1. Purpose

The aevol model was designed to study the long-term, indirect selective pressures that can shape the structural features of genomes, like the number of genes, the amount of non-coding DNA, the presence of gene clusters.... Indeed, classical genomics often focus on short-term pressures like mutational biases (e.g. “insertions are spontaneously more frequent than deletions”) or direct selective costs (e.g. “a large genome is long and costly to replicate and hence reduces the fitness”), while population genetics, evolutionary computation and artificial life approaches have revealed the existence of other types of selective pressures, acting in a longer term. Such pressures, referred to as “indirect selection”, “second-order selection” or “modifier selection”, act on traits that do not directly influence the immediate fitness but do influence the mutational robustness and/or the evolvability of the phenotype (and hence the average fitness of the offspring). The best known examples of such traits are the per-base mutation rate and the robustness of gene networks, but the structural features of the genome can also play a major role: the amount of non-coding DNA influences the average impact of a point mutation, the number of repeated sequences influences the frequency of intra-chromosomal recombination, etc. Thus the aim of the model is to investigate whether the indirect selection of robustness and/or evolvability contributes (along with mutational biases and direct selective costs) to control the evolution of such structural features.

2. Description of the model

When we study indirect selective pressures, we aim at investigating the unexpected features that individuals acquire because they are part of an evolving population. We are thus interested in the local (individual) properties that emerge from a global context of competition. Because it involves an explicit “micro” level in addition to the “macro” one, individual-based modelling is particularly relevant to study this phenomenon of “micro-emergence”. This is why we designed an individual-based model, called aevol (artificial evolution). When designing the individuals, we took into account three organisation levels that play a role in the robustness and the evolvability of the phenotype, namely the structure of the genome, the topology of the protein network and, of course, the phenotype itself.

2.1 Overview

The aevol platform simulates the evolution of a population of $N$ artificial organisms. The population size, $N$, is constant over time. Each artificial organism owns a circular\(^1\), double-strand\(^2\) chromosome which is actually a string of binary nucleotides, 0 being complementary of 1 and reciprocally (Figure 1). This chromosome contains coding sequences (genes) separated by non-coding regions. Each coding sequence is detected by a transcription-translation process and decoded into a “protein” able to either activate or inhibit a range of abstract “biological functions”. The interaction of all proteins yields the set of functions the organism is able to perform. Those global functional capabilities constitute here the phenotype. Adaptation is then measured by comparing the phenotypic capabilities to an arbitrary set of functions to perform to survive in the environment. The most adapted individuals have higher chances of reproduction: $N$ new individuals are created by reproducing preferentially the most adapted individuals of the parental generation. In the default

\(^1\) This avoids edge effects in the frequency of rearrangements.  
\(^2\) This allows us to perform sequence inversions.
setting, reproduction is strictly asexual, but options are available to allow for lateral transfer. While a chromosome is replicated, it can undergo point mutations, small insertions and small deletions, but also large chromosomal rearrangements: duplications, large deletions, inversions, translocations. The various types of mutation can modify existing genes, but also create new genes, delete some existing genes, modify the length of the intergenic regions, modify gene order…

![Figure 1: Overview of the aevol model.](image)

### 2.2 From genotype to phenotype

In this section, we describe in details the transition from genotype to phenotype. To offer several levels of reading, the justifications of most choices appear in footnotes.

**Transcription**

From the genomic sequence, the phenotype computation starts by searching in both strands for promoter and terminator sequences, delimiting the transcribed regions. Promoters are sequences whose Hamming distance $d$ with a pre-defined consensus is less or equal than $d_{\text{max}}$. In the default setting, the consensus\(^1\) is 0101011001110010010110 (22 base pairs) and up to $d_{\text{max}} = 4$ mismatches are allowed. Terminators are sequences that would be able to form a stem-loop structure, as the ρ-independent bacterial terminators do\(^2\). By default the stem size is 4 and the loop size 3, hence terminators had the following structure: $abcd * * * \overline{d}e\overline{b}a$, where $a = 0$ if $a = 1$, and conversely.

