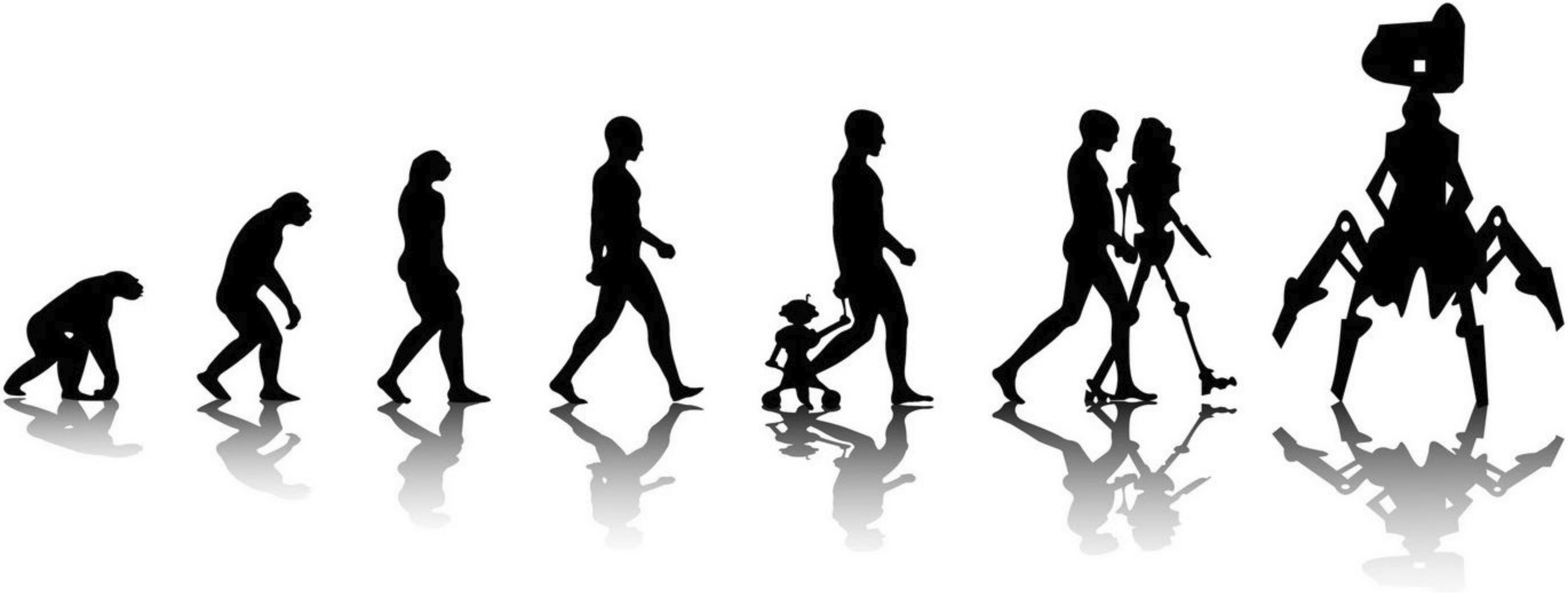


Modern Robotics: Evolutionary Robotics

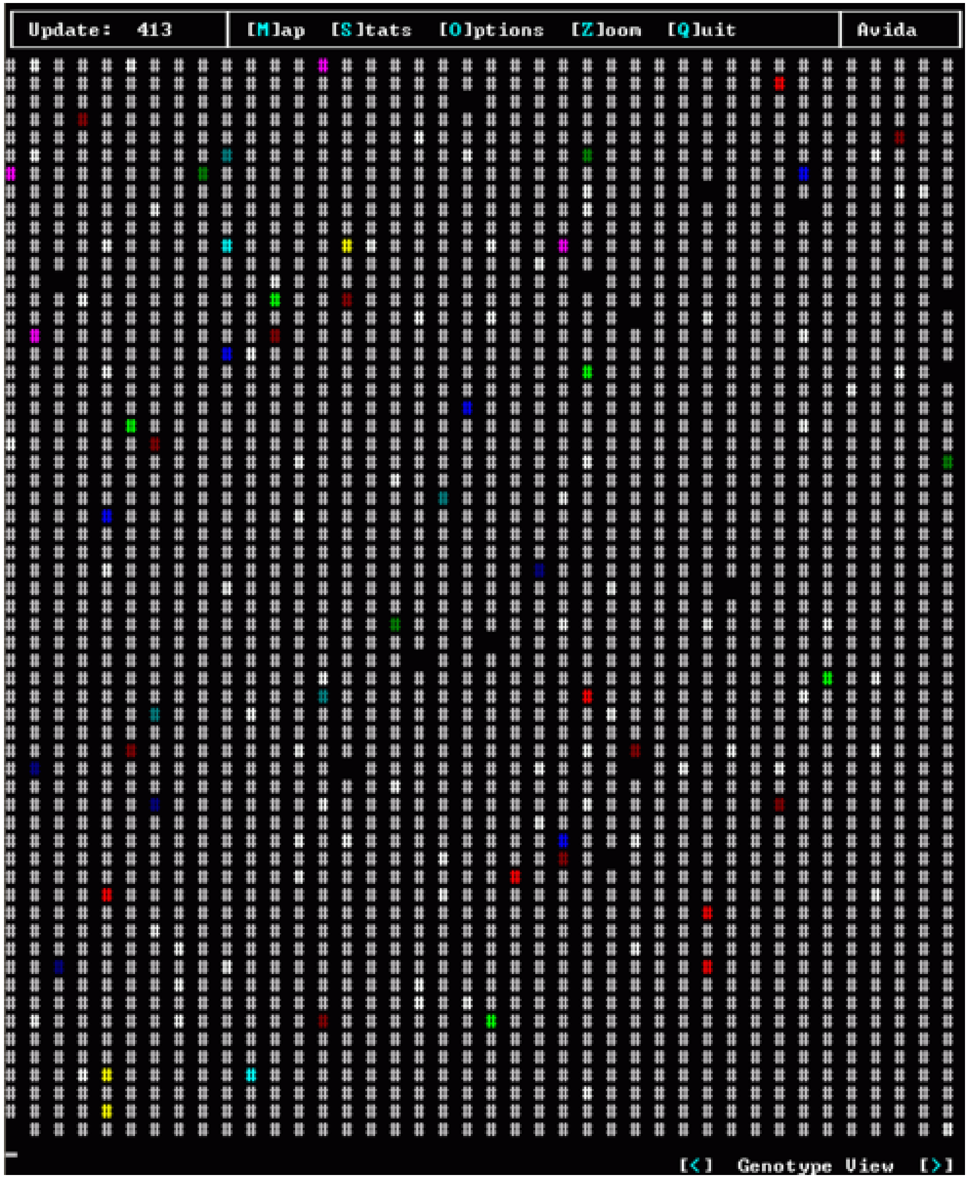
COSC 4560 / COSC 5560

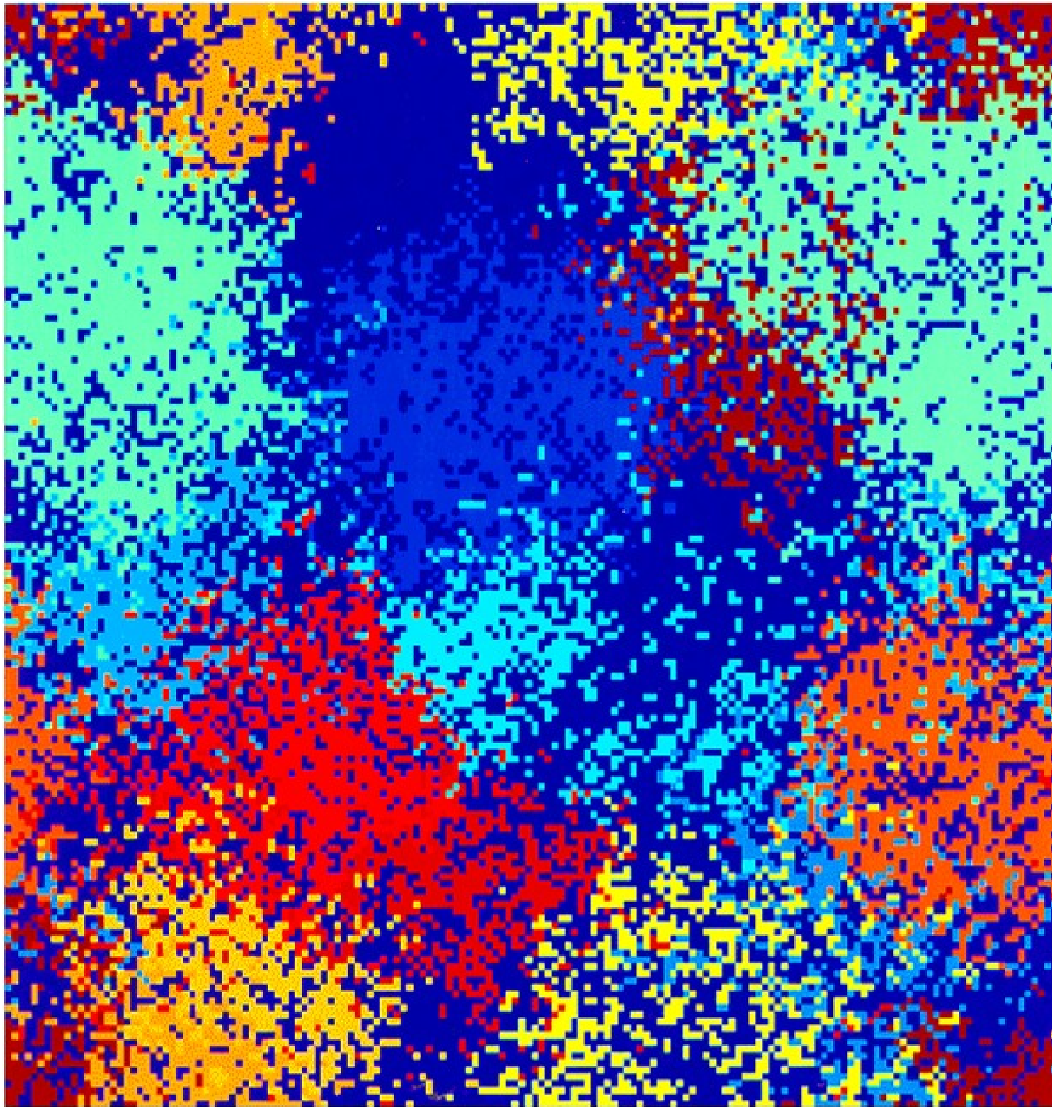
Professor Cheney

5/2/18

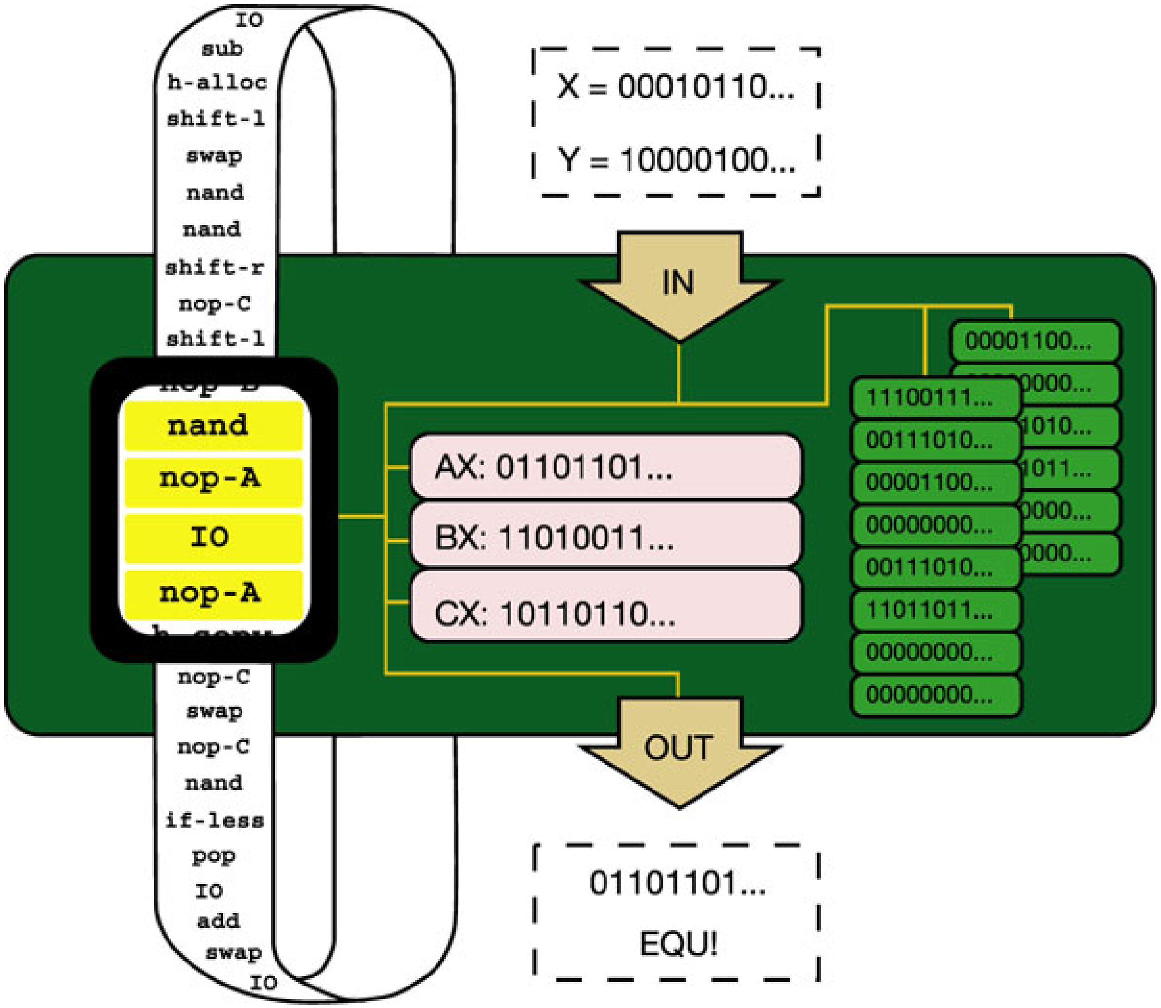
Studying Biology with Evolutionary Computation

Computer (Virus) Evolution:
Avida (Ofria, Adami, Brown '93)





Computer (Virus) Evolution: Avida (Ofria, Adami, Brown '93)



Computer (Virus) Evolution: Avida (Ofria, Adami, Brown '93)

(a) **nop-A** No-operation instruction; modifies other instructions

(b) **nop-B** No-operation instruction; modifies other instructions

(c) **nop-C** No-operation instruction; modifies other instructions

(d) **if-n-equ** Test if two registers contain equal values

(e) **if-less** Test if one register contains a lesser value than another

(f) **pop** Remove a number from a stack and place it in a register

(g) **push** Copy the value of a register onto the top of a stack

(h) **swap-stk** Toggle the active stack

(i) **swap** Swap the contents of two specified registers

(j) **shift-r** Shift all the bits on a register one to the right

(k) **shift-l** Shift all the bits on a register one to the left

(l) **inc** Increment a register

(m) **dec** Decrement a register

(n) **add** Calculate the sum of the values in two registers

(o) **sub** Calculate the difference between the values in two registers

(p) **nand** Perform a bitwise NAND on the values in two registers

(q) **IO** Output the value in a register and replace with a new input

(r) **h-alloc** Allocate memory for an offspring

(s) **h-divide** Divide off an offspring contained in memory (specified by heads)

(t) **h-copy** Make a copy of a single instruction in memory (specified by heads)

(u) **h-search** Find a pattern of nop-instruction in the genome

(v) **mov-head** Move a head to point to the same position as the flow-head

(w) **jmp-head** Move a head by a fixed amount stored in a register2

(x) **get-head** Write the position of a specified head into a register

(y) **if-label** Test if a specified pattern of nops has recently been copied

(z) **set-flow** Move the flow-head to a specified position in memory

Introduction to Avida-ED Digital Evolution Software

by
Wendy Johnson

Michigan State University
BEACON Center for Evolution in Action
Lansing Catholic High School
wjohnson@lansingcatholic.org

