EvoEvo Deliverable 2.1

Guillaume Beslon, Paul Andrews, Carole Knibbe, Charles Rocabert

SEVENTH FRAMEWORK PROGRAMME

EvoEvo Deliverable 2.1

Specifications of the genome-network model

Due date:	M3 (January 2013)
Person in charge:	Guillaume Beslon
Partner in charge:	INRIA
Workpackage:	WP2 (Development of an integrated modelling platform)
Deliverable description:	Specifications of the genome-network model: Description of the modeling choices for the genome-network integrated model. This model should include a realistic genomic structure as well as a metabolic network translated from the genome.

Revisions:

Revision no.	Revision description	Date	Person in charge
1.0	First version of the specification of the genome-network model	27/01/14	C. Rocabert (INRIA)
1.1	Corrections by G. Beslon	29/01/14	C. Rocabert (INRIA)
1.2	Corrections by G. Beslon	30/01/14	G. Beslon (INRIA)
1.3	Corrections by C. Knibbe and P. Andrews	30/01/14	G. Beslon (INRIA)
1.4	Adding references section	30/01/14	G. Beslon (INRIA)
1.5	Modification of figure 7	30/01/14	G. Beslon (INRIA)
1.6	Fix an error in kinetic equation	30/01/14	C. Rocabert (INRIA)

Table of Contents

1. Introduction

> « *Although there has been much discussion on what is the appropriate level on which Darwinian selection operates, we now know that in many cases the interesting features arise through the occurrence of multiple levels of selection which act in concordance and/or in conflict.* »
>
> Hogeweg and Takeuchi (2002)

Micro-organisms react to many environmental changes by evolving through mutations and Darwinian selection. This remarkable ability to quickly evolve suggests that it could itself have been selected, if **evolution favored the most evolvable lineages.** This "evolution of evolution" relies on two important concepts: the **genotype-to-phenotype mapping** and the **fitness landscape**. The central concept of EvoEvo is the following: if the genotype-to-phenotype mapping and the fitness landscape are allowed to change over time, if they can be (indirectly) selected, then they can evolve and acquire properties that could favor evolution in changing environments.

Thus, to model and study EvoEvo, it is necessary to deal with **multiple levels of organization** (genetic structure and regulation, protein-protein network, metabolic network, population and species networks). INRIA and UU have developed independently two formalisms that are specifically dedicated to the study of indirect selection. INRIA used the "sequence-of-nucleotides" formalism to develop the *aevol* model (Knibbe, 2007a; Knibbe 2007b). Using this model, INRIA showed that indirect selection could select specific genetic and transcriptomic structures depending on the mutational and selective pressure (Knibbe, 2007b; Beslon, 2010a; Beslon, 2010b). UU proposed the "pearls-on-a-string" (*PoaS*) formalism and used it to show that, in time-varying environments, regulation networks, metabolic networks and species networks can acquire structures that increase the evolvability of the organisms (Crombach & Hogeweg, 2008). However, both models are restricted to specific levels of organization (figure 1).