The transcription algorithm proceeds as follows. We first search for promoters on one strand. Then, for each promoter, we walk on the strand until we find a terminator. This delimits the transcribed region. Note that several promoters can share the same terminator, in which case we obtain

---

1. This consensus is long enough to ensure that random, non-coding sequences have a low probability to become coding by a single mutation event. It is not a palindrome, meaning that a given promoter promotes transcription on one strand only.
2. In the first versions of the model, terminators were defined like the promoters, i.e. with a consensus sequence. Since we needed frequent terminators to limit gene overlaps, we chose a short consensus, 11111 for instance. However, this turned out to be problematic because no coding sequence could contain this short motif, which heavily constrained the evolution. Thus we needed both long and frequent terminators, which is not possible with the consensus method. This is why, in the end, we used the biological way.
overlapping transcribed regions. We assign an expression level\(^1\) \(e\) to the region, according to the similarity of the promoter with the consensus: \(e = 1 - \frac{d}{d_{\text{max}} + 1}\). These steps are then repeated on the other strand.

**Translation & Protein representation**

Once all transcribed regions have been localized, we search inside each of them for the initiation and termination signals of the translation process. These signals delimit the coding sequences. The initiation signal is the motif 011011***000 (Shine-Dalgarno-like signal followed by a start codon, 000 by default). The termination signal is simply the stop codon, 001 by default\(^2\). Each time an initiation signal is found, the following positions are read three by three (codon by codon) until a stop codon is encountered. If no stop codon is found in the transcribed region, no protein is produced. A given transcribed region can contain several coding sequences (overlapping or not), meaning that operons are allowed.

Then the translation process must determine the phenotypic contribution of each detected coding sequence, by defining the functional abilities of the protein it encodes. To do so in the simplest way, we use the fuzzy logic framework and the corresponding theory of possibility. We consider an abstract set of functions that can be performed. This set is called \(\Omega\). Each protein can contribute to or inhibit a subset of \(\Omega\), with a variable *degree of possibility* depending on the function: some functions are more possible than others. Each protein is thus described by a *fuzzy subset* of \(\Omega\) and the coding sequence encodes the parameters of this subset. The decoding process is detailed in the following paragraphs.

To keep the model simple, \(\Omega\) is one-dimensional space, more precisely a real interval: \(\Omega = [a, b]\) (\(a = 0\) et \(b = 1\) by default). This means that in the model, a biological function is simply a real number. Since \(R\) is an ordered set, some “biological functions” are closer than others: the function 0.10, for instance, is closer to the function 0.11 than to the function 0.20, just as – in a very informal manner – glucose metabolism can be considered to be closer to lactose metabolism than to DNA repair.

Now that \(\Omega\) is defined as the real interval \([a, b]\), we can represent the fuzzy subset of each protein by a mathematical function \(f\) from \(\Omega = [a, b]\) to \([0, 1]\), called *possibility distribution*. It defines for each “biological function” \(x\) the degree of possibility \(f(x)\) with which the protein can perform \(x\). We have chosen to use piecewise-linear distributions with a triangular shape (simplified bell-shaped distributions, see Figure 1). Three parameters are necessary to fully characterize such distributions:

- the position \(m\) (“mean”) of the triangle on the axis, which corresponds to the main function of the protein,
- the height \(H\) of the triangle, which determines the degree of possibility for the main function,

\(^1\) This modulation of the expression level models only (in a simplified way) the basal interaction of the RNA polymerase with the promoter, without additional regulation. The purpose here is not to accurately model the regulation of gene expression, but rather to provide duplicated genes a way to reduce temporarily their phenotypic contribution while diverging toward other functions. It also induces a link of co-regulation between the coding sequences of a same transcribed region, which is a necessary property to study the evolution of operons.

\(^2\) As for the transcription, the initiation signal is longer and hence rarer than the termination signal. This ensures that non-coding regions have a low probability to become coding.
• the half-width \( w \) of the triangle, which represents the functional scope of the protein and is thus a way to quantify its pleiotropy.