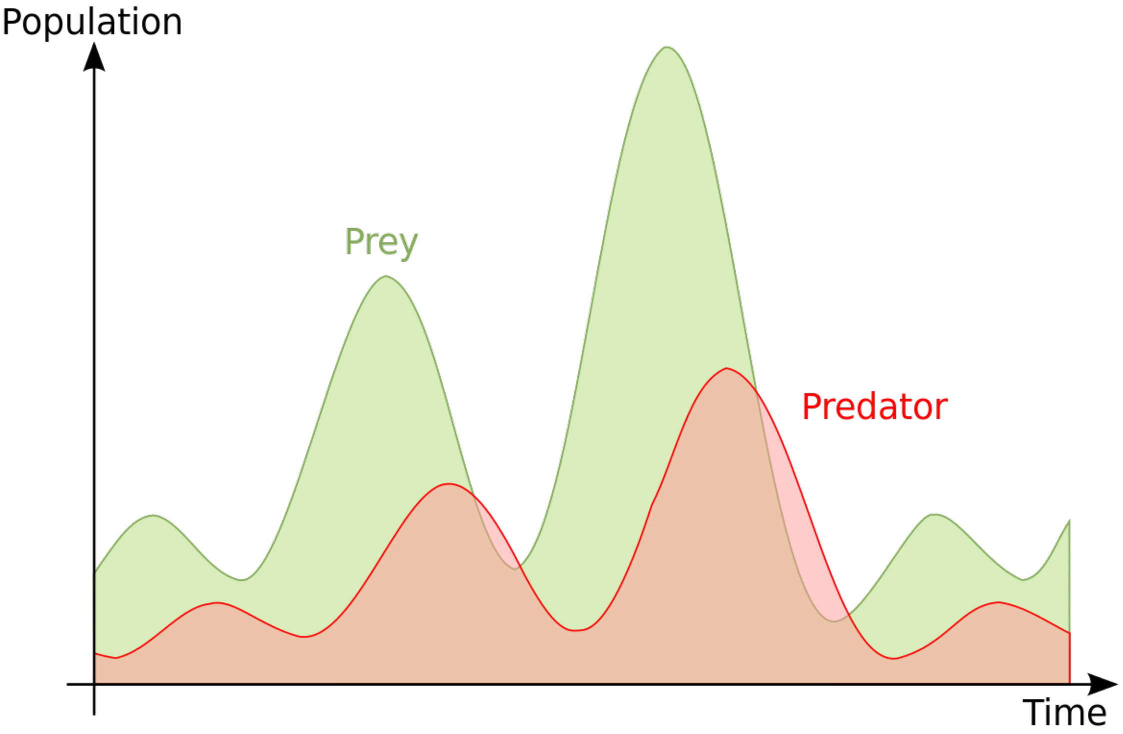
Lotka–Volterra equations

$$\frac{dx}{dt} = \alpha x - \beta xy,$$
$$\frac{dy}{dt} = \delta xy - \gamma y,$$

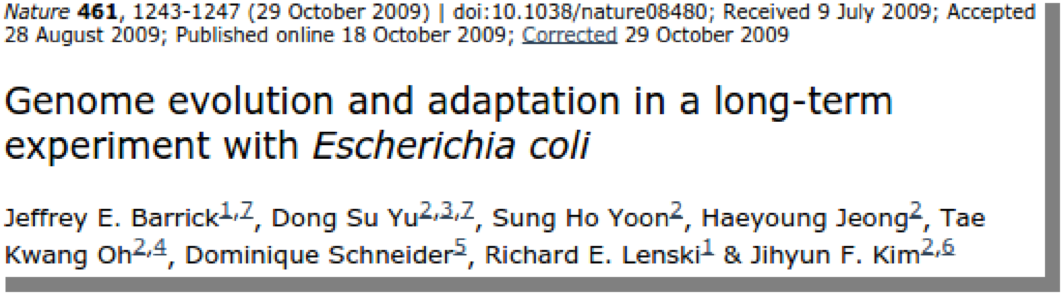
where

x is the number of prey (for example, rabbits);

y is the number of some predator (for example, foxes);



Computer (Virus) Evolution:
Avida (Ofria, Adami, Brown '93)



Nature **461**, 1243-1247 (29 October 2009) | doi:10.1038/nature08480; Received 9 July 2009; Accepted 28 August 2009; Published online 18 October 2009; Corrected 29 October 2009

Genome evolution and adaptation in a long-term experiment with *Escherichia coli*

Jeffrey E. Barrick[1,7], Dong Su Yu[2,3,7], Sung Ho Yoon[2], Haeyoung Jeong[2], Tae Kwang Oh[2,4], Dominique Schneider[5], Richard E. Lenski[1] & Jihyun F. Kim[2,6]

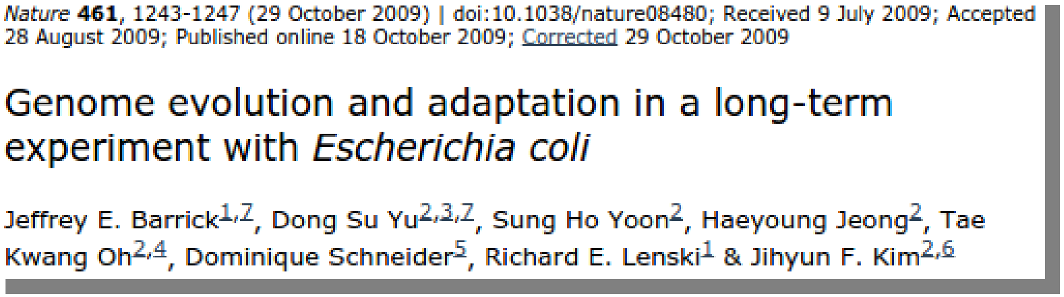
Every day, the cultures are propagated;

Every 75 days (500 generations),
mixed-population samples are frozen away;

Mean fitness, relative to the ancestor,
is estimated using the mixed-population samples.

Now 60,000+ generations!
(since 1988)

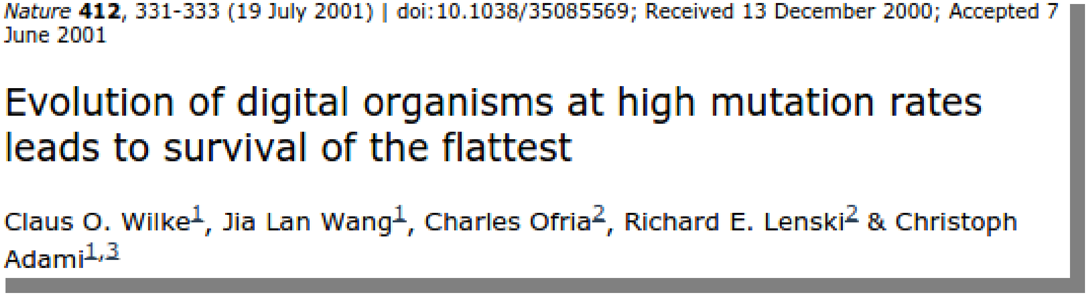
Computer (Virus) Evolution:
Avida (Ofria, Adami, Brown '93)



Nature **461**, 1243-1247 (29 October 2009) | doi:10.1038/nature08480; Received 9 July 2009; Accepted 28 August 2009; Published online 18 October 2009; Corrected 29 October 2009

Genome evolution and adaptation in a long-term experiment with *Escherichia coli*

