Seminars in Cell & Developmental Biology 35 (2014) 90–97

Contents lists available at ScienceDirect

Seminars in Cell & Developmental Biology

journal homepage: www.elsevier.com/locate/semcdb

Review

Evolving phenotypic networks in silico

Paul François*

Ernest Rutherford Physics Building, McGill University, 3600 rue University, H3A2T8 Montreal, QC, Canada

ARTICLE INFO

Article history:
Available online 20 June 2014

Keywords:
Evolution in silico
Biochemical adaptation
Immunology
Ligand discrimination
Somitogenesis
Fitness

ABSTRACT

Evolved gene networks are constrained by natural selection. Their structures and functions are consequently far from being random, as exemplified by the multiple instances of parallel/convergent evolution. One can thus ask if features of actual gene networks can be recovered from evolutionary first principles. I review a method for in silico evolution of small models of gene networks aiming at performing predefined biological functions. I summarize the current implementation of the algorithm, insisting on the construction of a proper "fitness" function. I illustrate the approach on three examples: biochemical adaptation, ligand discrimination and vertebrate segmentation (somitogenesis). While the structure of the evolved networks is variable, dynamics of our evolved networks are usually constrained and present many similar features to actual gene networks, including properties that were not explicitly selected for. In silico evolution can thus be used to predict biological behaviours without a detailed knowledge of the mapping between genotype and phenotype.

Contents

1. Introduction

Like any complex emergent process, evolution combines dynamics at different spatial and temporal scales, and for this reason can be challenging to study and model mathematically. Microevolution corresponds to changes of allele frequencies in a population, over relatively "short" time-scales, and population genetics has long been the central mathematical theory to study microevolution [1]. Recent real-time experimental studies have also advanced our understanding of microevolution, e.g. long-term evolutionary experiments in the lab [2,3], artificial selection of complex mechanisms (such as bacterial altruism [4]) or observation of fast evolving systems (like the flu [5]).

Macroevolution, evolution of high order structures over long time-scales, is more challenging to study. It is of course still impossible to observe experimentally and thus can be studied only indirectly. Most data come from retrospective studies of genomes and fossils, having evolved over 4 billions years. Full access to ancestral phenotypes and measures of ecological pressures are impossible so that macroevolutionary mechanisms for apparition of complex features (such as full-blown organs or signalling pathways) remain speculative. As a consequence, very different views

* Tel.: +1 514 398 1635.
E-mail address: paulf@physics.mcgill.ca

http://dx.doi.org/10.1016/j.semcdb.2014.06.012

coexist: for instance, while many biologists (like Stephen Jay Gould [6]) think that evolved structures are historically contingent, others (like Simon Conway-Morris [7]) have used spectacular examples of convergent evolution to argue that solutions found by evolution are much more constrained than usually thought. Modern experimental attempts include the study of evolutionary history in conjunction to development ("evo-devo"), supported by genomic studies [8].

But just like population genetics is the central theory underlying microevolution, a quantitative theoretical framework would be useful for macroevolutionary studies. In particular, one question arising is the nature of constraints on evolvable biological functions: given a complex phenotype, can we use some mathematical theory to predict anything on the underlying gene networks? The issue is that we do not have (yet) a proper formalism to answer such questions: among other problems and despite recent advances (see e.g. [9–11]) the nature of the mapping between phenotype and genotype is still an open question. For this reason, we turn to computational approaches and propose a generic in silico evolution procedure to "predict" what kind of networks can evolve to perform a given biological function [12,13]. In the following, I first describe our method and then discuss three interesting case-studies.

2. Network implementation and algorithm philosophy

In this section, I summarize how we model gene networks and their simulated evolution.

There are two levels in the algorithm: the individual level where genotypes and phenotypes of individuals are defined and computed, and the population level, where evolution is performed.

2.1. Individual level

In our approach, an individual genotype is a mathematical object encoding dynamics of a gene network. Networks consist in bipartite graphs. The first category of nodes is interacting components, typically proteins or DNA sequence. They are themselves connected to the second category of nodes, corresponding to interactions. A grammar of possible interactions is predefined, accounting for various biochemistry, such as transcription, transcriptional regulations, phosphorylations, protein–protein interactions. A network behaviour (i.e. its phenotype) is modelled using ordinary differential equations. We use classical biochemical kinetics to account for the various interaction, e.g. mass-action laws for protein–protein interactions, or Hill functions for transcriptional interactions [14].