Hence the protein can be involved in the “biological functions” ranging from \( m - w \) to \( m + w \), with a maximal degree of possibility for the function \( m \). The fuzzy subset of the protein is thus the interval \( ]m - w, m + w[ \subset \Omega \). While \( m \) and \( w \) are fully specified by the coding sequence, \( H \) is a composite parameter taking into account both the quantity of the protein in the cell and the efficiency of the molecule: \( H = e|h| \), where \( e \) is the expression level of the transcribed region and \( h \) is specified by the coding sequence. As we shall see below, the sign of \( h \) determines whether the protein contributes to or inhibits the functions \( ]m - w, m + w[ \).

In computational terms, the coding sequence is interpreted as the mix of the Gray\(^1\) codes of the three parameters \( m, w \) and \( h \). In more biological terms, the coding sequence is read codon by codon and an artificial genetic code is used to translate it into the three real numbers \( m, w \) and \( h \). This genetic code (shown in the Figure 1), two codons are assigned to each parameter. For instance, \( w \) is calculated from the codons \( W_0 = 010 \) and \( W_1 = 011 \). All the \( W \) codons encountered while reading the coding sequence form the Gray code of \( w \). The first digit of the Gray code of \( w \) is a 0 (resp. a 1) if the first \( W \) codon of the sequence is a \( W_0 \) (resp. a \( W_1 \)). In the example shown by Figure 1, the coding sequence contains three \( W \) codons: \( W_1 \ldots W_1 W_0 \). The full Gray code of \( w \) would thus be 110 in this example, which corresponds to 100 in the traditional binary code, that is, 4. Hence, if the coding sequence contains \( n \) \( W \) codons, we get an integer comprised between 0 and \( 2^n - 1 \). A normalisation enables us to bring the value of the parameter in the allowed range, specified at the beginning of the simulation. The parameter \( w \), which determines the width of the triangle, is normalised between 0 and \( w_{\text{max}} \) (the value of \( w_{\text{max}} \) being set at the beginning of the simulation): The integer value, 4 for in our example, is multiplied by \( \frac{w_{\text{max}}}{2^n - 1} \). The values of parameters \( m \) and \( h \) are obtained in a similar manner, \( m \) being normalised between \( a \) and \( b \), and \( h \) between -1 and 1. If \( h \) is positive, the protein is activator: It activates the functions \( ]m - w, m + w[ \). If it is negative, it inhibits these functions. If it equals naught, it does not contribute to the phenotype.

**Functional interactions between proteins & Phenotype computation**

The fuzzy subsets of several proteins – or, in graphical terms, their triangles – can overlap partially or completely. This means that several proteins can contribute to a same “biological function”, meaning that they have a functional interaction\(^2\). Thus, to know the degree of possibility with which the individual can perform a given function, we must take into account all the contributing proteins and combine their elementary possibility distributions. This can be done easily because the fuzzy logic framework provides the operators to compute the complement (NOT), the union (OR) and the intersection (AND) of elementary fuzzy subsets. The global functional abilities of an individual are the functions that are activated AND NOT inhibited, a function being considered as activated (resp. inhibited) if it is activated (resp. inhibited) by the protein 1 OR by the protein 2 OR by the protein 3, as so on. More formally, if \( A_i \) is the fuzzy subset of the \( i \)-th activator protein, and \( I_j \) the fuzzy

\(^1\) The Gray code is a variant of the traditional binary code. It is widely used in evolutionary computation because it avoids the so-called “Hamming cliffs”: in the Gray code representation, consecutive integers are assigned bit strings that differ in only a single bit position.