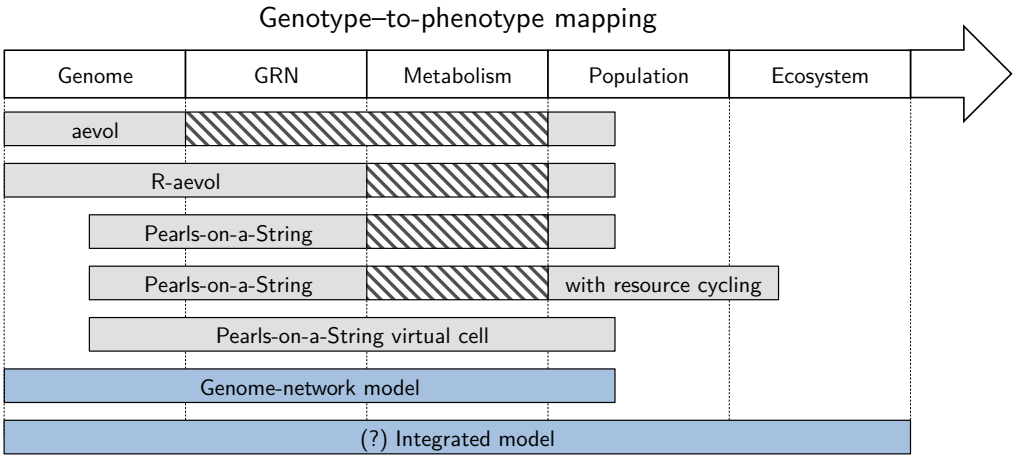

Figure 1 - Models developed by INRIA and UU cover multiple levels of organization. Aevol models the evolution of genome structure. R-aevol also permits to evolve genetic regulation networks (GRN). "Pearls-on-a-string" models have a simpler genome representation, but have been used by UU to cover several levels of organization. Genome-network model and integrated model are WP2 tasks.

As a first step on the road of the integrated model (final objective of WP2), the task 2.1 offers to exploit INRIA and UU knowhow to develop a model merging realistic genetic structure and mutational operators (*aevol*), and efficient network formalisms (*PoaS*).

1.1. Overview of *aevol* and *PoaS* formalisms

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, double-strand chromosome that is actually a string of binary nucleotides, 0 being complementary of 1 and reciprocally (figure 2). 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 the arbitrary set of functions organisms must 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 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 chromosomic rearrangements: duplications, large deletions, inversions, and 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... (See also http://www.aevol.fr/ for details).

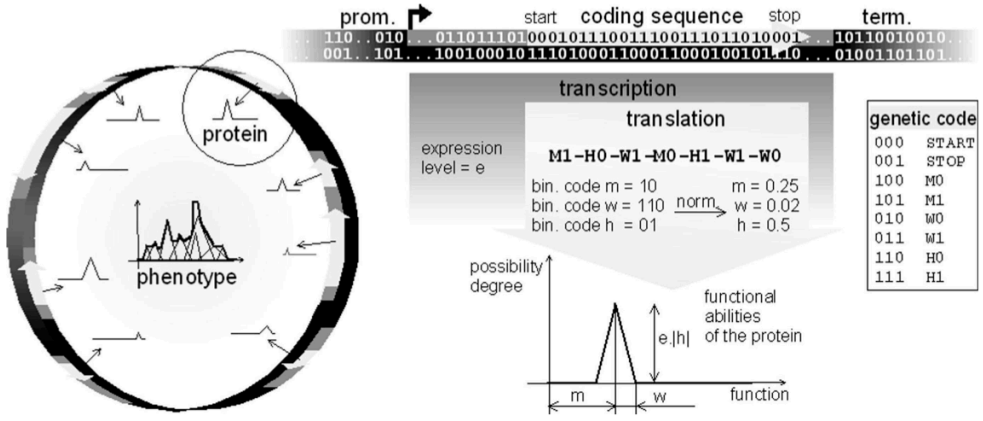

Figure 2 - Overview of the aevol model (from Knibbe, 2007a).

Standard *PoaS* model is an individual-based simulation with a population on a lattice subjected to an environment that changes over time (figure 3). Using a lattice enables a computationally efficient method for competition among individuals and is biologically sound, as organisms virtually always live in a spatial system with a certain degree of locality. Each individual owns a linear chromosome containing *n* different genes and on average two binding sites per gene. The network is derived from this chromosome with genes as nodes and interactions between genes defined by which gene binds to which binding site. The fitness of an individual is defined on the level of network states, i.e. which genes are activated or inhibited. Reproduction of an individual is based on this fitness. The network consists of genes with interactions among them. A gene has a state of expression *s* (on = 1, off = 0), a threshold $\Theta \in \{-2,-1,0,1,2\}$ and an identification tag $t \in \{0,1,2,...,n\}$. Binding sites specify which gene may bind to them via their own identification tag (i.e. if tags are

equal), which is called the binding preference. They also determine the type of interaction *w*: *activation* (*w* = 1) or *inhibition* (*w* = -1). If there are multiple copies of a binding site present in the upstream region of a gene, there will be parallel edges in the resulting network. Symmetrically, if there are multiple copies of a gene, they all bind to the same binding sites (Crombach & Hogeweg, 2008). UU have developed several versions of *PoaS*, including interactive evolution with transposons, micro RNAs regulation, and resource cycling.