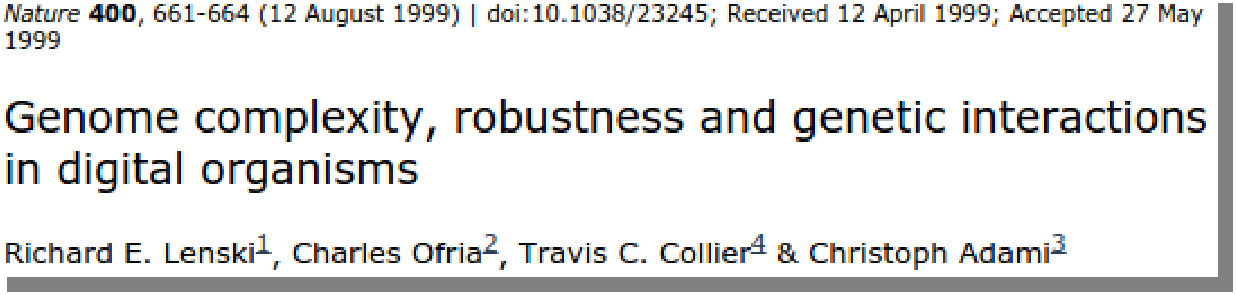
Jeffrey E. Barrick[1,7], Dong Su Yu[2,3,7], Sung Ho Yoon[2], Haeyoung Jeong[2], Tae Kwang Oh[2,4], Dominique Schneider[5], Richard E. Lenski[1] & Jihyun F. Kim[2,6]



Nature **412**, 331-333 (19 July 2001) | doi:10.1038/35085569; Received 13 December 2000; Accepted 7 June 2001

Evolution of digital organisms at high mutation rates leads to survival of the flattest

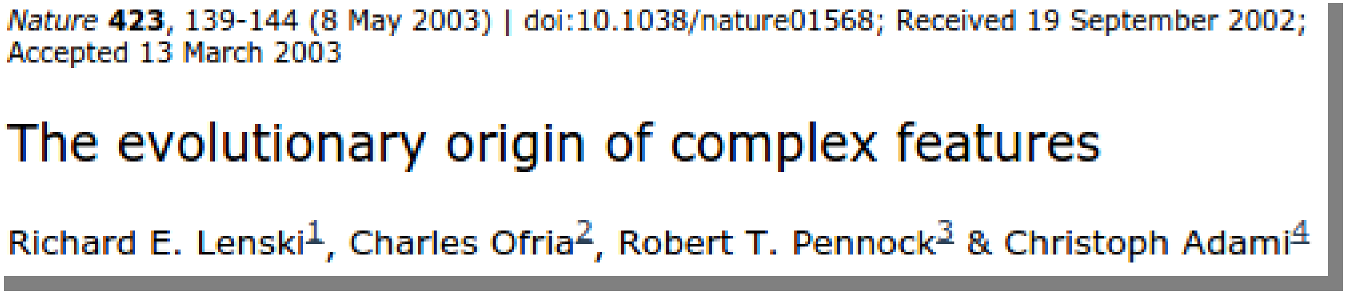
Claus O. Wilke[1], Jia Lan Wang[1], Charles Ofria[2], Richard E. Lenski[2] & Christoph Adami[1,3]



Nature **400**, 661-664 (12 August 1999) | doi:10.1038/23245; Received 12 April 1999; Accepted 27 May 1999

Genome complexity, robustness and genetic interactions in digital organisms

Richard E. Lenski[1], Charles Ofria[2], Travis C. Collier[4] & Christoph Adami[3]

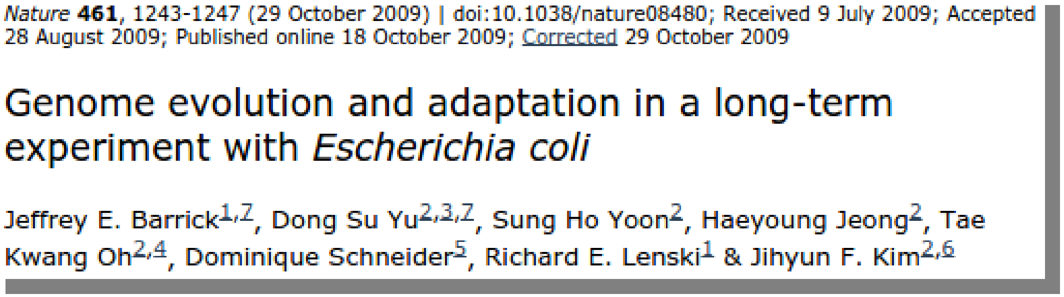


Nature **423**, 139-144 (8 May 2003) | doi:10.1038/nature01568; Received 19 September 2002; Accepted 13 March 2003

The evolutionary origin of complex features

Richard E. Lenski[1], Charles Ofria[2], Robert T. Pennock[3] & Christoph Adami[4]

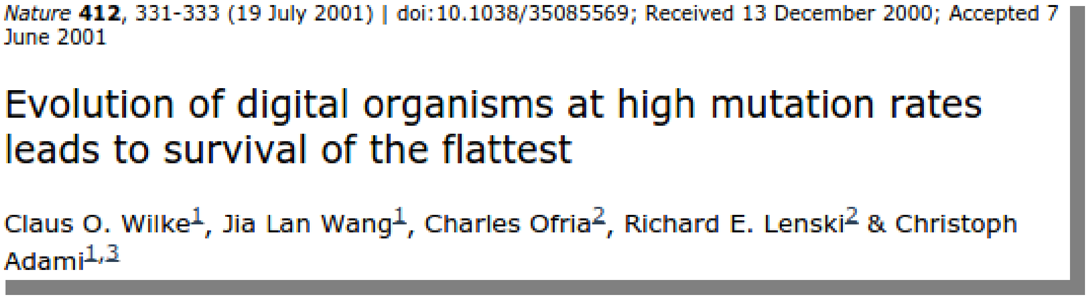
Computer (Virus) Evolution: **Avida** (Ofria, Adami, Brown '93)



Nature **461**, 1243-1247 (29 October 2009) | doi:10.1038/nature08480; Received 9 July 2009; Accepted 28 August 2009; Published online 18 October 2009; Corrected 29 October 2009

Genome evolution and adaptation in a long-term experiment with *Escherichia coli*

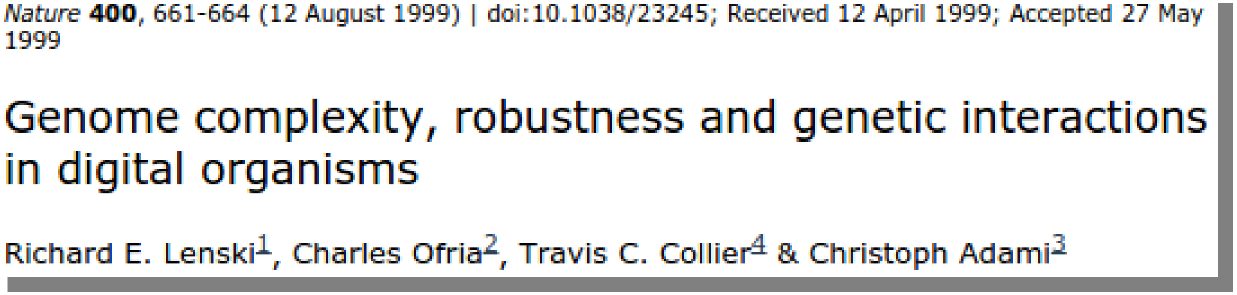
Jeffrey E. Barrick[1,7], Dong Su Yu[2,3,7], Sung Ho Yoon[2], Haeyoung Jeong[2], Tae Kwang Oh[2,4], Dominique Schneider[5], Richard E. Lenski[1] & Jihyun F. Kim[2,6]



Nature **412**, 331-333 (19 July 2001) | doi:10.1038/35085569; Received 13 December 2000; Accepted 7 June 2001

Evolution of digital organisms at high mutation rates leads to survival of the flattest