\(^2\) The term is to be understood in a broad sense. It is not necessarily a physical interaction. It could also be the involvement in a same metabolic pathway or a same signalling cascade, for instance.
subset of the \( j \)-th inhibitory protein, then the fuzzy subset of feasible functions is 
\[ P = \bigcup_{j} A_{j}\cap \bigcup_{i} I_{i} \]. We use Lukasiewicz’ fuzzy operators\(^1\) to perform this combination:

\[
\begin{align*}
\text{NOT} : \quad f_{A_{j}}(x) & = 1 - f_{A_{j}}(x) \\
\text{OR} : \quad f_{A_{j} \cup A_{2}}(x) & = \min(f_{A_{j}}(x) + f_{A_{2}}(x), 1) \\
\text{AND} : \quad f_{A_{j} \cap A_{2}}(x) & = \max(f_{A_{j}}(x) + f_{A_{2}}(x) - 1, 0)
\end{align*}
\]

More intuitively, to compute the phenotype (the global possibility distribution \( f_P \) which describes the functional abilities of the individual), we sum up the possibility distributions of all activator proteins, do the same with the possibility distributions of all inhibitory proteins, and finally subtract the second sum from the first one, but at each step the result is kept between 0 and 1. Note that these thresholds, 0 and 1, induce non-linear effects: the joint efficiency of two proteins is not always equal to the sum of their elementary efficiencies.

2.3 Environment, adaptation and selection

The environment in which the population evolves is also modelled by a fuzzy subset \( E \subset \Omega \) and hence by a possibility distribution \( f_E \) on the interval \([a,b]\). \( f_E \) specifies the optimal degree of possibility for each “biological function” and it can be naught for some functions. This distribution is chosen at the beginning of the simulation and can randomly vary over time if desired. An example of environment is shown in Figure 2. The environment can be constant over time or fluctuate randomly around a specified curve.

The adaptation of an individual is measured by the gap \( g \) between its functional abilities (\( f_P \), the phenotypic possibility distribution) and the optimal ones (\( f_E \), the environmental distribution):

\[
g = \int_a^b f_E(x) - f_P(x) \, dx.
\]

As shown by Figure 2, this measure penalizes both the under-realised functions and the over-realized ones.

---

\(^1\) The most widely used operators in fuzzy logic are actually the so-called min-max operators, but they are not appropriate to model the cooperation between two proteins because the resulting degree of possibility is equal to one of the elementary degrees. On the contrary, with Lukasiewicz’ operators, the resulting degree is the sum of both elementary degrees (or 1 if the sum would exceed 1).
The population is completely asexual and is managed very simply: The population size, \( N \), is fixed and the population is completed renewed at each time step (generation). A probability of reproduction is assigned to each of the \( N \) potential parents according to its adaptation measure \( g \) and the actual numbers of reproductions are drawn by a multinomial drawing. Three methods are available to assign the probability of reproduction knowing the gap \( g \). These three selection schemes are respectively called “fitness-proportionate selection”, “linear ranking selection” and “exponential ranking selection” (see Blickle and Thiele, 1996, for more details). In the linear ranking scheme, the individuals were sorted by decreasing gap, such that the best individual (with the smallest gap) had rank \( N \). Then the probability of reproduction of the individual with rank \( r \) was

\[
\frac{1}{N} \left( \eta^- + \left( \eta^+ - \eta^- \right) \frac{r - 1}{N - 1} \right),
\]

with \( \eta^+ = 1.998 \) and \( \eta^- = 2 - \eta^+ \). Note that \( \eta^+ \) and \( \eta^- \) are the expected numbers of reproductions for the best and the worst individuals, respectively. In the fitness-proportionate selection, the probability of reproduction of an individual with gap \( g \) is

\[
\frac{e^{-kg}}{\sum_{i=1}^{N} e^{-kg_i}}
\]

where \( k \) is a parameter controlling the strength of selection.

### 2.4 Mutations

Every time an individual reproduces, its genome is replicated and several types of mutations can occur during this replication. Let us assume a circular chromosome of \( L \) positions, numbered from 0 to \( L-1 \). It can undergo:

- **Point mutations:** a position \( p \) is randomly (uniformly) chosen on the chromosome. If this position is a 0, it is changed to 1. Conversely, if it is a 1, it is changed to 0.

