Haploid & diploid recombination and their evolutionary impact

W. Garrett Mitchener
College of Charleston Mathematics Department
MitchenerG@cofc.edu
http://mitchenerg.people.cofc.edu

Introduction

The basis of biological computation is the reaction or regulatory network. How are such networks discovered by selection-mutation processes?

The Utrecht Machine (UM) is a discrete abstraction of a gene regulatory network. For this project, an evolutionary simulation is used to discover UM-based agents that solve a data encoding problem.

UM State: Table mapping *patterns* p to integer *activation levels* A_p.

- A_p is metaphorically how many units of protein p are present

Reaction instruction: $A_s \geqslant \theta \implies \text{inc}\, A_p, \text{dec}\, A_q$

- s : switch pattern; θ : threshold
- p : pattern to activate, *p-up*; q : pattern to inhibit, *p-down*
- If $A_s \geqslant \theta$ then add 1 to A_p and subtract 1 from A_q
- All instructions performed simultaneously in discrete time steps
- Encoded as a binary genome and subject to mutation and recombination

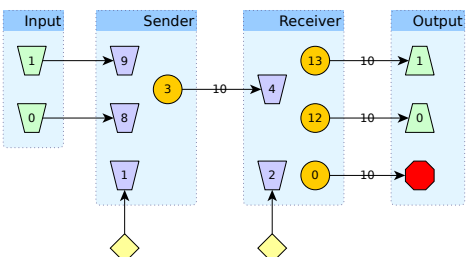
Channels: A bit of input is provided by increasing the activation of a particular pattern during each time step the input bit is set.

FYI: I call it the Utrecht Machine because the conference where I first presented it was EvoLang 2010 in Utrecht.

Experimental task: Transmit 2 bits over time

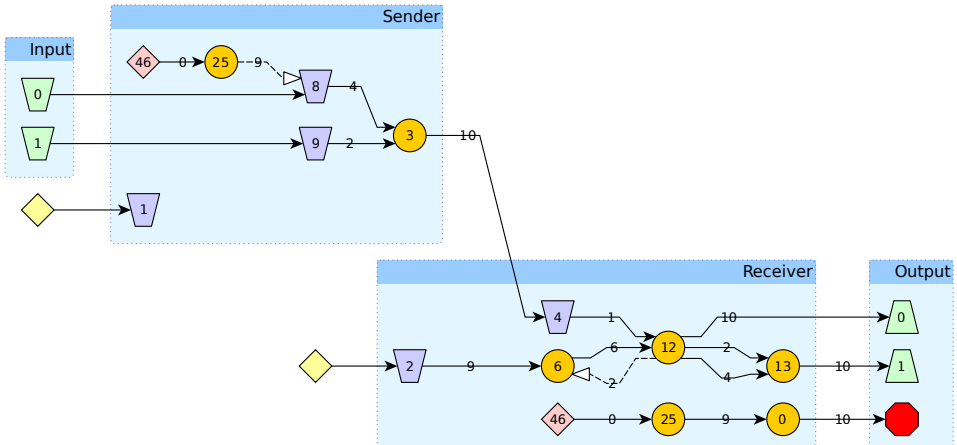
Use a selection-mutation process to evolve solutions to a sequential coding problem. A genome is built into an *agent* follows: Two UMs are built from the genome, one sender and one receiver. The sender gets two bits of input, plus constant input into its role pattern 1. The receiver must generate two bits of output that reproduce the input, and signal when to stop. The receiver gets input from the sender through a single synapse, plus constant input into its role pattern 2.

Sketch of agent & scoring details

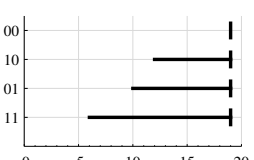


Each agent is presented with all four possible input words ⟨00⟩, ⟨10⟩, ⟨01⟩, ⟨11⟩, starting at a zero state for each and running for up to 100 time steps. It earns $10,000$ points for each bit correctly transmitted, plus $100 - \max(20, t)$ each time it stops after t steps. Maximum possible score: $10,000 \times 4 \times 2 + 80 \times 4 = 80,320$.

Example evolved solution



An evolved solution (above) and its synaptic code (right).

