The evolution of modularity in genome architecture

John W. Pepper[1]

[1]Santa Fe Institute, Santa Fe, NM 87501

Abstract

It has been suggested that the capacity to evolve can itself evolve. One factor that may be important in determining evolvability is modularity of the genotype-phenotype map. A related phenomenon is modularity in genome architecture, in which genes with epistatic effects are co-localized on the chromosome. This changes a population's response to recombination by maximizing recombination among genes with additive effects and minimizing recombination of genes with epistatic interactions. The effects and origins of genome modularity were investigated using a computational model, in which a population of individuals evolved in a changing fitness landscape under mutation and recombination. Populations with modular genomes evolved faster than did populations with non-modular genomes. Moreover, in long runs that started with non-modular genomes but allowed inversion, populations evolved high levels of genome modularity. This demonstrates that over long time periods, evolvability can itself evolve through reorganization of the genome. The results may explain some observations of genome modularity in bacteria and other organisms.

Introduction

Evolvability can be defined as the capacity to acquire adaptation through the process of natural selection and analogous processes. The question I address here is whether evolvability can itself evolve through selection. In other words, in addition to the short-term process by which evolving entities acquire adaptation, is there a longer term process in which they increase their capacity to adapt?

A number of factors have been suggested that might affect evolvability, but one of the most general is the idea of modularity, or functional independence among multiple subunits of a larger entity. The key quality of modular organization is that changes in one subunit do not change how other subunits function. Modularity is important in the design of almost any complex functional entity, because it permits innovation and improvement in one subunit without losing functionality already achieved in other subunits. Adaptive evolution is often compared to a design process, and for the same reason it facilitates intelligent design, modularity may be crucial to evolvability.

At least two types of modularity may effect evolvability. One that has been discussed by several authors is modularity of the genotype-phenotype map. This refers to the situation in which different traits or functions are influenced by different sets of genes, so that they can evolve independently without epistatic interference (Wagner & Altenberg 1996, Ancel & Fontana 1999).

A second form of modularity, which I address here, is modularity in genome architecture. This kind of modularity is independent of the genotype-phenotype map, as it involves only the physical organization of genes on chromosomes. A genome's architecture is modular if sets of genes with epistatic effects on fitness are physically clustered on the chromosomes. Unlike modularity of the genotype-phenotype map, genotype modularity does not effect how a given mutation will alter the individual's phenotype or fitness. Its only significance is that it affects the response to crossover, because crossover events are more likely between than within sets of epistatically interacting genes.

In this paper I use a computational model based loosely on genetic algorithms (Holland 1975) to address the following two questions: 1) Does genome modularity affect evolvability under crossover? 2) If so, can genome modularity itself evolve?

The Model

The model consisted of a population of evolving individuals. In each generation the fitness of each individual was determined by its interaction with a simple but changing environment. Fitness determined how many offspring each individual produced. The parental generation was then discarded, and the procedure repeated with reproduction by the new generation.

The individuals

Individuals had several phenotypic traits with additive effects on individual fitness, each represented by a real-number value. Each trait was controlled by a unique set of genes, which interacted epistatically to determine the trait value. Each gene had a real-number value, and their interaction was represented by multiplication, so that each trait value was the product of the values of its associated genes. Each gene also had a position on a single linear chromosome. In addition to the genes under selection, chromosomes also contained neutral loci that had no effect on fitness but allowed the selected genes to disperse more widely on the chromosome. At the start of a run, each individual's chromosome was arranged to be either modular or non-modular. In non-modular chromosomes the genes and neutral loci were placed in randomized order. In modular chromosomes all genes affecting a given trait made up a contiguous block on the chromosome. Each such block was followed by a block of neutral alleles of uniform size. Thus genes were as close as possible to other genes affecting the same trait, and as far as possible from genes affecting other traits.

The environment and fitness

The environment consisted of an optimal value for each trait, represented as a real number within a specified range. These optima defined the value yielding the highest fitness for each trait, with fitness falling off as a linear function of distance from the optimum. At each time step these trait optima were changed by adding a random number from a normal distribution with mean of zero and a specified standard deviation. The new trait optima resulting from this addition were truncated to stay within the specified range.

The expected fitness of an individual was the sum of the fitness contributions of each of its traits, normalized so that the lowest fitness individual in the population had no offspring, and the expected total number of offspring equaled the nominal population size. The realized fitness, or actual number of offspring each individual had, was determined probabilistically for fractional fitnesses. For example, an individual with expected fitness 2.3 would have two offspring with probability 1, plus a third offspring with probability 0.3. Thus the actual population size fluctuated around the nominal population size.

