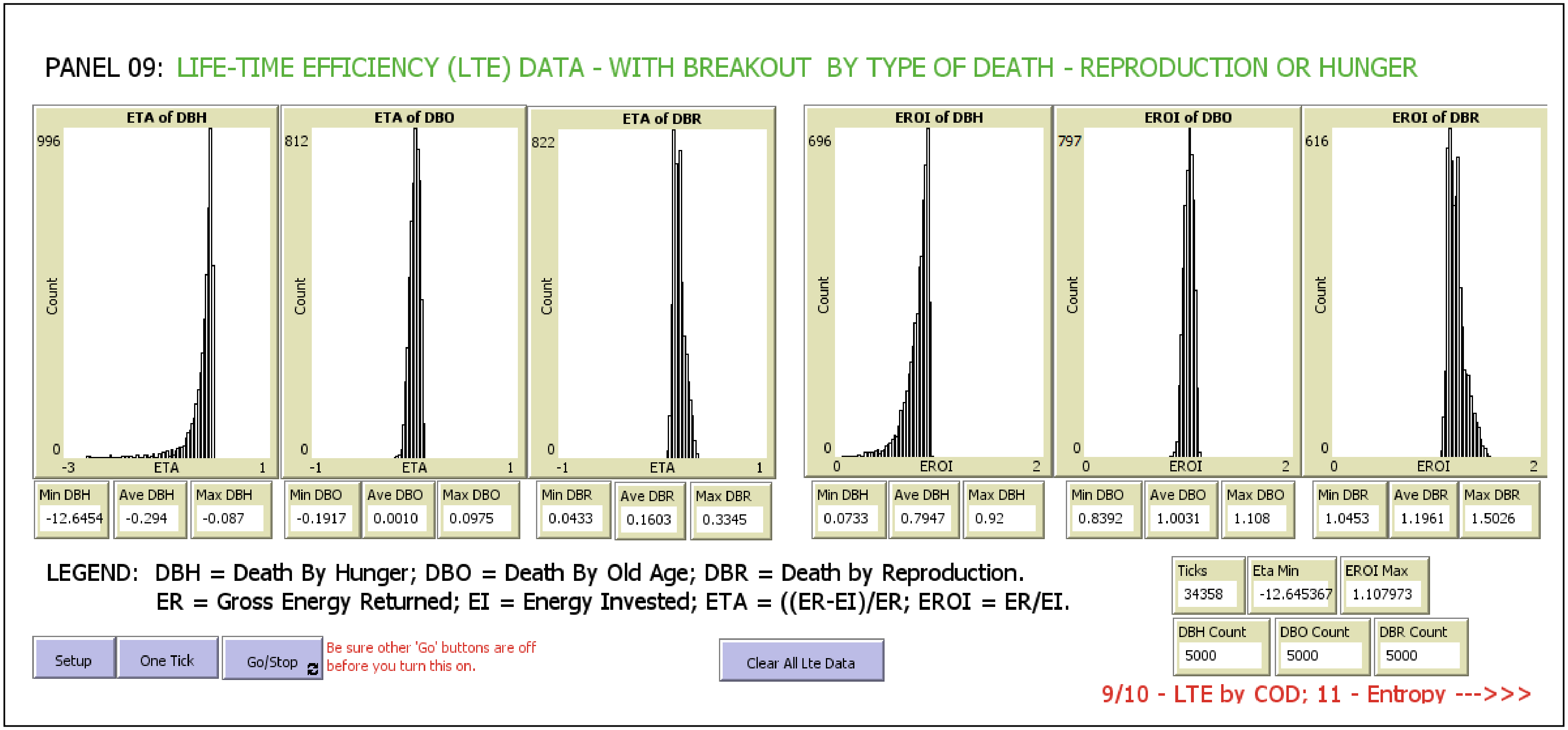
Here we see the evolving structure of the C1 chromosome. The [P] values display the signature of an emerging cruiser phenotype. It is another representation of the “clock face” of the C1 chromosome, as also seen in Panels 04 and 11.

