Supplementary Information for *Nature* article by Lenski, Ofria, Pennock & Adami on "The Evolutionary Origin of Complex Features" (2003)

The first three sections below provide some background information on Avida and the logic functions that digital organisms perform. The fourth section provides some additional data on phenotypic and genomic evolution along the line of descent in the case-study population, up to the time of origin of the EQU function.

The Avida program and configuration files used in our experiments can be obtained free at **myxo.css.msu.edu/papers/nature2003/**. One can also view more background information about Avida as well as additional data from our experiments.

I. Avida instruction set

Each position in the genome sequence of a digital organism is one of 26 possible instructions. The table below shows the alphabetical code for each instruction, its mnemonic, and a brief description of its function.

(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 register

(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

II. One- and two-input logic operations

There are two distinct one-input logic operations and eight distinct two-input operations. The tables below show the output obtained by performing these operations on each possible bit-wise input. The bottom rows indicate the minimum number of nand computations required to perform each operation, proven through exhaustive search. To receive the reward for doing a particular logic function, a digital organism must return the correct values for an entire series of 32 bit-wise problems, including multiple examples of each possible combination of inputs.

1-Input	Output	
A	**ECHO**	**NOT**
0	0	1
1	1	0
# nand	**0**	**1**

2-Input		Output							
A	**B**	**NAND**	**AND**	**OR_N***	**OR**	**AND_N***	**NOR**	**XOR**	**EQU**
0	0	1	0	1	0	0	1	0	1
0	1	1	0	0	1	0	0	1	0
1	0	1	0	1	1	1	0	1	0
1	1	0	1	1	1	0	0	0	1
# nand		**1**	**2**	**2**	**3**	**3**	**4**	**4**	**5**

***** The order of the two inputs is arbitrary. Thus, the rewards for performing OR_N and AND_N are triggered by the reciprocal operations, in which A and B are reversed.

III. Shortest hand-written EQU program

The following hand-written program appears to be the shortest one using the set of available instructions that performs the EQU function, and that does not depend on the initial content of stacks and registers (whose initial contents are represented by '?' below). However, it has not been proven that this is the shortest program. Also, this program does not encode self-replication, but only performs the EQU function.

EQU						
#	**Inst**	**AX**	**BX**	**CX**	**Stack**	**Output**
1	**IO**	?	X	?	?	?
2	**IO**	?	X	Y	?	?

3	**nop-C**					
4	**push**	?	X	Y	X, ?	
5	**nand**	?	X nand Y	Y	X, ?	
6	**swap**	?	Y	X nand Y	X, ?	
7	**nand**	?	X or ~Y	X nand Y	X, ?	
8	**swap**	X or ~Y	?	X nand Y	X, ?	
9	**nop-A**					
10	**pop**	X or ~Y	X	X nand Y	?	
11	**nand**	X or ~Y	Y or ~X	X nand Y	?	
12	**swap**	X nand Y	Y or ~X	X or ~Y	?	
13	**nop-C**					
14	**nand**	X nand Y	X xor Y	X or ~Y	?	
15	**push**	X nand Y	X xor Y	X or ~Y	X xor Y, ?	
16	**pop**	X nand Y	X xor Y	X xor Y	?	
17	**nop-C**					
18	**nand**	X nand Y	X equ Y	X xor Y	?	
19	**IO**	X nand Y	Z	X xor Y	?	X equ Y

IV. Phenotypic and genomic evolution along the line of descent in the case-study population through the origin of the EQU function at step 111

The entire line of descent is available at **myxo.css.msu.edu/papers/nature2003/** along with functional-genomic arrays for all 345 genotypes in the line of descent.

Legend: PD = phylogenetic depth. Born = update when genotype first appeared. Functions 0 and 1 denote inability or ability, respectively, to perform nine logic functions: NOT, NAND, AND, OR_N, OR, AND_N, NOR, XOR, and EQU. Fit = fitness relative to immediate parent calculated as product of replication efficiency and computational merit; gains are shown in bold, losses in italics. Mutations are highlighted in bold in the genome sequences, as are changes in functions performed.