Reproduction and genetic operators

After fitnesses were calculated, all individuals with realized fitness > 0 were placed on a parent list. The parent with the highest realized fitness was then mated with a randomly chosen second individual from the list, and two offspring with different genotypes were produced by crossover, so that each parental gene was represented once in the two offspring. The realized fitness of each parent was decremented by one, and if it was still positive they were returned to the parent list. This process was repeated until the list was either empty (if realized fitnesses summed to an even number), or contained a single parent with remaining realized fitness of one (if the total number of offspring was odd). If one parent was left, it produced its last offspring without recombination.

Three operators could cause genotypes to change between parents and offspring: mutation, inversion, and crossover. Mutation occurred as part of every reproduction. When an offspring was created, each of its gene values was altered slightly by adding a random number taken from a normal distribution with a mean of zero and a specified standard deviation.

The inversion operator occurred with a specified probability per offspring. Two locations on the offspring's chromosome were chosen randomly, and the ordering of genes beginning with one and ending with the other was reversed.

Crossover between parental chromosomes was attempted during each mating. First a single location between loci was chosen randomly as a crossover point. If the two parents shared the same genes at both positions adjacent to this point (i.e., if the two chromosomes were aligned there), crossover occurred. If, due to a past inversion, the parents had different genes at either adjacent location then crossover did not occur, and each parent produced one offspring without crossover. Following a crossover, offspring that had a copy of each gene survived and entered the next generation, while those that did not were deemed to have lost an essential gene, and died without entering the next generation. This biologically realistic procedure prevented loss of genes from the population which would otherwise have made the experiments difficult to interpret.

Experiments and results

In each run the population was seeded with a single individual, and grew to approximately the nominal population size in a single generation. The initial location of all trait optima was set at half the maximum, and the initial value of each gene prior to mutation was set at $(T_0)^{1/g}$, where T_0 = the initial trait optimum and g = the number of genes per trait. The effect was that runs started with all individuals at near-optimal values for each trait and with minimal variation between individuals. Both variation and sub-optimal trait values arose quickly through mutation, crossover, and environmental change. Each experiment consisted of 10 runs under each condition, with the same parameter settings in each run but different random number seeds. Unless otherwise noted, parameters were set as shown in Table 1.

Parameter	Setting
Nominal population size	100
Number of traits	10
Genes per trait	2
Neutral loci per trait	20
Mutation size (SD)	0.1
Inversion rate	0.001
Range of trait optima	0-100
Change in trait optima per generation (SD)	0.1
Generations per run	1000

Table 1: Standard parameter settings.

Experiment 1: Effect of modularity on the rate of evolution

The first experiment investigated the effect of modularity on the rate of evolution under recombination. In one treatment the seed individual had the ordering of genes on its chromosome randomized, and in the other treatment seed individuals had modular chromosomes, with all genes for the same trait adjacent to one another. The inversion rate was set to zero so that modularity did not change during the runs. Each run consisted of 1000 generations.

Average distance from phenotypic optima was smaller in the runs with modular genomes than with non-modular genomes, indicating that modularity increased the rate of adaptive evolution (Fig. 1).

Experiment 2: Evolution of modularity

The first experiment showed that populations evolved faster when genomes were modular than when they were not. The second investigated whether modularity itself could evolve. Each run was seeded by an individual with a non-modular (randomized) genome, but inversion was allowed at a rate of 0.001 per birth, so that the degree of modularity could change from parent to offspring. Runs were extended to 500,000 generations. Genome modularity was measured by calculating the average distance on the chromosome between pairs of genes affecting the same trait.

Starting from approximately the level of modularity expected under random distribution of genes, distances between epistatically interacting genes decreased over time, indicating that chromosomes did become more modular. This process occurred over a much longer time scale than the evolution of gene values (Fig. 2).

Discussion

How did modularity speed up evolution?

These experiments show that genome modularity can increase evolvability under recombination. This occurs because recombination has different consequences for loci with purely additive fitness effects than for loci with epistatic effects. In both cases recombination removes linkage disequilibrium, or correlations across individuals in the values of different loci. This can only be helpful for purely additive loci. Linkage disequilibrium among additive loci cannot speed up evolution, and tends to slow it down by allowing alleles with low fitness to persist or spread by “hitchhiking” along with high-fitness alleles they are linked to. In contrast, linkage disequilibrium among epistatic loci can create high-fitness combinations of genes. By breaking up these high-fitness combinations, recombination can reduce adaptation.