6.8 - Panel 08 – Power and EROI (Moving Averages)



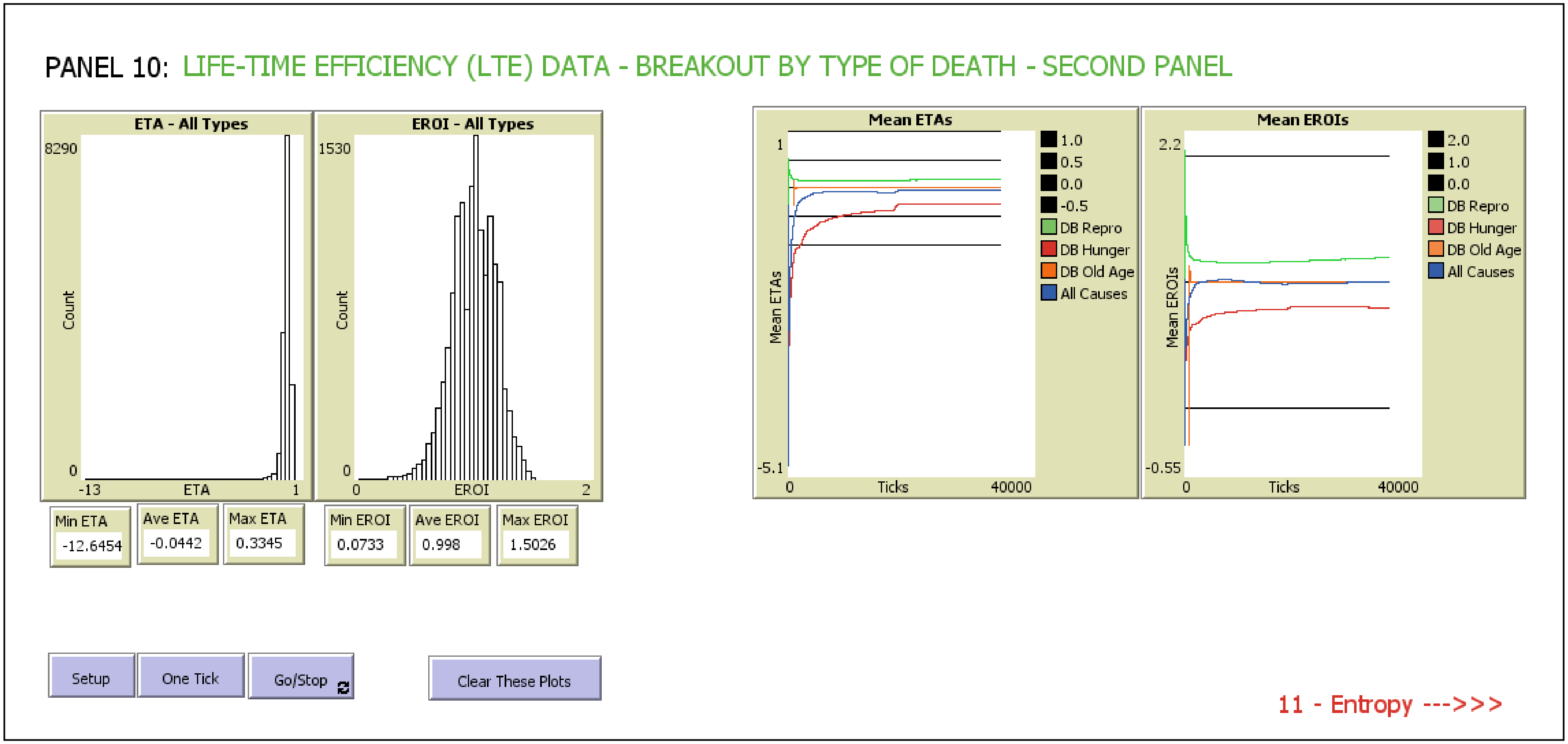
The lifetime average EROIs of the individuals in the society, and of the society as a whole, remain close to 1. The power limbs towards a maximum.