To be more specific, let us consider one example, similar to one of the adaptive network evolved in [15]. A full-blown representation of the network is displayed in Fig. 1A, and a simplified representation of the same network in Fig. 1B. There are three proteins (subsequently called S_0, S_1, S_2) that we call 'Species'. `regulatory_module`, DNA are nodes used to model regulatory and coding sequence of gene S_1. Finally PPI is an interaction node corresponding here to a complexation between S_0 and S_1 into S_2.

To this graph correspond differential equations. Equations are automatically generated by the algorithm to account for the interactions. For node S_1, the regulatory module and DNA part here simply encode a default basal transcription rate ρ. The PPI interaction adds a non linear forward interaction term $\gamma S_0 S_1$ for complex formation, and a linear backward term αS_2 for complex dissociation. Finally, we assume that all species have a linear degradation or dilution rate. For this case, the complete set of differential equations for S_1 and S_2 thus is

$$\dot{S}_1 = \rho - \delta_1 S_1 - \gamma S_1 S_0 + \alpha S_2 \quad (1)$$

$$\dot{S}_2 = \gamma S_1 S_0 - (\alpha + \delta_2) S_2 \quad (2)$$

All parameters in these equations are randomly chosen and selected by the algorithm.

In the present case, there is no equation for S_0 because it is an external Input, with a prescribed dynamics (here a sequence of steps of random heights). Integration of networks dynamics under control of this Input is performed. Fig. 1 illustrates dynamics of this network for $\delta_1 = 0$. This makes the Output variable S_2 adaptive, i.e. after a change of Input value, its values changes before returning to its initial value. This adaptive response can be quantified in various ways, for instance by measuring the deviation from the baseline or by quantifying how the stationary value of the Output depends on the stationary value of the Input. These quantities can be used to define a coarse-grained phenotype. From this phenotype, a fitness or scoring function is computed by the algorithm and is later used for selection (see below for a more detailed description of evolution of adaptive behaviour corresponding to Fig. 1).

2.2. Population level

Our algorithm works very much like actual evolution and other evolutionary algorithms: (1) it takes a population of genotypes; (2) computes their phenotype and fitness as indicated above; (3) selects and mutates networks; and (4) iterates this process over as many generations as desired.

Selection is based on the network fitness. We run the simulations in a very elitist mode. At each generation, the worst half of the networks (based on the fitness) is discarded. Then the best half is ordered, kept, duplicated, and the duplicated half is mutated. To ensure some population mixing even among the best networks, we also systematically add some small random component in the fitness.

Mutations consist in random modifications of the genotype, via changes of either parameters or network topology (addition or removal of nodes). It is important to stress at this stage that individual networks can grow with time, which is different from classical genetic algorithms where genome size is fixed. On the one hand, this prevents any simple implementation of genetic cross-over, but on the other hand, network growth opens up the possibility of dimensionality increase in phase-space and of evolution of new combinatorics that could be crucial to implement new complex dynamics.

Given our pre-defined grammar of interactions, each possible evolutionary move is systematically computed at each generation for each network, and actual mutations are randomly drawn. Individual mutations are assumed to be Poisson processes, with a fixed pre-defined rate. Typically, we choose rates so that the most probable move is a change of kinetics, then second most probable move is removal of nodes, and least probable move is addition of new interactions. This fits the idea that most evolutionary moves are neutral or deleterious, and that addition of new function should be a priori rare.

The time and nature of the next mutation for a given network is chosen using a Gillespie algorithm [16]. An evolutionary time therefore needs to be defined. As networks grow, generation time is dynamically changed so that the average number of mutations per generation per individual is fixed (currently taken to one). This implements an analogue of the "Drake's rule", the idea that the mutation rate inversely scales with genome size [17]. This also prevents uncontrolled explosion of network size that would naturally occur given the combinatorial explosion of possible interactions as networks grow.

2.3. Evolution of specific biological functions

A key aspect of any evolutionary computation is the choice of scoring (or fitness) function. By analogy with energy minimization