- **Small insertions:** a position \( p \) is chosen as above. A short random sequence (from 1 to 6 base pairs) is inserted between the positions \( p-1 \) and \( p \).

- **Small deletions:** a position \( p \) is chosen as above. We randomly draw the number \( n \) of positions to be deleted (between 1 and 6 again) and we excise the positions \( \{p, p+1, \ldots, p+n-1\} \).

- **Large deletions:** Two positions \( p_1 \) and \( p_2 \) are chosen on the chromosome and the segment ranging from \( p_1 \) to \( p_2 \) (included) in the clockwise sense is excised. Since the chromosome is circular, \( p_2 \) can be smaller than \( p_1 \). In this case, we excise the segments \( \{p_1, \ldots, L-1\} \) and \( \{0, \ldots, p_2\} \).

- **Inversions:** Like the deletion, except that the segment \( \{p_1, \ldots, p_2\} \) is inverted, (that is, replaced by the reverse complementary sequence) instead of being excised. The chromosome length is unchanged.

- **Duplications:** Two positions \( p_1 \) and \( p_2 \) are chosen on the chromosome and the segment \( \{p_1, \ldots, p_2\} \) is copied. A position \( p_3 \) is randomly chosen on the chromosome and the copied segment is inserted (in its original orientation) between the positions \( p_3-1 \) and \( p_3 \).

- **Translocations:** Like the duplication, but the segment \( \{p_1, \ldots, p_2\} \) is excised (rather than copied) before being re-inserted between the positions \( p_3-1 \) and \( p_1 \). The chromosome length is unchanged. The duplication can be seen as a “copy and paste” and the translocation as a “cut and paste”.

To choose the breakpoints of the rearrangements, \( p_1 \) and \( p_2 \), the most realistic manner is to search for sequence similarities and use repeats as the breakpoints. This option is available but is very
time-consuming. Hence we propose by default a faster option where any position can be used as a breakpoint. In this case, $p_1$ and $p_2$ are uniformly chosen on the chromosome.

For each of the seven types of mutation, a per-position rate $u$ is chosen at the beginning of the simulation. The mutation algorithm proceeds as follows: when an individual reproduces, we compute the four numbers of rearrangements its genome will undergo. The number of large deletions is drawn from the binomial law $B(L, u_{\text{largedel}})$, the number of duplications from the law $B(L, u_{\text{duplic}})$, and so on. All these rearrangements are then performed in a random order. The genome length can vary throughout this process: if we perform an inversion after a duplication, the inversion breakpoints $p_1$ and $p_2$ are chosen between 0 and $L' - 1$, where $L'$ is the genome size after the duplication. Successive rearrangements are thus not independent. Once all rearrangements have been performed, we draw the three numbers of local mutations (point mutations, small insertions and small deletions) and we perform all these events in a random order.