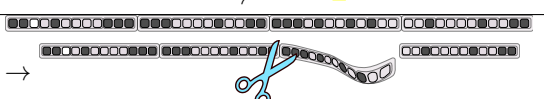


Genetic encoding

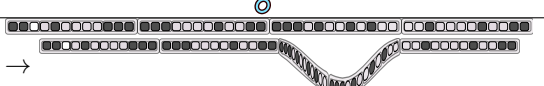
- Each instruction ⇔ one gene
- Integers represented by Gray code
- Further majority-of-three redundant encoding

Supported mutations

single bit substitutions
prob 0.005 per genome bit



gene (instruction) deletion
prob 0.001 per gene

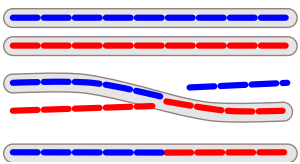


gene (instruction) duplication
prob 0.001 per gene

Recombination processes

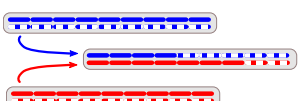
Genetic recombination accelerates discovery of reaction networks. Nature uses many recombination mechanisms. Do the details matter?

Recombination variant: haploid crossover



Agents have a single chromosome. Chromosomes from each parent are aligned at the beginning and split at a random location. The first part of one is attached to the second part of the other to form the child chromosome.

Recombination variant: diploid crossover



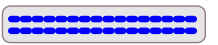
Agents have pairs of chromosomes. These are mixed and matched within each parent, then each parent contributes one mixed chromosome to the child pair.

Genome configurations

Genome configurations are specified by how many bundles of chromosomes (#b), how many chromosomes per bundle (#c), and how many genes initially present in each chromosome (#g). Crossover is haploid for 1c configurations, diploid for 2c.

1. *Haploid* configuration: 1b 1c 32g

2. *Short diploid* configuration: 1b 2c 16g



3. *Long diploid* configuration: 1b 2c 32g

In diploid genomes, genes come in pairs, and recombination increases the chances of having similar genes in each pair locus. So due to dynamics, a diploid genome has less effective information capacity than a haploid genome with the same number of genes but unpaired loci. So we'll test these two diploid configurations against one haploid configuration.

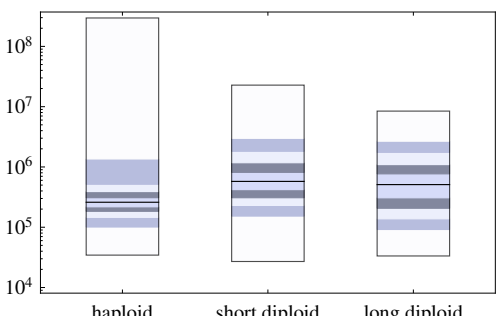
Initially, each agent in the *Haploid* configuration 1b 1c 32g has a total of 32 genes and 32 loci. In the *Short diploid* configuration 1b 2c 16g, each agent initially has 32 genes but 16 pair loci. In the *Long diploid* configuration 1b 2c 32g, each agent initially has 32 pair loci but 64 genes total.

Genome lengths change over time due to gene duplication and deletion mutations. The resulting genomes can vary in length over orders of magnitude.

Aggregate analysis

- Look at 1000 sample runs using each genome configuration
- Results reported by medians and on log-scale
  (less sensitive to extreme outliers)

Time to first perfect solution

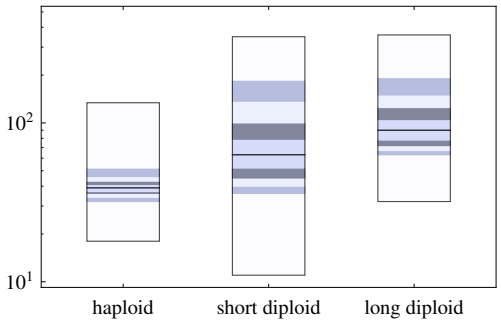


Decile charts for the number of agents searched before finding one that scores perfectly, logarithmic scale. Diploid configurations have wider spread but fewer extreme outliers. Haploid configuration has distinctly lower median.

Genome size of first perfect solution

Decile charts for total number of genes in first perfect solution, logarithmic scale.

Medians	
Haploid	39
Short diploid	63
Long diploid	90



Haploid gives shortest on average, which is to be expected since diploid configurations tend toward redundancy. But the diploid runs are not simply twice as long: Median length under short diploid is less than twice as long as haploid, but under long diploid is more than twice as long.