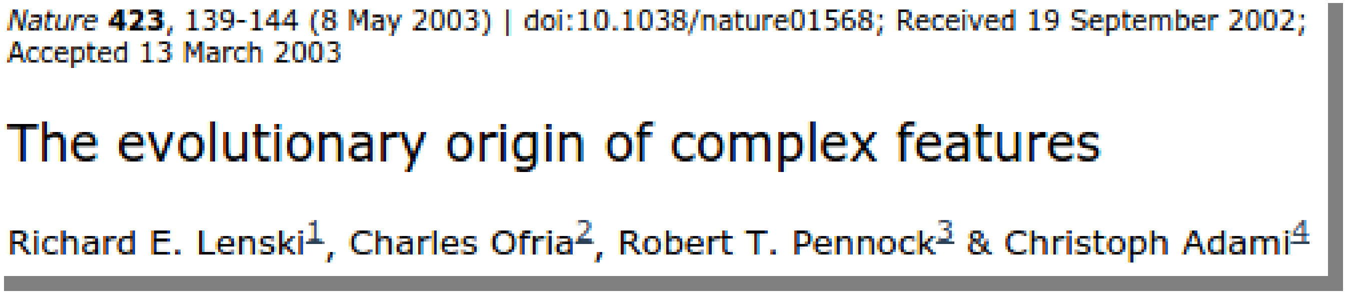
Claus O. Wilke[1], Jia Lan Wang[1], Charles Ofria[2], Richard E. Lenski[2] & Christoph Adami[1,3]



Nature **400**, 661-664 (12 August 1999) | doi:10.1038/23245; Received 12 April 1999; Accepted 27 May 1999

Genome complexity, robustness and genetic interactions in digital organisms

Richard E. Lenski[1], Charles Ofria[2], Travis C. Collier[4] & Christoph Adami[3]



Nature **423**, 139-144 (8 May 2003) | doi:10.1038/nature01568; Received 19 September 2002; Accepted 13 March 2003

The evolutionary origin of complex features

Richard E. Lenski[1], Charles Ofria[2], Robert T. Pennock[3] & Christoph Adami[4]

Lenski Publications:

Nature (24)

Science (18)

PNAS (18)

**Including Letters/ Reviews

The evolutionary origin of complex features

Richard E. Lenski*, Charles Ofria†, Robert T. Pennock‡ & Christoph Adami§

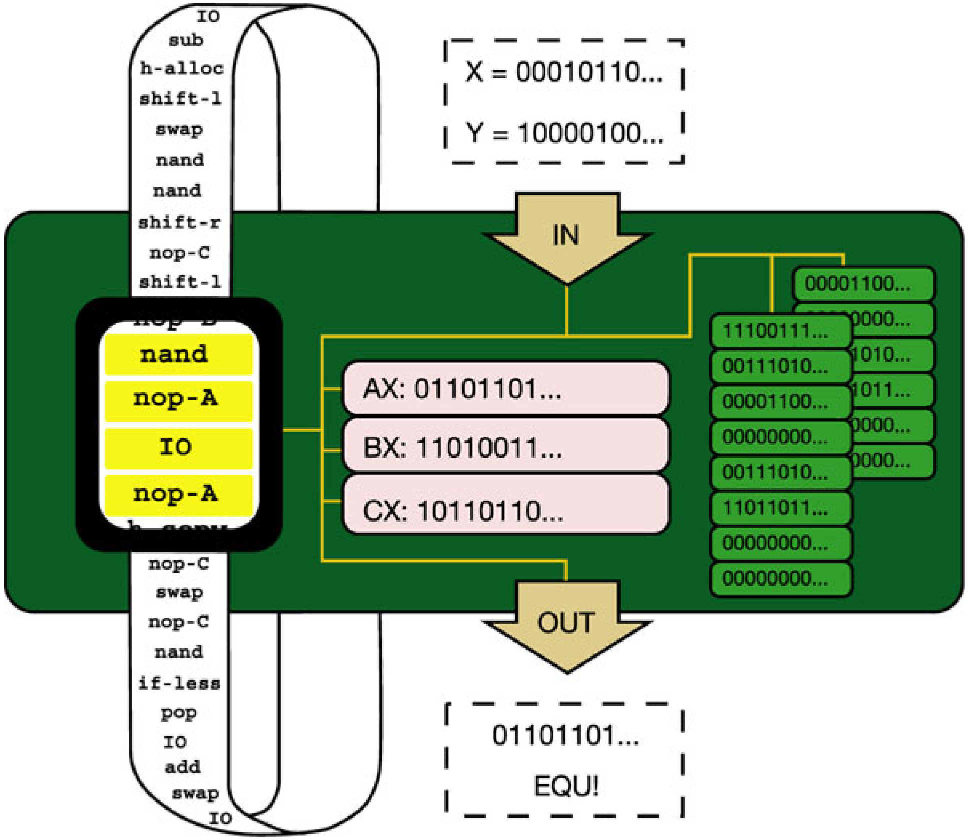
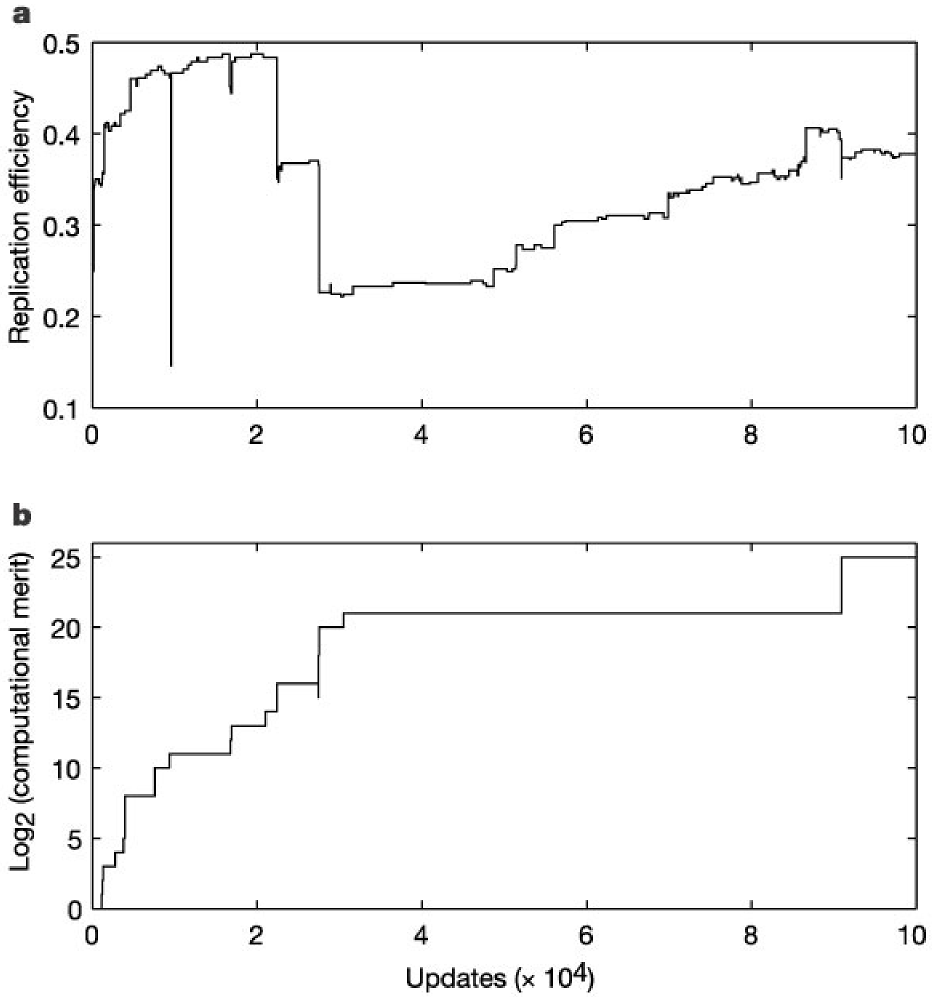


Table 1 **Rewards for performing nine one- and two-input logic functions**

Function name	Logic operation	Computational merit
NOT	~A; ~B	2
NAND	~(A and B)	2
AND	A and B	4
OR_N	(A or ~B); (~A or B)	4
OR	A or B	8
AND_N	(A and ~B); (~A and B)	8
NOR	~A and ~B	16
XOR	(A and ~B) or (~A and B)	16
EQU	(A and B) or (~A and ~B)	32

The symbol '~' denotes negation. The reward for computational merit increases with 2^n, where *n* is the minimum number of nand operations needed to perform the listed function. Symmetrical operations, shown separated by a semi-colon, are treated as the same function. No added benefit is obtained for performing any function multiple times. These functions include all one- and two-input logic operations except ECHO, which requires no nand operations and was not rewarded.