PD	Born	Functions	Fit	Genome Sequence
0	0	0 0 0 0 0 0 0 0 0 0	—	rucavccccccccccccccccccccccccccccccccccccutycasvab
1	32	0 0 0 0 0 0 0 0 0 0	1.00	rucavccccccccccccccccccccccccc**h**ccccccutycasvab
2	93	0 0 0 0 0 0 0 0 0 0	**1.01**	rucavccccccccccccccccccccccccc**p**ccchccccccutycasvab
3	143	0 0 0 0 0 0 0 0 0 0	**1.36**	rucavccccccccccccccccccccccccp_ccchccccccutycas**t**vab
4	225	0 0 0 0 0 0 0 0 0 0	**1.01**	rucavccccccccccccccccccccccccpc**n**chccccccutycastvab
5	284	0 0 0 0 0 0 0 0 0 0	**1.01**	rucavcc**z**ccccccccccccccccccccccpcnchcccc**q**cutycastvab
6	352	0 0 0 0 0 0 0 0 0 0	1.00	rucavczccccccccccccccccccccccc**q**cnchccccqcutycastvab
7	361	0 0 0 0 0 0 0 0 0 0	1.00	rucavc**o**zccccccccccccccccccccccqcnchccccqcutycastvab
8	379	0 0 0 0 0 0 0 0 0 0	**1.01**	rucavcozccccccccccccccccccc**x**ccqcnchccccqcutycastvab
9	411	0 0 0 0 0 0 0 0 0 0	1.00	rucavcozccccccccccccccccc**a**cxccqcnchccccqcutycastvab
10	567	0 0 0 0 0 0 0 0 0 0	1.00	rucavcozccccccccccccccccacx**e**cqcnchccccqcutycastvab