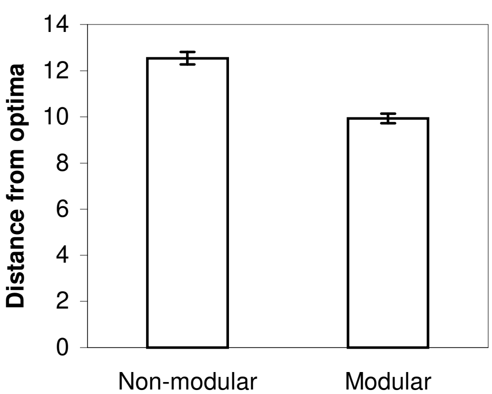

Figure 1: Average distance from trait value optima. Averages were calculated across all time steps within a run, then across runs. Error bars show +/- 1 SD across 10 runs. Populations with modular vs. non-modular genomes were closer to the optima, indicating faster adaptive evolution.

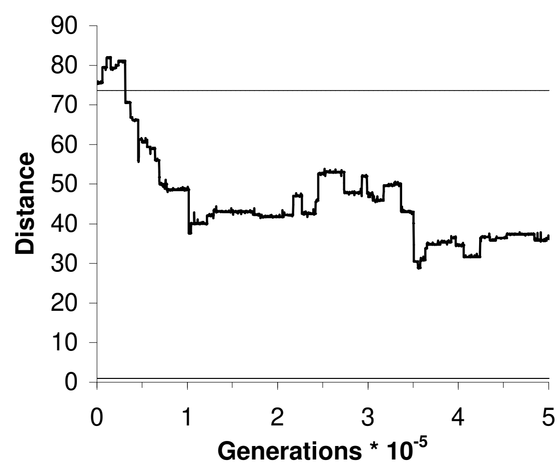

Figure 2: Decrease over time in the average distance (number of loci) between genes affecting the same trait. The upper line shows the expected value under random dispersal of genes, and the lower line shows the minimum possible distance. One run of 500,000 generations is shown.

In this model, genes for the same trait have multiplicative effects on fitness, which includes both additive and epistatic components, while genes controlling different traits have purely additive effects The importance of modularity is that it maximizes recombination among additive loci, and minimizes recombination among epistatic loci.

How did modularity evolve?

By definition, genome modularity has no effect on classical measures of fitness such as expected number of offspring. How then can it evolve? The answer can be framed in several ways. We can think of a more modular chromosome as spreading by hitchhiking on the reproductive success of the high-fitness descendants that it eventually gives rise to. To use a different metaphor for the same process, we can think of selection acting at both the level of individuals and the level of lineages. Although genome modularity is neutral under individual selection, it is favored under lineage selection because it provides a fitness advantage to lineages, increasing their probability of surviving and spawning new lineages.

Why didn't modularity evolve further?

Modularity did not reach its theoretical maximum, even though this would have allowed faster adaptive evolution (Fig. 2). Presumably this reflects the equivalent of a mutation-selection balance on genome modularity. Inversions play the role of mutations, because they create the variation selection acts on, allowing evolution to proceed. As modularity increases however, inversions are increasingly likely to reduce rather than further increase it. An equilibrium is reached when new inversions reduce modularity at the same rate that selection increases it.

Biological implications

The chromosomes of real organisms contain pairs and clusters of linked genes that are conserved over long time scales (Dandekar et al. 1998). There are several possible reasons for this conservation of proximity on the chromosome, including origination through ancient gene duplication events, co-regulation, and physical interaction during protein folding. However, many cases are probably not explained by these mechanisms. In these cases, clustering of functionally related genes may reduce the rate at which recombination breaks up co-adapted sets of genes (Dandekar et al. 1998). The current study supports this hypothesis, and extends it in several ways. It shows that selection for evolvability not only can maintain the linkage of co-adapted gene clusters, but can also create it. This solves what might otherwise appear to be a chicken-and-egg problem, in which linkage is required before genes can become co-adapted, but co-adaptation is required before there is selection pressure for linkage.

References

Ancel, L. W. and Fontana, W. 1999. Plasticity, Evolvability and Modularity in RNA. Santa Fe Institute Working Paper 99-12-077.

Dandekar, T., Snel, B., Huynen, M. & Bork, P. 1998. Conservation of gene order: a fingerprint of proteins that physically interact. *Trends in Biochemical Sciences* 23: 324-328.

Holland, J. H. 1975. *Adaptation in Natural and Artificial Systems*. MIT Press, Cambridge MA.

Wagner, G. P. & Altenberg, L. 1996. Complex adaptations and the evolution of evolvability. *Evolution* 50:967-976.