Instruction	Repl. 🟩	NOT 🟩	NAND 🟥	AND 🟥	OR_N 🟩	OR 🟩	AND_N 🟩	NOR 🟩	XOR 🟥	EQU 🟩
1 r *h-alloc*	🟥									
2 m *dec*						🟥	🟥	🟥		🟥
3 z *set-flow*	🟥									
4 a *nop-A*	🟥									
5 v *mov-head*	🟥									
6 c *nop-C*	🟥	🟥			🟥	🟥	🟥	🟥		🟥
7 g *push*										
8 m *dec*										
9 c *nop-C*							🟥			
10 l *swap*										
11 q *IO*										🟥
12 q *IO*										🟥
13 p *nand*		🟥								
14 t *h-copy*										🟥
15 q *IO*		🟥		🟩		🟥		🟥		🟥
16 p *nand*				🟩		🟥		🟥		🟥
17 q *IO*				🟩		🟥	🟥	🟥		🟥
18 c *nop-C*						🟥	🟥	🟥		🟥
19 p *nand*						🟥	🟥	🟥		🟥
20 c *nop-C*				🟩		🟥	🟥	🟥		🟥
21 t *h-copy*										🟥
22 l *inc*						🟥		🟥		🟥
23 e *if-less*										
24 t *h-copy*										🟥
25 n *add*						🟥	🟥	🟥		🟥
26 c *nop-C*						🟥	🟥	🟥		🟥
27 o *sub*							🟥			
28 g *push*				🟩		🟥				🟥
29 c *nop-C*						🟥				🟥
30 b *nop-B*										🟥
31 e *if-less*										
32 a *nop-A*										
33 m *dec*							🟥			
34 q *IO*					🟥		🟥			🟥
35 d *if-n-equ*										
36 t *h-copy*										🟥
37 q *IO*					🟥	🟥		🟥		🟥
38 c *nop-C*					🟥			🟥		🟥
39 p *nand*			🟩		🟥	🟥				🟥
40 t *h-copy*										🟥
41 l *swap*						🟥				🟥
42 p *nand*					🟥					
43 q *IO*					🟥					🟥
44 f *pop*						🟥				🟥
45 p *nand*						🟥				🟥
46 g *push*				🟩						🟥
47 q *IO*						🟥				🟥
48 x *get-head*										
49 u *h-search*	🟥		🟩	🟩						🟥
50 t *h-copy*										
51 y *if-label*										
52 c *nop-C*										
53 u *h-search*										🟥
54 a *nop-A*										🟥
55 s *h-divide*	🟥									
56 t *h-copy*										
57 t *h-copy*										
58 t *h-copy*										
59 v *mov-head*	🟥		🟩							🟥
60 a *nop-A*										
State changes	🟥 8	🟥 3	🟩 3	🟩 7	🟥 7	🟥 19	🟥 12	🟥 13	🟩 0	🟥 35

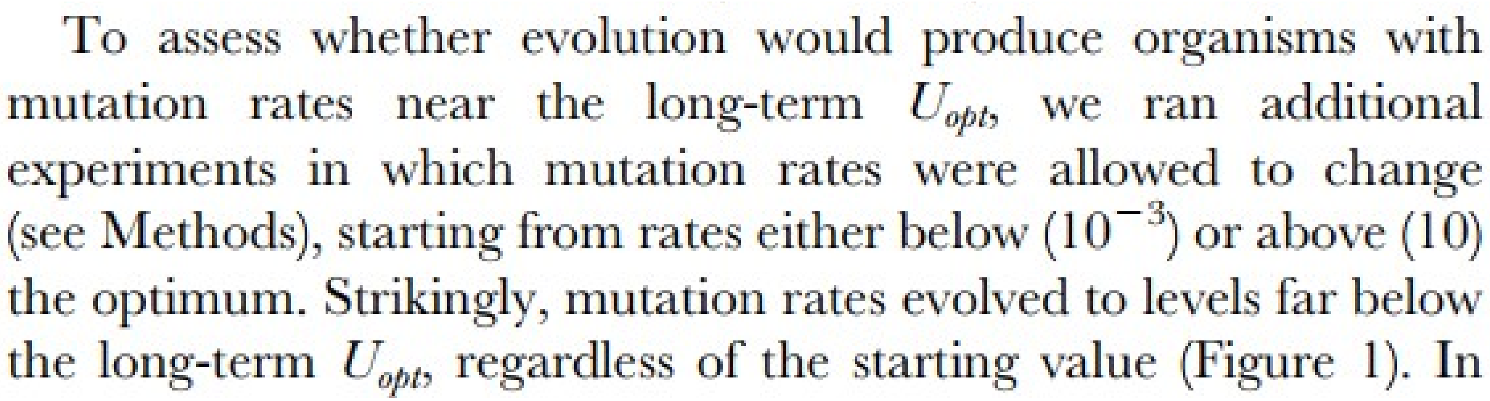
Figure 4 Functional-genomic array for the first organism to perform EQU in the case-study population. Its genome sequence is shown to the left; the instruction highlighted in yellow is the pivotal mutation that yielded EQU but simultaneously eliminated AND. Top labels denote replication (Repl.) and logic functions; associated colours show whether this organism can (green) or cannot (red) perform the function. The fill in each interior cell shows the effect on the function of replacing the instruction with a null instruction. Red, null mutation destroys existing function; blank, null mutation has no qualitative effect; green, null mutation produces new function. The number of state changes for each function is shown at the bottom.

Constraints of Evolutionary Processes

Natural Selection Fails to Optimize Mutation Rates for Long-Term Adaptation on Rugged Fitness Landscapes

Jeff Clune[1,2]*, Dusan Misevic[3], Charles Ofria[1], Richard E. Lenski[4], Santiago F. Elena[2,5], Rafael Sanjuán[2,6]

We studied the evolution of mutation rates using the Avida digital evolution platform [25–34]. To test empirically whether there was an intermediate, optimal rate of mutation that maximized adaptation, we performed a series of evolution experiments. In each experiment, a genetically homogenous population was placed in a novel environment where it evolved for 150,000 updates (~15,000 generations) at a constant mutation rate (see Methods). We explored 15 different mutation rates spanning six orders of magnitude (10^{-5} to 10 mutations per genome per generation). The final fitness values confirmed that there was an optimal mutation rate at an intermediate value, with $U_{opt} \approx 4.641$ (Figure 1). An analysis of the temporal dynamics of these experiments showed that this rate yielded the highest fitness from about generation 230 onward. Interestingly, for the very earliest time points (before generation 50), the lowest mutation rate (10^{-5}) produced the highest fitness values, whereas for generations 50–230 a mutation rate of 2.2 gave the highest fitness values.



To assess whether evolution would produce organisms with mutation rates near the long-term U_{opt}, we ran additional experiments in which mutation rates were allowed to change (see Methods), starting from rates either below (10^{-3}) or above (10) the optimum. Strikingly, mutation rates evolved to levels far below the long-term U_{opt}, regardless of the starting value (Figure 1). In