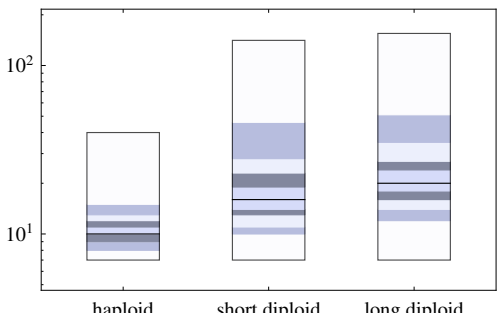
Genomes spontaneously accumulate significant length under diploid configurations for reasons that are not yet known.

Efficacious network size of first perfect solution

Genes connected directly or indirectly to output channels are called *efficacious* and form the regulatory network of interest. Other genes have no regulatory purpose and are called *inefficacious.*

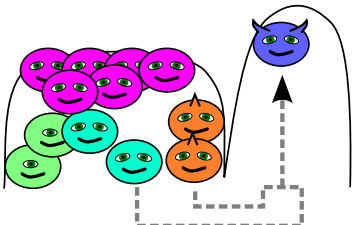
Decile charts for number of efficacious genes in first perfect solution, logarithmic scale.

Medians	
Haploid	10
Short diploid	16
Long diploid	20.5



Overall, the haploid configuration yields smaller efficacious networks. Median count of efficacious genes under short diploid configuration is less than twice as many as haploid; under long diploid, just above twice as many.

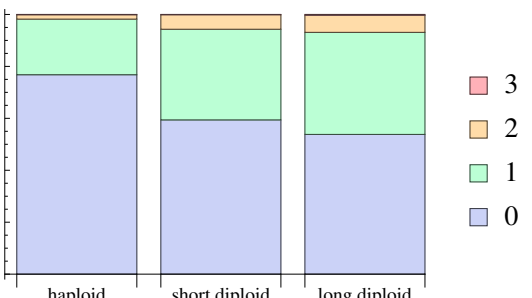
Innovations through outliers



During the long equilibrium phases, most of the population hovers near the peak of a ridge in the fitness landscape. Genetic diversity ensures that there are always a few *outliers* that don't achieve the highest score present in the population at that time. But they are more likely to be near the edge of the basin of attraction of the ridge, in which case their offspring are more likely to jump to another ridge—an innovation.

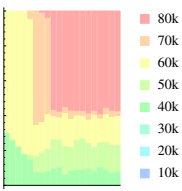
Chart of how many sample runs had n major innovations where at least one parent is an outlier.

Under diploid configurations, more samples have at least one such major innovation.



Case studies: Dynamics & genetic diversity

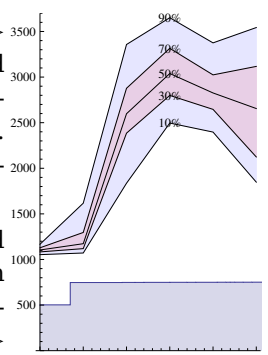
- Look at a few sample runs in detail
- Look at many agents from the same population



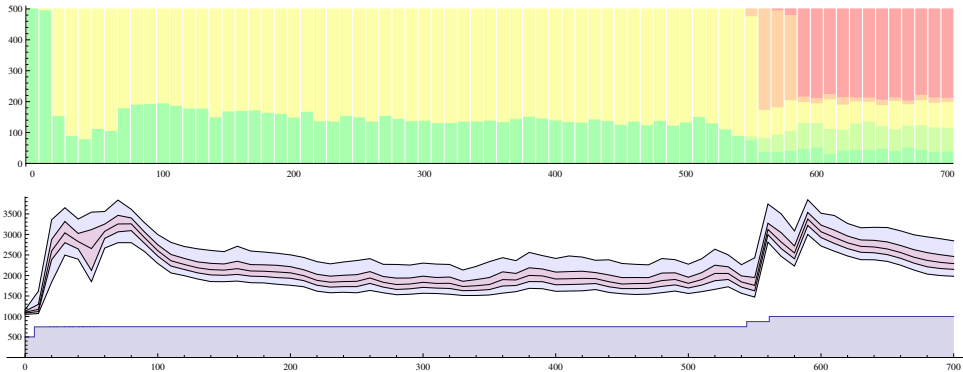
⇐ **Rating trajectory charts**
In each generation, agents have a variety of scores between 0 and 80320. Most of the score comes from the communication task. These charts show the distribution of these correctness scores as stacked bars, every 10 generations.

Alignment distribution charts ⇒
The bitstring genomes in each generation can be compared pairwise using the Smith-Waterman string alignment & scoring algorithm (high alignment score ⇔ similar strings). These charts show the distribution of alignment scores, every 10 generations.