11	734	0	0	0	0	0	0	0	0	1.00	rucavcozccccccccccccccccam	xecqcnchccccqcutycastvab
12	775	0	0	0	0	0	0	0	0	1.00	rucavcozcccs	ccccccccccccccamxecqcnchccccqcutycastvab
13	793	0	0	0	0	0	0	0	0	*0.99*	rucavcozcccsci	cccccccccccccamxecqcnchccccqcutycastvab
14	835	0	0	0	0	0	0	0	0	1.00	rucavcozcccsciccccccccccccamxe	lqcnchccccqcutycastvab
15	949	0	0	0	0	0	0	0	0	1.00	rucavcoz	jccsciccccccccccccamxelqcnchccccqcutycastvab
16	963	0	0	0	0	0	0	0	0	*0.99*	rucavcozjccsciccccccccccccamxelqcn	qhccccqcutycastvab
17	1118	0	**1**	0	0	0	0	0	0	**2.01**	rucavcozjccsciccccccccccccamxelqcnqhcc	pcqcutycastvab
18	1194	0	1	0	0	0	0	0	0	**1.01**	rucavco	tzjccsciccccccccccccamxelqcnqhccpcqcutycastvab
19	1250	0	1	0	0	0	0	0	0	**1.01**	rucavcotzjcisci	ccccccccccccamxelqcnqhccpcqcutycastvab
20	1252	0	**0**	0	**1**	0	0	0	0	**2.01**	rucavcotzjcisciccccccccccccamxelqcnqh	pcpcqcutycastvab
21	1263	0	0	0	1	0	0	0	0	**1.00**	rucavcotzjcisciccc	ncccck	camq	elqcnqhpcpcqcutycastvab
22	1289	0	**1**	0	1	0	0	0	0	**2.00**	rucavcotzjcisciccc	ncccccckcamqelqc	pqhpcpcqcutycastvab
23	1325	0	1	0	1	0	0	0	0	1.00	rucavcotzjcisciccd	ncccccckcamqelqcpqhpcpcqcutycastvab
24	1433	0	1	0	1	0	0	0	0	*0.99*	rucavcotzjcisciccdnc	ecccckcamqelqcpqhpcpcqcutycastvab
25	1439	0	1	0	1	0	0	0	0	**1.15**	rucavcotzjcisciccdnceccckcamqelqcpqhpcpcqcutycast	tvab
26	1455	0	1	0	1	0	0	0	0	*0.99*	rucavcotzjcisciccdnceccck	ramqelqcpqhpcpcqcutycasttvab
27	1645	0	1	0	1	0	0	0	0	**1.01**	rucavcotzj	aisciccdnceccckramqelqcpqhpcpcqcutycasttvab
28	1910	0	1	0	1	0	0	0	0	1.00	rucavcotzjaisciccdnceccc	yramqelqcpqhpcpcqcutycasttvab
29	1947	0	1	0	1	0	0	0	0	*0.98*	rucavcotz	saisciccdnceccyramqelqcpqhpcpcqcutycasttvab
30	2286	0	1	0	1	0	0	0	0	**1.01**	rucavcotzsaisciccdnce	iccyramqelqcpqhpcpcqcutycasttvab
31	2479	0	1	0	1	0	0	0	0	1.00	rucavcotzsa	jsciccdnceiccyramqelqcpqhpcpcqcutycasttvab
32	2669	0	1	0	1	0	0	0	0	1.00	rucavcot	dsajsciccdnceiccyramqelqcpqhpcpcqcutycasttvab