6.9 - Panel 09 – LTE – Breakout by Type of Death (#1)

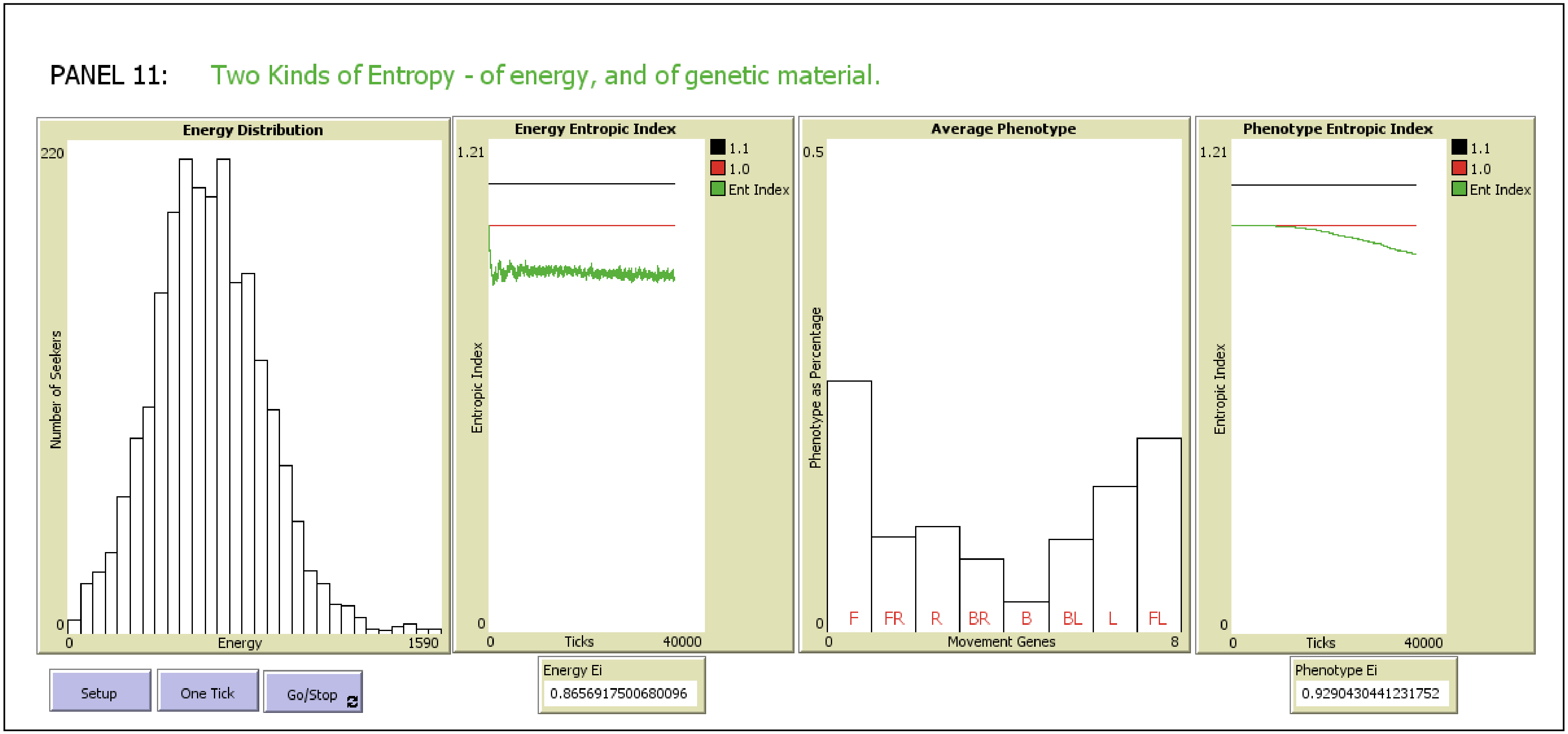


The distribution of nrg among the seekers is the now-familiar pattern of the Maxwell-Boltzmann distribution. It is much harder to decide what the distribution of ages is. Note that when the original population overshot, there was a die-off (the little red bump) and the synchrony of the births and death was gradually broken as many generations of seekers died of old age and hunger thereafter.

6.10 - Panel 10 – LTE – Breakout by Type of Death (#2)



6.11 - Panel 11 – Nrg & Age Histograms, and Causes of Death



The distribution of nrg among the seekers is the now-familiar pattern of the Maxwell-Boltzmann distribution. The histogram of the C1 phenotype can also be represented as a “clock face” (see Panel 04) or as a set of time lines (see Panel 07).