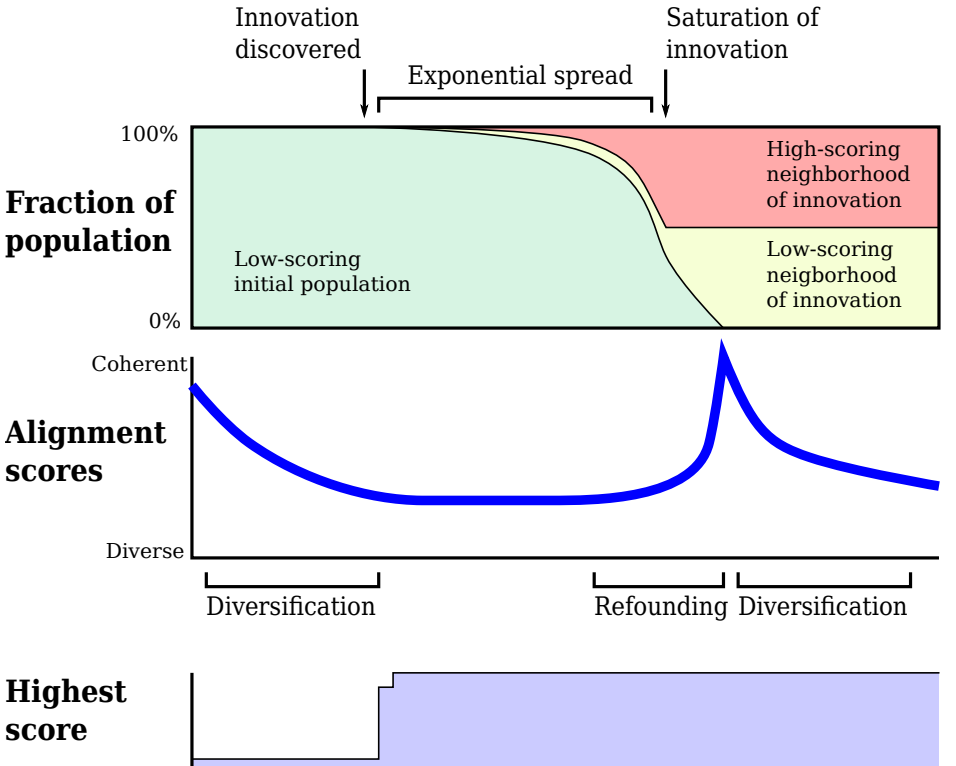
The graph at the bottom shows the highest rating achieved by any agent so far, not to scale. An innovation occurs when a new agent correctly transmits more bits than any that preceded it. Its rating is much higher, shown as a jump. ⇒



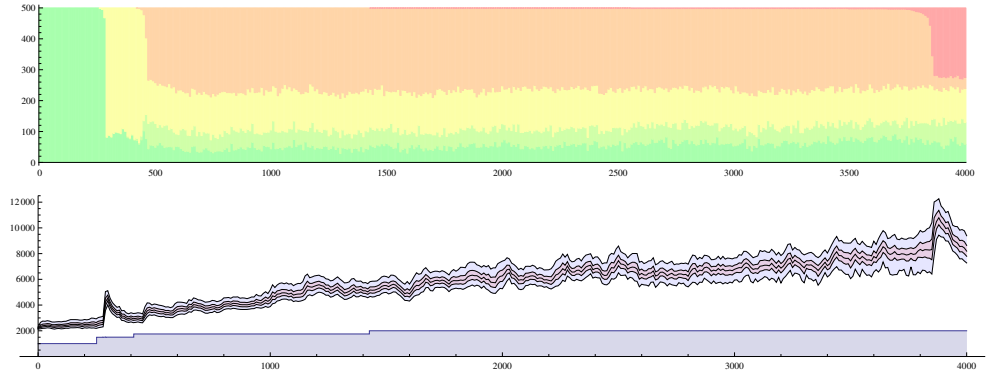
Haploid



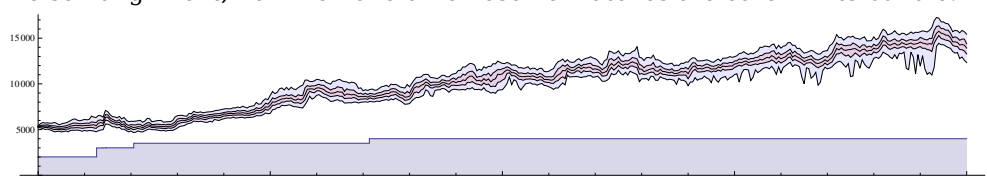
Each time an innovation is discovered, its descendants take over the population, resulting in a temporary decrease in diversity and spike in alignment scores (the “founder” effect). Since reproduction is binary, there can be a long delay between the innovation and the time it saturates the population with a large proportion of high-scoring descendants. The loss of diversity continues past the time of saturation. Greater diversity increases the probability that an innovation will be discovered. Sometimes a second innovation is discovered before the re-founding.