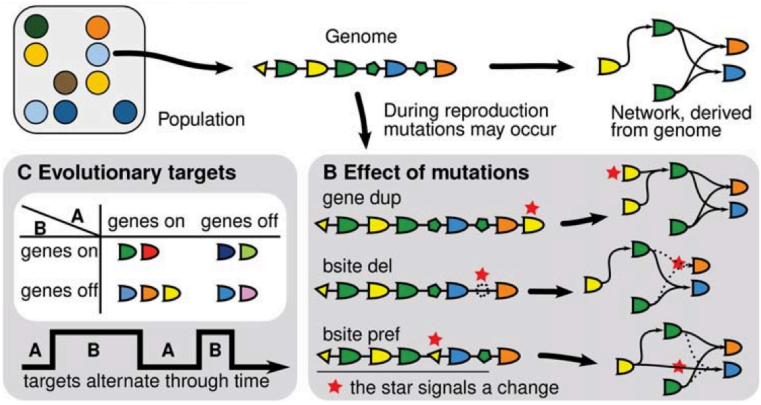

Figure 3 - Overview of PoaS model (from Crombach & Hogeweg, 2008). (A) Simulations are run on a 150x50 lattice for 6.105 time steps. The lattice harbours a population of genomes, where a genome is a linear chromosome of genes with binding sites. A Boolean threshold network is built from each genome. During each time step the network may update the expression level of the genes for 11 propagation steps. (B) The impact of several gene and binding site mutations is shown. The change in the genome and network topology is signalled by a red star. In a typical simulation the parameters are (per gene, binding site): gene duplication (dup) 2.10^{-4}, deletion 3.10^{-4}, threshold 5.10^{-6}, binding site (bsite) duplication 2.10^{-5}, innovation 1.10^{-5}, deletion (del) 3.10^{-5}, preference (pref) 2.10^{-5} and weight 2.10^{-5}. (C) Typically the environment changes over time with a probability of $\lambda = 3.10^{-4}$. The two evolutionary targets A and B determine which genes should be expressed (on) or inhibited (off). The result is four categories of genes; some should be always on, some should toggle their expression state and some should never be expressed. In a typical simulation, the target expression states are, from gene 0 to 19, A: 00011 11000 00000 11111 and B: 11010 01001 01100 01011.

1.2. A common formalism: the *n*-tuples bag

Our first objective is to develop a modelling formalism that merges *aevol* and *PoaS* principles in an integrated "genome-network" model. To this aim we first extract the principle shared by those two models.

Whatever the formalism used to store information in the genomic data-structure, both *aevol* and *PoaS* models extract a **set of *n*-tuples** from it. This **mapping** can be seen as a projection (potentially complex and non-linear) of the genome space on the *n*-tuples *n*-dimensional space (figure 4). This set permits to build the phenotype in a specified phenotypic space (a fuzzy set in *aevol*, a gene network in *PoaS* models).

A *n*-tuple is an ordered list $(x_1, x_2, ..., x_n) : T_1 \times T_2 \times ... \times T_n$, with T_i the type of x_i, for example T_1=integer, T_2=real, T_3=integer, etc. Projection operators are defined to project a genomic subsequence on the *n*-dimensional space $T_1 \times T_2 \times ... \times T_n$ and get a *n*-tuple. The projection of the whole genome on the *n*-dimensional space $T_1 \times T_2 \times ... \times T_n$ gives a set of several *n*-tuples. One

can use any formalism to implement the genomic data-structure, providing it is possible to project it in the space $T_1 \times T_2 \times ... \times T_n$.

