

Main differences to the original model:

- periodic boundary conditions.
- topological interaction range, that defines the 'neighborhood' of an individual (easier to parameterize the metric interaction ranges).
- genome comprises of three bit-strings (phenotype, compat and neutral) of length *Bits*.
- phenotype comparison based on bit-count (number of 1s), same matrix as original.
- genetic compatibility based on 'simple matching coefficient' (number of matching bits / *Bits*), equivalent to original.

Here a typical config-file:

```
# mutual/config.ini
# simulation configuration file

N = 65536          # number of individuals per guild
E = 65536          # number of attempted mating events per generation
G = 100000         # generations

update = "synchronous"      # "synchronous" or "asynchronous"

# define the spatial distance range
# range_type == "strict_topological": range_value = number of neighbors per guild,
#                                     distance range: N/A
# range_type == "topological":        range_value = approx. number of neighbors per guild,
#                                     distance range <- sqrt(range_value * N / pi)
# range_type == "euclidean":         distance range <- range_value
# range_type == "homogeneous":       range_value: N/A, distance range N/A
range_type = "strict_topological"    # distance metric, see above.
range_value = 20                    # range value
spacing = "grid"                    # spacing, "random" or "grid"

animal.phenotype_dist = c(-1, 0)    # tolerable mutual phenotype distance
animal.threshold = 0.97              # threshold genetic compatibility
animal.mutation_rate = 0.001        # mutation rate per bit

# init.x.type = "random":    complete random, init.x.bits ignored
# init.x.type = "binomial": bits set from binomial B(n,p) with n = init.x.bits, p = 1/2
# init.x.type = "strict":   exact init.x.bits bits set
# prefix with "clone_" to generate clones
animal.init.p.type = "binomial"
animal.init.p.bits = 256
animal.init.c.type = "strict"
animal.init.c.bits = 1024
animal.init.n.type = "random"
animal.init.n.bits = 0

plant.phenotype_dist = c(0, 1)      # tolerable mutual phenotype distance
plant.threshold = 0.97              # threshold genetic compatibility
plant.mutation_rate = 0.001        # mutation rate per bit

# init.x.type = "random":    complete random, init.x.bits ignored
# init.x.type = "binomial": bits set from binomial B(n,p) with n = init.x.bits, p = 1/2
# init.x.type = "strict":   exact init.x.bits bits set
# prefix with "clone_" to generate clones
plant.init.p.type = "binomial"
plant.init.p.bits = 64
plant.init.c.type = "strict"
plant.init.c.bits = 0
plant.init.n.type = "random"
plant.init.n.bits = 0
```

The slightly simplified 'algorithm' is:

- 1) pivot <- pick an animal at random
- 2) Pick an animal from the neighborhood of pivot and determine whether they are genetically compatible and of opposite sex. Goto 5) on failure.
- 3) Pick a mutualistic partner (plant) from the neighborhood of pivot at random. Goto 5) on failure.
- 4) If the plant and animal are not morphologically compatible, goto 5). Pick a random animal from the neighborhood of pivot to be replaced by the offspring and choose the sex at random.
- 5) pivot <- pick a plant at random
- 6) Pick a plant from the neighborhood of pivot and determine whether they are genetically compatible and of opposite sex. Goto 1) on failure.
- 7) Pick a mutualistic partner (animal) from the neighborhood of pivot at random. Goto 1) on failure.
- 8) If the plant and animal are not morphologically compatible, goto 1). Pick a random plant from the neighborhood of pivot to be replaced by the offspring and choose the sex at random.
- 9) Go to 1)

Some observations:

Simple matching coefficient (SMC) is a very strict distance measure. We need a lot of bits (*Bits* >= 512) to ensure that we can find compatible individuals at all. However, that supports speciation.

Drift is strong in this model and causes the bit-strings to become 50/50 ones and zeros. That seems to be especially crucial for the 'mutualistic' part of the model: *there is no diversification based on the phenotype*. While intraspecific trait-values are more similar than interspecific ones, the two guilds become *globally* phenotypically compatible. Not so good.

Equilibrium (in terms of number of species) is reached quickly – 10000 generations should be enough.

IMHO, the most interesting aspect of this model is space and how evolution depends on the interaction range. The topological range must be pretty small to support many species. Too big, and the simulations result in single-species per guild. The transition is abrupt and it seems that we have a proper phase-transition based on topological range.

Phase transition and hysteresis experiment

Starting with topological range of 5, I let the simulation run for 10000 generations, took a snapshot, increased the topological range by 5 and loop until max topological range. Then, we do the same thing backwards, with decreasing topological range.

Hyst_16384_1024.pdf shows a typical outcome of this procedure. The blue line in the 'log-number of species vs topological range'- graph is the 'up'-path (increasing topological range), the red line is the 'down'-path (decreasing topological range). The pictures showing the distribution of (animal) species in space. The lighter the higher the abundance.

Hyst.pdf contains the hysteresis-curves for different number of individuals (N) and number of Bits in the genes.

Similar.pdf seems to show that simulations with different N are *similar* – the species distribution for N=4096 looks like a scaled-down version of the distribution for N=16384!

Hyst_nx.pdf compares simulation with and without crossing-over. Without crossing-over, the wonderful phase-transition is gone!