3. A typical run

For a relatively large range of parameter values, the behaviour of the model exhibits a number of regularities. To illustrate them, we present here a typical run whose parameters are summarized in Table 1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size</td>
<td>$N$</td>
<td>1,000</td>
</tr>
<tr>
<td>Size of the initial (random) genome</td>
<td>$L_{\text{init}}$</td>
<td>5,000 base pairs</td>
</tr>
<tr>
<td>Promoter sequence</td>
<td>-</td>
<td>010101100111001010110</td>
</tr>
<tr>
<td></td>
<td></td>
<td>with up to $d_{\text{max}} = 4$ mismatches</td>
</tr>
<tr>
<td>Terminator sequences</td>
<td>-</td>
<td>$abcde * * * d\ddot{e}\ddot{b}\ddot{a}$</td>
</tr>
<tr>
<td>Initiation signal for the translation</td>
<td>-</td>
<td>011011***000</td>
</tr>
<tr>
<td>Termination signal for the translation</td>
<td>-</td>
<td>001</td>
</tr>
<tr>
<td>Genetic code</td>
<td>-</td>
<td>See Figure 1</td>
</tr>
<tr>
<td>Global set of “biological functions”</td>
<td>$\Omega$</td>
<td>[0, 1]</td>
</tr>
<tr>
<td>Maximal pleiotropy of the proteins</td>
<td>$w_{\text{max}}$</td>
<td>0.033</td>
</tr>
<tr>
<td>Environmental possibility distribution</td>
<td>$f_{E}$</td>
<td>See Figure 2</td>
</tr>
<tr>
<td>Selection scheme</td>
<td>-</td>
<td>Linear ranking</td>
</tr>
<tr>
<td>Selection intensity</td>
<td>$\eta^+$</td>
<td>1.998</td>
</tr>
<tr>
<td>Point mutation rate</td>
<td>$u_{\text{point}}$</td>
<td>$10^{-5}$ per position</td>
</tr>
<tr>
<td>Small insertion rate</td>
<td>$u_{\text{smallins}}$</td>
<td>$10^{-5}$ per position</td>
</tr>
<tr>
<td>Small deletion rate</td>
<td>$u_{\text{smalldel}}$</td>
<td>$10^{-5}$ per position</td>
</tr>
<tr>
<td>Large deletion rate</td>
<td>$u_{\text{largedel}}$</td>
<td>$10^{-5}$ per position</td>
</tr>
<tr>
<td>Duplication rate</td>
<td>$u_{\text{duplic}}$</td>
<td>$10^{-5}$ per position</td>
</tr>
<tr>
<td>Inversion rate</td>
<td>$u_{\text{inv}}$</td>
<td>$10^{-5}$ per position</td>
</tr>
<tr>
<td>Translocation rate</td>
<td>$u_{\text{transloc}}$</td>
<td>$10^{-5}$ per position</td>
</tr>
<tr>
<td>Length of small indels</td>
<td>-</td>
<td>Uniform law between 1 and 6 positions</td>
</tr>
<tr>
<td>Length of rearrangements</td>
<td>-</td>
<td>Uniform law between 1 and $L$ positions</td>
</tr>
</tbody>
</table>

Table 1: Parameter values used for the run detailed in this section.
As shown by Figure 3, the initial genome (a random one, identical for all individuals) contains few genes (two in this typical run, one activator and one inhibitor), the rest of the chromosome being non-coding. The functional capabilities of the individuals are thus very limited. However, during the first generations, the initial genes – as well as some adjacent non-coding sequences – are rapidly duplicated, which causes a temporary explosion of genome size (Figure 5). The copied genes undergo local mutations that change the parameters of the protein: for instance, when a $M$ codon is mutated, the “triangle” of the protein is translated on the functional axis and new biological functions can thus be performed. In a similar manner, its height can be tuned by mutating the promoter (and hence the expression level $e$) or the $H$ codons in the coding sequence (and hence the value of $h$). Thus a biologically-relevant process of gene acquisition by duplication-divergence takes place in the simulation.

Figure 3: Initial and final individuals in a typical run.
This burst of genome size is, however, only temporary (see Figure 5). Some genes and some non-coding sequences are lost. After about 6,000 generations, genome size stabilises at an equilibrium value (of the order of 10,000 base pairs in this example). This equilibrium size is independent of the size chosen for the initial genome. It is important to note that this equilibrium does not correspond to the loss of all non-coding sequences. On the contrary, in this example, more than 80% of the genome remains non-coding until the end of the simulation.

Once the amount of non-coding positions is stabilised, the adaptation of the individuals continues to improve (the gap $g$ decreases), but more and more slowly. The existing coding sequences are modified by local mutations. New genes continue to appear by duplication-divergence, but less often than at the beginning, because most duplications are deleterious once the phenotype is close to the optimum. Some coding sequences also appear by local mutations in existing transcribed regions, which can lead to overlapping coding sequences (Figure 4).