Innovation discovered
Saturation of innovation
Exponential spread
100%
Fraction of population
0%
High-scoring neighborhood of innovation
Low-scoring neigborhood of innovation
Low-scoring initial population
Coherent
Alignment scores
Diverse
Diversification
Refounding
Diversification
Highest score

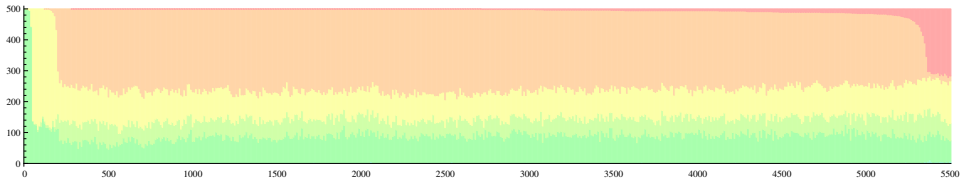
Short diploid

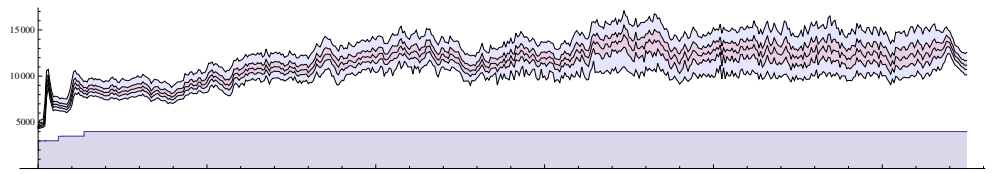


Diploid genome configurations usually lead to an overall upward drift in genome length and therefore in alignment scores. It also makes sense to examine self-alignment, how well one chromosome matches the other in its bundle:

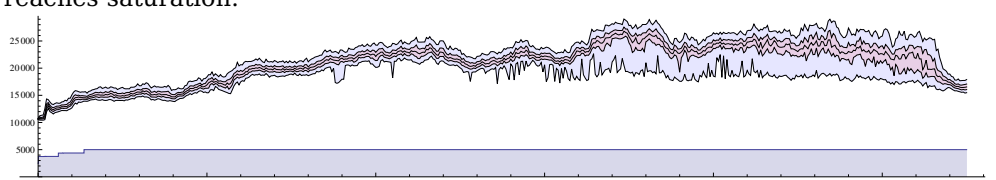


Long diploid





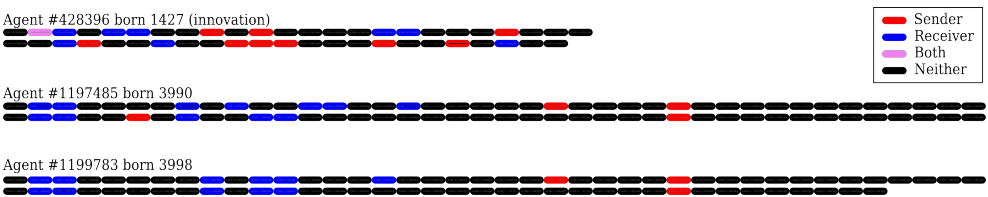
This sample run develops spread in self-alignment before the last innovation reaches saturation:



Genome organization

After finding the first perfectly scoring agent, evolution continues due to implicit pressures (ex: robustness against mutation). Sometimes the genome evolves to include spatial organization. Paired chromosomes become more similar. The sender mechanism becomes smaller. Genes for the sender mechanism become a cluster, separated from the larger receiver mechanism.

From the short diploid example, which happened to be especially clear:



References

W. Garrett Mitchener. A discrete artificial regulatory network for simulating the evolution of computation. In *EvoNet2012: Evolving Networks, from Systems/Synthetic Biology to Computational Neuroscience*, East Lansing, Michigan, USA, 2012. URL http://www.evosys.org/evonet2012_proceedings.pdf.