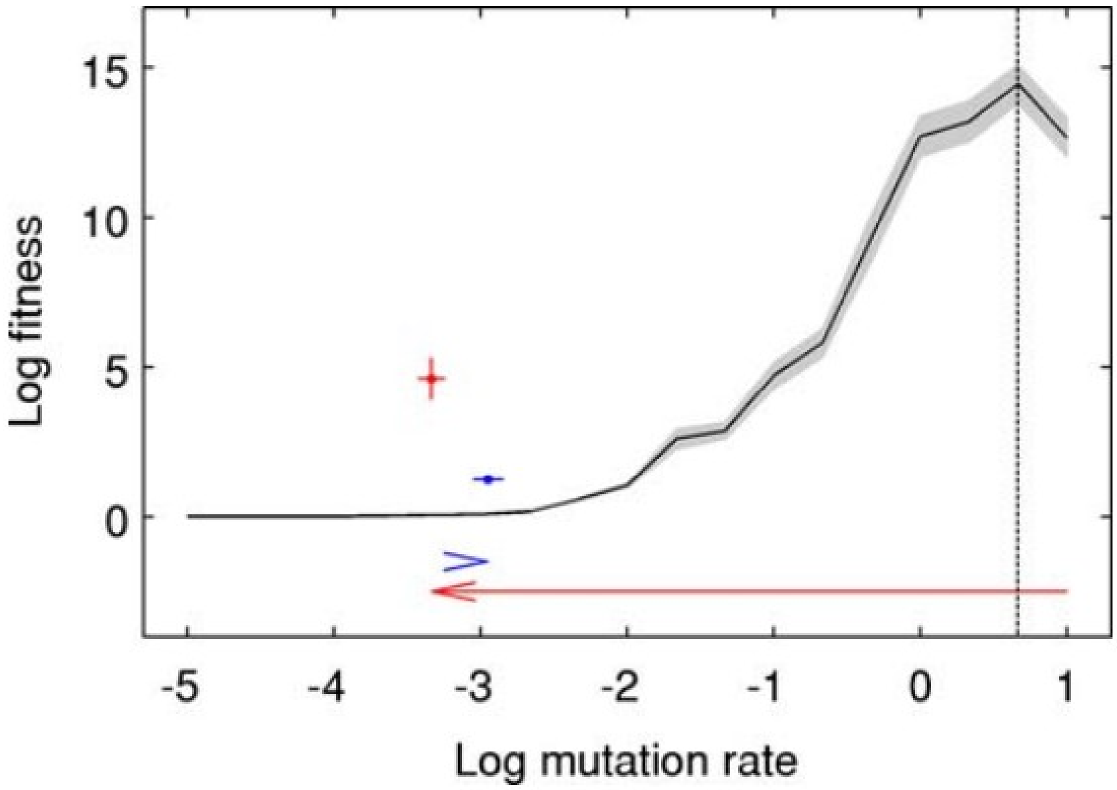


Figure 1. Evolution of suboptimal mutation rates on a complex fitness landscape. Fitness is shown as a function of the genomic mutation rate. The solid line shows mean fitness of the final population, itself averaged over 50 runs, for 15 different static mutation rates ($U = 10^{-5}$, 10^{-4} and from 10^{-3} to 10 at 1/3 $\log_{10}$ intervals). The shaded area represents ±1 s.e.m. The optimal mutation rate—the rate that maximized final fitness—was $U_{opt} \approx 4.641$ (vertical dashed line). The two colored points show the mean fitness and mutation rate of the final population, averaged over 50 runs, in experiments where mutation rates freely evolved with starting values of either 10 (red) or 10^{-3} (blue) (error bars represent ±1 s.e.m). Evolved mutation rates and fitness values were both orders of magnitude lower than those observed in the experiment with U_{opt}.

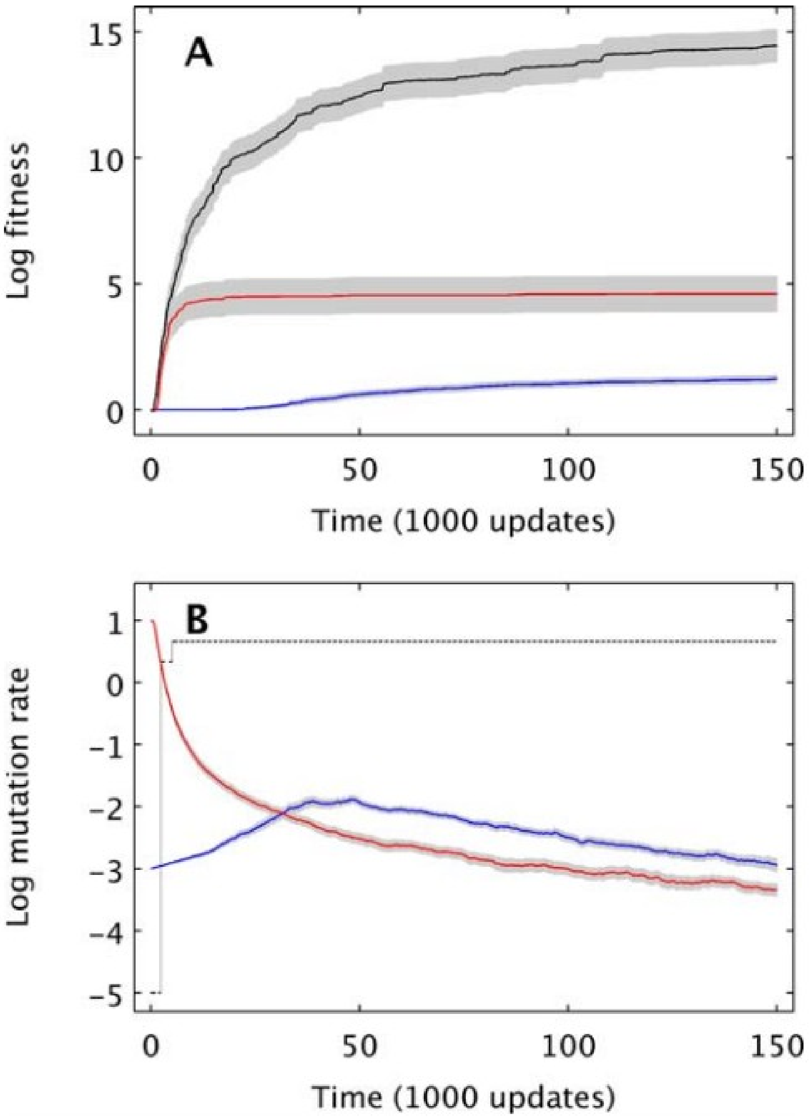


Figure 2. Evolutionary trajectories for fitness and mutation rate on a complex fitness landscape. (A) Evolution of average log-fitness±1 s.e.m. for treatments with the mutation rate fixed at $U_{opt} = 4.641$ (black) and for treatments with variable mutation rates starting at either 10 (red) or 10^{-3} (blue). (B) Evolution of average log genomic mutation rate±1 s.e.m. for treatments with variable mutation rates starting at either 10 (red) or 10^{-3} (blue). The black line indicates the mutation rate that had produced the highest average fitness for that time point.

We conclude from the results presented thus far that the failure of the evolving populations to achieve or even maintain the mutation rates that maximize long-term adaptation reflect the conflict between the short-term cost of deleterious mutations and the long-term potential for adaptive evolution. We further hypothesize that the resolution of this tension may depend on the topology of the fitness landscape on which evolution occurs. In a rugged fitness landscape, where there are multiple peaks separated by maladaptive valleys [35,36], populations at a local optimum must traverse regions of low fitness in the short-term in order to reach higher-fitness solutions in the long-term. This conflict leads us to hypothesize that the inability of natural selection to optimize mutation rates may depend on the ruggedness of the fitness landscape. The ideal test of this hypothesis requires comparing the evolution of mutation rates on fitness landscapes with and without fitness valleys. This test cannot be performed using the standard Avida setup, owing to the presence of extensive genetic interactions that make the fitness landscape complex and rugged [23]. We therefore modified Avida to allow simple, explicit, user-defined fitness functions that allowed us to manipulate the ruggedness of the fitness landscape (Methods, Figure 3). Adaptation occurs so fast when using these simple configurations that we also had to make the environment fluctuate between two 'seasons' in order to ensure a continual opportunity for beneficial mutations. These fluctuations mean that genotypes that are more fit in one season are less fit in the other (Figure 3).