33	2711	0	1	0	1	0	0	0	0	**1.01**	rucavcotd	qajsciccdnceiccyramqelqcpqhpcpcqcutycasttvab
34	2786	**1**	1	0	1	0	0	0	0	**2.00**	rucavcotdqajsciccdnc	oiccyramqelqcpqhpcpcqcutycasttvab
35	2828	1	1	0	1	0	0	0	0	1.00	rucavcotd	qaqsciccdncoiccyramqelqcpqhpcpcqcutycasttvab
36	2830	1	1	0	1	0	0	0	0	*0.99*	rucavcotdqaqsciccdnco	sccyramqelqcpqhpcpcqcutycasttvab
37	2987	1	1	0	1	0	0	0	0	1.00	rucavcotdqaqsciccdncosccyramqe	tqcpqhpcpcqcutycasttvab
38	3420	1	1	0	1	0	0	0	0	**1.03**	rucavcotdqaqs	qciccdncosccyramqetqcpqhpcpcqcutycasttvab
39	3551	1	1	0	1	0	0	0	0	1.00	rucavc	itdqaqsqciccdncosccyramqetqcpqhpcpcqcutycasttvab
40	3808	**0**	1	**1**	1	0	0	0	0	**2.00**	rucavcitdqaqsqc	pccdncosccyramqetqcpqhpcpcqcutycasttvab
41	3914	0	1	1	1	0	0	0	0	**1.01**	rucavcitdqaqsqcpccdncosccy	pamqetqcpqhpcpcqcutycasttvab
42	3939	0	1	1	1	0	**1**	0	0	0	**8.00**	rucavcitdqaqsqcpc	ldncosccypamqetqcpqhpcpcqcutycasttvab
43	4190	0	1	1	1	0	1	0	0	0	1.00	rucavc	otdqaqsqcpcldncosccypamqetqcpqhpcpcqcutycasttvab
44	4646	0	1	1	1	0	1	0	0	0	**1.08**	rucavcotdqaqsqcpcldncosccypamqetqcpqhpcpcqcutycastt	tvab
45	5292	0	1	1	1	0	1	0	0	0	1.00	rucavcotdqaqsqcpcldncosccypamq	rtqcpqhpcpcqcutycasttvab
46	5302	0	1	1	1	0	1	0	0	0	1.00	rucavcotdqaqsqcpcldncoscc	dpamqrtqcpqhpcpcqcutycasttvab
47	5323	0	1	1	1	0	1	0	0	0	*0.99*	ru	zavcotdqaqsqcpcldncosccdpamqrtqcpqhpcpcqcutycasttvab
48	5337	0	1	1	1	0	1	0	0	0	*0.99*	ruzavcotdqaqsqcpcldncosccdpamqrtqcpqhp	gpcqcutycasttvab
49	5481	0	1	1	1	0	1	0	0	0	**1.02**	ruzavcot	cqaqsqcpcldncosccdpamqrtqcpqhpgpcqcutycasttvab
50	6067	0	1	1	1	0	1	0	0	0	1.00	ruzavcot	iqaqsqcpcldncosccdpamqrtqcpqhpgpcqcutycasttvab
51	6200	0	1	1	1	0	1	0	0	0	1.00	ruzavcoti	zaqsqcpcldncosccdpamqrtqcpqhpgpcqcutycasttvab
52	6520	0	1	1	1	0	1	0	0	0	**1.01**	ruzavcotizaqsqcpcldncosccdpamq	btqcpqhpgpcqcutycasttvab
53	6765	0	1	1	1	0	1	0	0	0	1.00	r	jzavcotizaqsqcpcldncosccdpamqbtqcpqhpgpcqcutycasttvab
54	7117	0	1	1	1	0	1	0	0	0	**1.01**	rjzavcot	tizaqsqcpcldncosccdpamqbtqcpqhpgpcqcutycastttvab