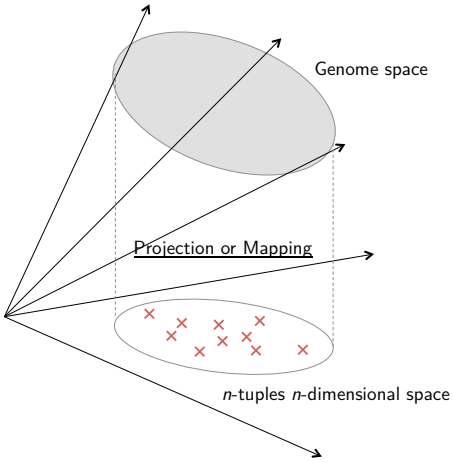

Figure 4 - A set of *n*-tuples is extracted from the genomic data-structure by projecting genomic subsequences on the *n*-tuples *n*-dimensional space. Projection on the *n*-tuples space can be a complex and non-linear process, and is usually called a mapping.

For example, *aevol* uses an artificial genetic code to extract several triplets of real numbers (x_1, x_2, x_3) : $\mathbb{R}^3$ from the genomic binary sequence. This is done in two steps modelling transcription and translation. The mapping from the DNA binary sequence to the set of triplets is non-linear. In *PoaS* models, *n*-tuples are represented as is (one-to-one mapping), in a sequence of "pearls" on a single-strand genome (thus there is actually no complex mapping at this stage of the genome decoding process, each “pearl” being directly a *n*-tuple).

In both models, each *n*-tuple corresponds to a gene, or possibly a part of a gene, a promoter, a binding site... In both models too, the genes are ordered on the genomes. On the contrary, the set of *n*-tuples $\{(x_{11}, x_{12}, ..., x_{1n}), ..., (x_{n1}, x_{n2}, ..., x_{nn})\}$ is generally unordered ("bag" of *n*-tuples). This implies that gene order does not affect the phenotype and that inversions can modify gene order without impairing fitness. Such neutral modifications of gene order can, however, affect the evolvability of the lineage, because mutations like large deletions or duplications may affect several genes at once.

A very noticeable fact in both *aevol* and *PoaS* models is that, thanks to realistic mutational operators (point mutations but also small insertions and deletions, large duplications, large deletions, translocations or inversions), the number of *n*-tuples encoded in the genome is thus variable, both within the population (inter-individual variation) and in time. Few *in silico* evolution models allow for this variation, while this is a key feature of the genotype-to-phenotype mapping if one wants to observe and study EvoEvo.

2. Proposal for the genome-network model

Merging INRIA and UU models should permit us to find out a formalism to represent a simplified metabolic network. A very first step is to develop an **artificial chemistry** (AChem), as a basis for metabolic chemical reactions. To this aim, we shall keep the n-tuples formalism.

2.1. Basic concepts on cell metabolism

Cell metabolism is the set of life-sustaining chemical reactions within the cells. These enzyme-catalysed reactions allow organisms **to grow and reproduce**, **maintain their structures**, and **respond to their environments**. Metabolism is usually divided into two categories. **Catabolism**, that breaks down organic matter and harvests energy by way of cellular respiration, and **anabolism** that uses energy to construct components of cells such as proteins and nucleic acids. Then, a metabolic network is the complete set of chemical reactions within the cell. As such, these networks comprise the **metabolic pathways**, as well as the **regulatory interactions** that regulate these reactions, through control of enzyme production.

2.2. Basic concepts on artificial chemistries

> « *An artificial chemistry is a man-made system that is similar to a real chemical system.* »
>
> P. Dittrich et al. (2001)

An artificial chemistry (AChem) can be defined as a triple (S, R, A), where S is the set of all possible molecules, R is a set of reaction rules representing the interaction among the molecules, and A is an algorithm describing the reaction vessel or domain and how the rules are applied to the molecules inside the vessel (Dittrich et al., 2001).

The set of molecules $S = \{s_1, s_2, ..., s_n\}$ can potentially be infinite. A reaction rule $r \in R$ is a chemical equation $s_1 + s_2 + ... + s_i \rightarrow s'_1 + s'_2 + ... + s'_j$, with the reactants (or the substrates) on the left side, and the products on the right side. *i* is the order of the reaction. The set of reaction rules R can be defined explicitly (all possible reactions *r* are defined and are in finite number), or implicitly. Notice that here, stoichiometry is 1 for all reactants, but there is no constraint on this point.