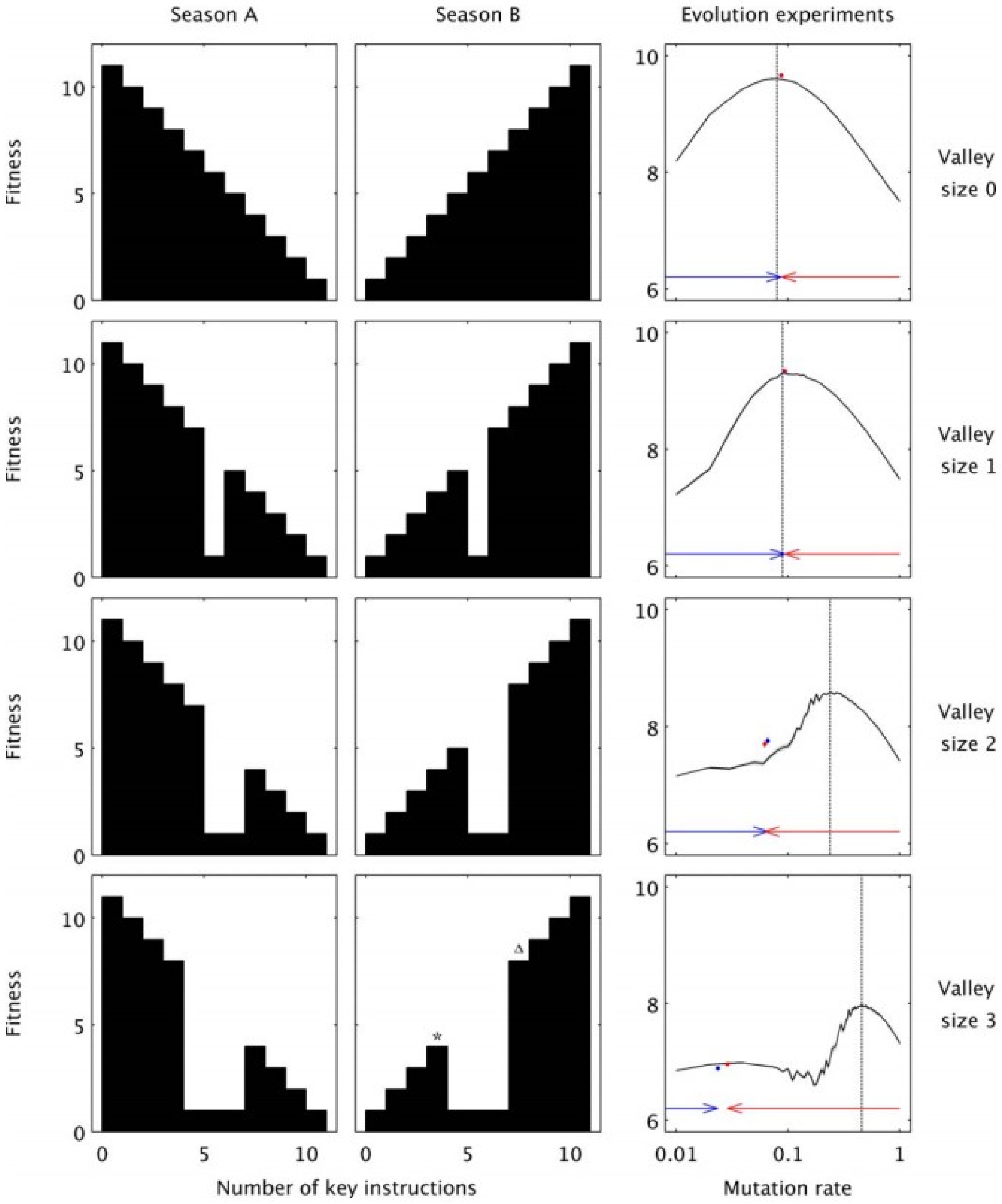


Figure 3. Evolution of mutation rates on simple fitness landscapes with different ruggedness. Here, fitness depended solely on the match between the environment and the number of a key instruction that organisms had in their genomes. In season A (left column) the key instruction was deleterious while it was beneficial in season B (center column). Rugged fitness landscapes with maladaptive valleys (rows 2–4) were introduced by setting the fitness of organisms with intermediate numbers of the key instruction to the minimum fitness level of one. The right-most column shows the results of evolution experiments under each of these selective regimes. Final fitness is shown as a function of genomic mutation rate for both static and dynamic mutation rates. The solid black line represents the average of the mean fitness across 10 runs for each of 100 different static mutation rates ranging from $U = 0.01$ to 1 in increments of 0.01. The two colored points represent the mean fitness and mutation rate,

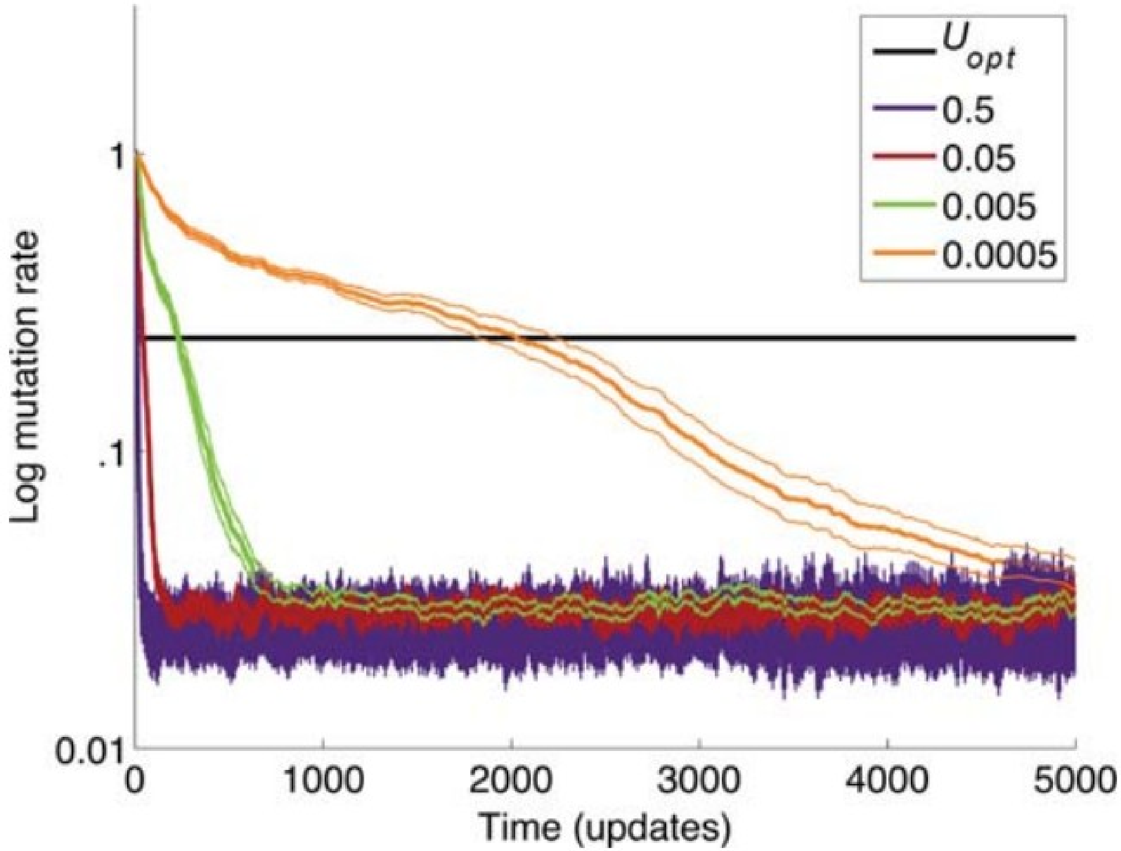


Figure 4. Evolutionarily stable mutation rate does not depend on the frequency with which the mutation rate changes (Π). The evolution of mutation rates in the explicit fitness landscape with a valley size of three is shown for several values of Π, as indicated by the colored key. Each curve shows the average of 20 runs; the adjacent bands represent ±1 s.e.m. The value of U_{opt} was determined in previous experiments (see text). The rate of approach toward the evolutionarily stable mutation rate depends on Π, but the equilibrium value itself does not.

Author Summary

Natural selection is shortsighted and therefore does not necessarily drive populations toward improved long-term performance. Some traits may evolve because they provide immediate gains, even though they are less successful in the long run than some alternatives. Here, we use digital organisms to analyze the ability of evolving populations to optimize their mutation rate, a fundamental evolutionary parameter. We show that when the mutation rate is constrained to be high, populations adapt considerably faster over the long term than when the mutation rate is allowed to evolve. By varying the fitness landscape, we show that natural selection tends to reduce the mutation rate on rugged landscapes (but not on smooth ones) so as to avoid the production of harmful mutations, even though this short-term benefit limits adaptation over the long term.

CONGRATS – you made it!!!

Thanks!

Questions?