**Figure 3**: Evolution of the adaptation measure (gap $g$) and of some genomic features in a typical run.
4. Strengths and limits of the model

The main strengths of this model are the following:

- The architecture of the genome, our principal object of interest, is biologically sound in the sense that (i) it includes both coding and non-coding sequences, (ii) it takes into account the notion of gene product, and (iii) the function of a gene is locally determined by its sequence, and not pre-defined according to its locus (as in classical genetic algorithms for instance).

- The architecture of the genomes possesses real degrees of freedom. The number of genes, the amount of non-coding-DNA and gene order can evolve by local mutations and by large rearrangements. Besides, it is possible for local mutations and for large rearrangements to modify genome structure (e.g. gene order, amount of non-coding DNA) without necessarily affecting the phenotype and the fitness. A given phenotype can thus be encoded by various genetic architectures; hence several genetic architectures of equal fitness can be in competition. This allows us to study the indirect selective pressures which will determine the outcome of this competition. For instance, can a specific amount of non-coding DNA and/or a specific gene order be indirectly selected?

- The complexity of the phenotype is not pre-defined once and for all. It is on the contrary allowed to co-evolve with the genome. In that sense, the model differs from the other genetic algorithms using a variable gene number, like the “Messy GA” (Goldberg et al., 1989) or the “Virtual Virus” (Burke et al., 1998). In both algorithms, the complexity of the phenotype was pre-defined and should not vary while gene number could. This forced the authors to imagine a biologically-unrealistic daemon to choose the expressed genes. In our model, the evolution of gene number is not perturbed by this type of artefact. Hence we can ask biologically relevant questions about the evolution of gene number: For instance, will it stabilise without selective cost on genome size?

- The genotype-phenotype map is not built on a “one gene – one trait” principle. On the contrary, it involves a protein network which exhibits two important biological features, namely pleiotropic genes and polygenic traits. These features are important here because they influence the phenotypic effect of gene mutations, which is a major component of the mutational variability of the phenotype (along with the mutation rate and the structure of the genome). In other words, the complexity of the protein network is a key level which influences the mutational robustness and the evolvability of the phenotype. The genome structure and the topology of the protein network can thus be indirectly linked by the fact that they both influence the same property. Because the level of gene pleiotropy is controllable in our model (through $w_{\max}$), we can control the complexity of this network and observe the consequences on genome structure at the evolutionary time scale. For instance, when protein interactions are more numerous, is gene order more constrained? Is there the same quantity of non-coding DNA?

- As in all artificial evolution approaches, we have an exhaustive knowledge of the kin relationships and of all the mutations that occurred. We can thus easily reconstruct the line of descent of the final organisms and study the mutations which occurred on this lineage – that is, the mutations which went to fixation. This is especially useful to reveal the keys of the evolutionary success: is it only the immediate fitness, or does genome size or gene order matter too? Can
beneficial mutations be actually counter-selected because they appear in genomes that are not optimally organised (from the robustness/evolvability point of view)?

However, any modelling approach implies a number of simplifications and the aevol model does not escape this rule. The main simplifications were performed at the functional level:

- We assumed the existence of a one-dimensional axis of “biological functions”, but it would be very difficult to place real biological functions on such a space, because (i) it would require a rigorous definition of the notion of function, with an homogenous level of description for all described functions and (ii) even in this case, the neighbourhood relationships between several functions are much more complicated than a simple ordering. A multi-dimensional space would certainly be more accurate. Hence the model is clearly not suited to study the evolution of a given gene network in a given (real) species. It rather aims at investigating the minimal conditions that allow for the emergence of complex evolutionary relationships between the genotype-phenotype map and the architecture of the genome. In that respect, the simplified formalism we have chosen already allows for a rich behaviour and it is thus necessary to fully understand it before adding new parameters.

- We assumed very simple possibility distributions for the proteins, with only one lobe: a protein can perform several functions but these are necessarily close on the functional axis. In real organisms, proteins can actually perform very different functions. Allowing several lobes in the possibility distribution would therefore provide a more general description of gene pleiotropy. This could be done by using longer codons or a richer alphabet. However, it is again necessary to fully understand the behaviour of the simplest model before adding new parameters.

5. References