55	7478	0	1	1	1	0	1	0	0	0	1.00	r	mzavcottizaqsqcpcldncosccdpamqbtqcpqhpgpcqcutycastttvab
56	7565	0	1	**0**	1	0	1	**1**	0	0	**4.00**	rmzavcottizaq	pqcpcldncosccdpamqbtqcpqhpgpcqcutycastttvab
57	7977	0	1	0	1	0	1	1	0	0	**1.01**	rmzavcottizaqpqcpcldncos	zcdpamqbtqcpqhpgpcqcutycastttvab
58	8443	0	1	0	1	0	1	1	0	0	*0.99*	rmzavcottizaqpqcpcldncoszcdpamq	gtqcpqhpgpcqcutycastttvab
59	8742	0	1	0	1	0	1	1	0	0	*0.99*	rmzavcottiz	pqpqcpcldncoszcdpamqgtqcpqhpgpcqcutycastttvab
60	8777	0	1	0	1	0	1	1	0	0	1.00	rmzavcot	htizpqpqcpcldncoszcdpamqgtqcpq_pgpcqcutycastttvab
61	9399	**1**	1	0	1	0	1	1	0	0	**1.98**	rmzavcoth	oiq	pqpqcpcldncosz	tdpamqgtqcpqpgpcqcutycastttvab
62	9428	1	1	0	1	0	1	1	0	0	1.00	rmzavcoth	hiqpqpqcpcldncosztdpamqgtqcpqpgpcqcutycastttvab
63	9582	1	1	0	1	0	1	1	0	0	*0.32*	rmzavcothhiqpqpqcpcldncosztdpamqgtqcpqp	sgpcqcutycastttvab
64	9590	1	1	0	1	0	1	1	0	0	**3.19**	rmzavcothhiqpqpqcpcl	tncosztdpamqgtqcpqpsgpcqcutycastttvab
65	9823	1	1	0	1	0	1	1	0	0	1.00	rmzavcot	ehiqpqpqcpcltncosztdpamqgtqcpqp	dgpcqcutycastttvab
66	9993	1	1	0	1	0	1	1	0	0	1.00	rmzavcotehiqpqpqcpcltncor	ztdpamqgtqcpqpdgpcqcutycastttvab
67	11029	1	1	0	1	0	1	1	0	0	**1.01**	rmzavcotehiqpqpqcpc	tltncorztdpamqgtqcpqpdgpcqcutycastttvab
68	11584	1	1	0	1	0	1	1	0	0	**1.01**	rmzavcotehiqp	tqpqcpctltncorztdpamqgtqcpqpdgpcqcutycastttvab
69	12028	1	1	0	1	0	1	1	0	0	**1.01**	rmzavcote	biqptqpqcpctltncorztdpamqgtqcpqpdgpcqcutycastttvab
70	12671	1	1	0	1	0	1	1	0	0	**1.01**	rmzavc	ytebiqptqpqcpctltncorztdpamq	btqcpqpdgpcqcutycastttvab
71	12853	1	1	0	1	0	1	1	0	0	1.00	rmzavcytebiqptqpqcpctltncor	ctdpamqbtqcpqpdgpcqcutycastttvab
72	12947	1	1	0	1	0	1	1	0	0	*0.99*	rmzavcytebiqptqpqcpctltncorctdpamq	ztqcpqpdgpcqcutycastttvab
73	13119	1	1	0	1	0	1	1	0	0	1.00	rmzavcyt	nbiqptqpqcpctltncorctdpamqztqcpqpdgpcqcutycastttvab
74	13537	1	1	0	1	0	1	1	0	0	1.00	rmzavc	ztnbiqptqpqcpctltncorctdpamqztqcpqpdgpcqcutycastttvab