The algorithm A is applied on an instance of the triplet (S, R, A), that is, a collection P of molecules. The set of chemical equations R can be solved with stochastic or deterministic methods, possibly adding spatial rules.

2.3. Defining an artificial chemistry with the *n*-tuples formalism

Since each *n*-tuple is the final product of a genomic subsequence (a gene in the simplest case), two options appear:

Option (i) - A tuple represents the reaction rule itself; in this case, the genome defines the set of reactions rules R_{cell} in the cell. For example, let us consider a *n*-tuple $(x_1, x_2, ..., x_i, x_{i+1}, x_{i+2}, ..., x_n)$ with *n* an even number. This tuple could represent the chemical equation of order *n/2*:

$$s_1 + s_2 + ... + s_{n/2} \rightarrow s_{n/2+1} + s_{n/2+2} + ... + s_n,$$

with $x_i \equiv s_i$, $i \in \{0,1,2,...,n\}$.

Additional elements in the tuple could also define the reaction rate constant and the stoichiometry. In this view a gene/tuple represents en enzyme that activates a reaction.

Option (ii) - A tuple represents a chemical species, which is potentially a reactant for a subset of reactions in R; in this case, R is defined once for all the cells, a reaction being possible if its reactants are present in the cell. For example, if the set of possible reactions R contains the subset of equations (1) $s_i + s_j \longleftrightarrow s_i.s_j$ and (2) $s_i.s_j \rightarrow s_k + s_j$ (s_i, s_j, $s_k \in S$, and "." symbolizing a chemical bond), the 1-tuple (x) with $x \equiv s_j$ catalyses an the enzymatic reaction $s_i + s_j \longleftrightarrow s_i.s_j \rightarrow s_k + s_j$. To this aim, a pair ($x$, c) can also being used, with c the concentration of x, and so forth. Indeed, in this option a gene can produce an unused compound, if this one is not present as a reactant in R. One can see option **(i)** as a special case of option **(ii)**, where the n-dimensional space of the tuples implicitly describes all possible reactions rules R.

Using the tuple formalism actually opens a large set of possibilities to define an artificial chemistry as complex and realistic as required. Taken together, the realistic genome structure and mutational operators, the n-tuples formalism, and the possibility to code an artificial chemistry with it, gives us a general framework to develop the genome-network model (figure 5).