75	13557	1	1	0	1	0	1	1	0	0	1.00	rmzavcztnbiqptqpqcpctltncorctdpamq**s**tqcpqpdgpcqcutycastttvab
76	13869	1	1	0	1	0	1	1	0	0	1.00	rmzavcztnbiqptqpqcpctltncorctdpamq**d**tqcpqpdgpcqcutycastttvab
77	13952	1	1	0	1	0	1	1	0	0	**1.01**	rmzavcztnbiqptqpqcpctltncorctdpamqdtqcpqpd**b**pcqcutycastttvab
78	14478	1	1	0	1	0	1	1	0	0	1.00	rmzavcztnbiqptqpqcpctltncorctdpamqdtqcpqp**c**bpcqcutycastttvab
79	15603	1	1	0	1	0	1	1	0	0	1.00	rmzavczt**w**biqptqpqcpctltncorctdpamqdtqcpqpcbpcqcutycastttvab
80	15769	1	1	0	1	0	1	1	0	0	**1.01**	rmzavcztwbiqptqpqcpctltnco**g**ctdpamqdtqcpqpcbpcqcutycastttvab
81	15936	1	1	0	1	0	1	1	0	0	1.00	rmzavcztwbiqptqpqcpctltncogct**b**pamqdtqcpqpcbpcqcutycastttvab
82	16039	1	1	0	1	0	1	1	0	0	1.00	rmzavcztwbiqptqpqcpctltncogctb**n**amqdtqcpqpcbpcqcutycastttvab
83	16090	1	1	0	1	0	1	1	0	0	1.00	rmzavcztwbiqptqpqcpctltncogctbnamqdtqcpqp**i**bpcqcutycastttvab
84	16672	1	1	0	1	0	1	1	0	0	*0.93*	rmzavcztwbiqptqpqcpctltncogctbnamqdtqcpqpibpcqcutycastt_vab
85	16769	1	1	0	**0**	**1**	1	1	0	0	**1.98**	rmzavcztwbiqptqpqcpctltncogctbnamqdtqcpqpi**f**pcqcutycasttvab
86	16786	1	1	0	0	1	1	1	0	0	*0.99*	rmzavcztwbiqptqpqcpctltncogctbnamqdtqcpqpifp**q**qcutycasttvab
87	16900	1	**0**	0	**1**	1	1	1	0	0	**2.00**	rmzavcztwbiqptqpqcpctltncogctbnamqdtqcp**e**pifpqqcutycasttvab
88	16979	1	0	0	1	1	1	1	0	0	**1.08**	rmzavcztwbiqptqpqcpctltncogctbnamqdtqcpepifpqqcutycastt**t**vab
89	17008	1	0	0	1	1	1	1	0	0	1.00	rmzavczt**g**biqptqpqcpctltncogctbnamqdtqcpepifpqqcutycastttvab
90	17145	1	0	0	1	1	1	1	0	0	1.00	rmzavc**j**tgbiqptqpqcpctltncogctbnamqdtqcpepifpqqcutycastttvab
91	17370	1	0	0	1	1	1	1	0	0	**1.01**	rmzavcjtgbiqptqpqcpctltncogctbnamqdtqcpepifpq**t**qcutycastttvab
92	19260	1	0	0	1	1	1	1	0	0	**1.01**	rmzavcjtgbiqptqpqcpctltncogctbnamqdtqcp**t**epifpqtqcutycastttvab
93	19828	1	0	0	1	1	1	1	0	0	1.00	rmzavcjtgbiqptqpqcpctltncogctbnamqdtqcpt**i**pifpqtqcutycastttvab
94	20740	1	0	0	1	1	1	1	0	0	*0.99*	rmzavcjtgbiqptqpqcpctltncogctbnamqdtqcptipifpq_qcutycastttvab
95	20804	1	0	0	1	1	1	1	0	0	1.00	rmzavcjtg**c**iqptqpqcpctltncogctbnamqdtqcptipifpqqcutycastttvab
96	21032	1	**1**	0	1	1	1	1	0	0	**2.00**	rmzavcjt**d**ciqptqpqcpctltncogctbnamqdtqcptip**q**fpqqcutycastttvab
97	21509	1	1	0	1	1	1	1	0	0	1.00	rmzavcjt**z**ciqptqpqcpctltncogctbnamqdtqcptipqfpqqcutycastttvab
98	22229	1	1	0	1	1	1	1	0	0	1.00	rmzavc**a**tzciqptqpqcpctltncogctbnamqdtqcptipqfpqqcutycastttvab
99	22240	1	1	0	1	1	1	1	0	0	1.00	rmzavc**g**tzciqptqpqcpctltncogctbnamqdtqcptipqfpqqcutycastttvab
100	22404	1	1	**1**	1	1	1	1	0	0	**2.90**	rmzavcgtzciqptqpqcpctltncogctbnamqdtqcptipqfpqqcutyc**u**astttvab
101	22412	1	1	1	1	1	1	1	0	0	1.00	rmzavcgtzciqptqpqcpctltncogctbnamqdtqcptipqfpqqcutycuastttva_
102	22487	1	1	1	1	1	1	1	0	0	1.00	rmzavcgt**m**ciqptqpqcpctltncogctbnamqdtqcptipqfpqqcutycuastttva
103	22586	1	1	1	1	1	1	1	0	0	*0.99*	rmzavcg_mciqptqpqcpctltncogctbnamqdtqcptipqfpqqcutycuastttva
104	22629	1	1	1	1	1	1	1	0	0	**1.05**	rmzavcgmciqptqpqcpctl**e**tncogctbnamqdtqcptipqfpqqcutycuastttva
105	22864	1	1	1	1	1	1	1	0	0	1.00	rmzavcgmciqptqpqcpctletncogctb**k**amqdtqcptipqfpqqcutycuastttva
106	22886	1	1	1	1	1	1	1	0	0	*0.99*	rmzavcgmciqptqpqcpctletncogc_bkamqdtqcptipqfpqqcutycuastttva
107	23002	1	1	1	1	1	1	1	0	0	**1.02**	rmzavcgmciq**q**ptqpqcpctletncogcbkamqdtqcptipqfpqqcutycuastttva
108	25881	1	1	1	1	1	1	1	0	0	1.00	rmzavcgmciqqptqpqcpctletncogcb**q**amqdtqcptipqfpqqcutycuastttva
109	26343	1	1	1	1	1	1	1	0	0	**1.01**	rmzavcgmciqqptqpqcpctletncogcb**e**amqdtqcptipqfpqqcutycuastttva
110	27437	1	**0**	1	1	1	1	1	0	0	*0.49*	rmzavcgmciqqptqpqcpctletncogcbeamqdtqcptipqfpqq**x**utycuastttva
111	27450	1	0	**0**	1	1	1	1	0	**1**	**8.00**	rmzavcgmciqqptqpqcpctletncogcbeamqdtqcptipqfp**g**qxutycuastttva