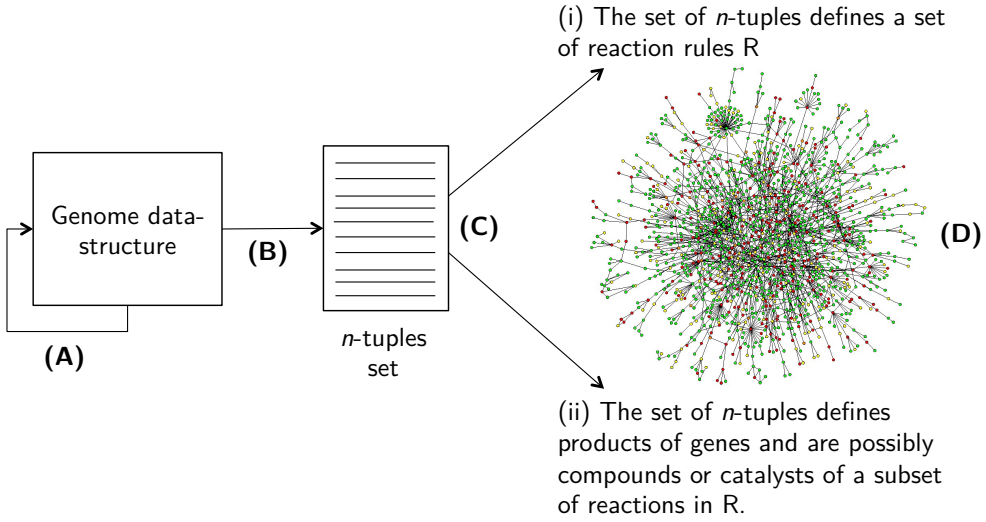

Figure 5 - A general framework for the genome-network model. (A) At each replication, the genome data-structure is subjected to mutations (point mutations, large rearrangements, recombination, horizontal transfer,). (B) A mapping, assimilated as transcription and translation processes, gives a set of n-tuples. (C) Depending on modelling choices, the set of n-tuples defines: (i) the set of reactions rules R_{cell} of the artificial chemistry in each cell, or (ii) genes products (proteins, catalysts...) involved or not in a subset of reactions belonging to R. (D) Since reactions rules are defined (edges of the network), nodes can potentially be metabolites (internal or external ones), binding sites, RNAs and so forth.

3. Instance of the genome-network model: deliverable 2.1)

From the general framework presented above, we instantiated an individual-based model integrating a realistic genome structure and mutational operators, as well as a metabolic network. The model does not integrate genetic regulation nor transcription. These aspects will be treated in task 2.2 of WP2.

3.1. Overview of the genome structure

Here, we simplified the genome structure (compared to *aevol* for example), to cope with computational load, in the perspective of an integrated model, since we have to maintain the model complexity low enough to enable its practical use.

Each individual owns a circular genome made of pearls representing either functional genes or non-coding regions. Each gene codes for a triplet (x_1, x_2, x_3) : $\mathbb{N}\times\mathbb{N}\times\mathbb{R}$ (meaning types: x_1: *integer*, x_2: *integer*, x_3: *real*). At each replication, the genome endergoes mutations (point mutations, large duplications, large deletions, translocations, inversions). Mutation rates are expressed for each pearl for each generation ($mutation.pearl^{-1}.generation^{-1}$). A mutation can have huge deleterious effects on a gene, and transform it into a pseudogene that is a non-coding pearl (figure 6). Thus, the amount of non-coding region the genome contains can also evolve.

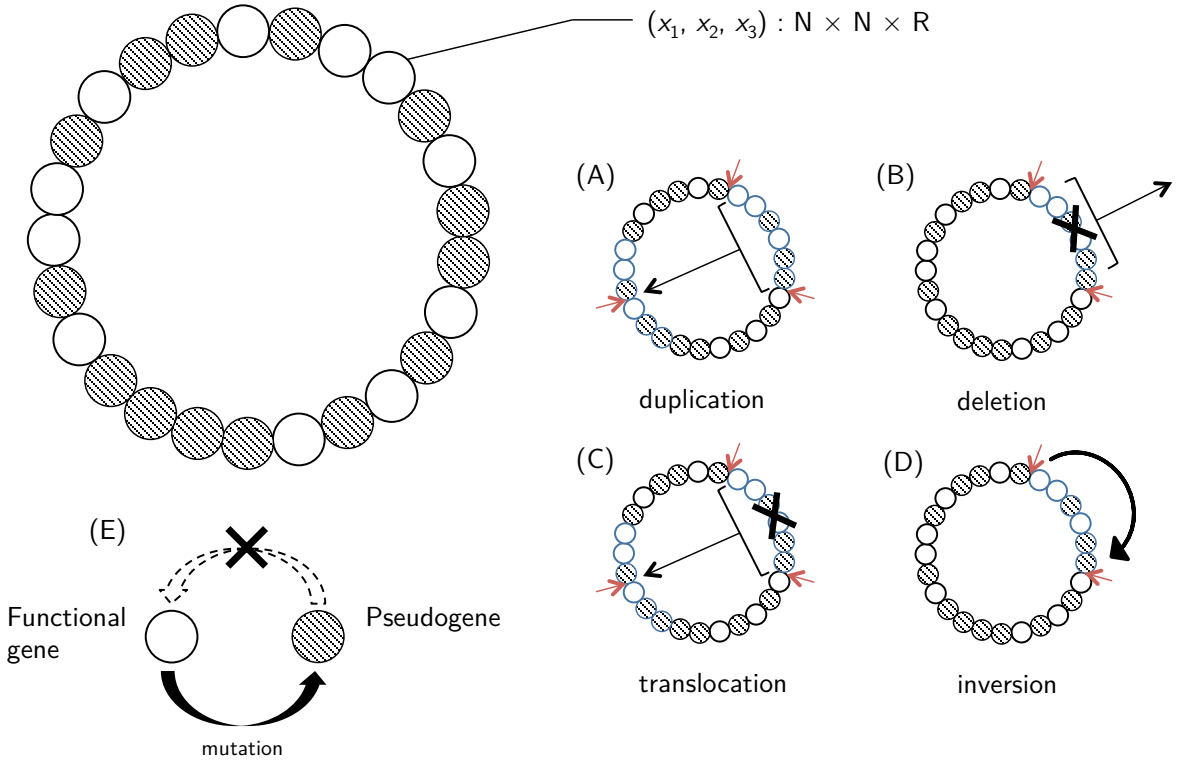

Figure 6 - Overview of the genome structure. Genome is a circular single-strand sequence of pearls, each coding for a triplet (x1, x2, x3) : $\mathbb{N}\times\mathbb{N}\times\mathbb{R}$. At each replication, the genome is subjected to mutational operators: point mutations (not shown here), (A) large duplications, (B) large deletions, (C) translocations, (D) inversions. Red arrows symbolize breakpoints in the sequence. (E) point mutations and breakpoints can have huge deleterious effects and transform a functional gene in a pseudogene (considered as non-coding region).

Point mutations (substitutions, indels) on real DNA sequences have non linear effects. Indeed, one point mutation can have huge deleterious effects on a gene, since it can potentially loose its functionality and become a pseudogene. To represent this in the model, a point mutation within a gene can either turn it into a pseudogene or move its triplet in the triplet space such that (x_1, x_2, x_3)$_{mutated}$ = (x_1, x_2, x_3) × M, with M the diagonal matrix representing mutational effects on each element of the triplet (figure 7). Such formalism permits to integrate non-linear effects of point mutations (substitutions, indels) on real DNA sequences. Indeed, one point mutation can have

huge deleterious effects on a gene, since it can potentially loose its functionality and become a pseudogene.

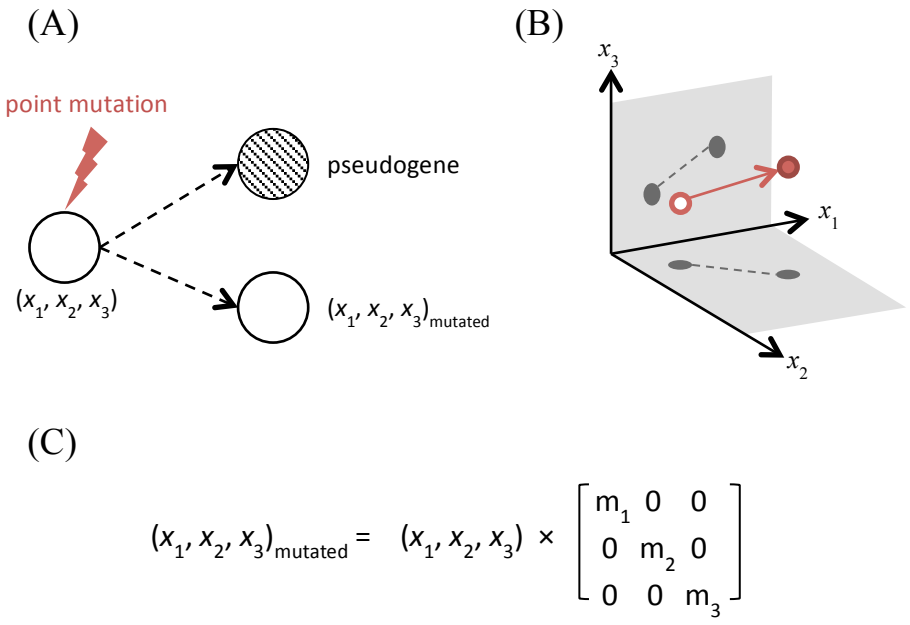

Figure 7 - Effects of a point mutation on a gene. (A) Due to potential deleterious effects of a point mutation (substitution, indels) on a real nucleotidic sequence, a point mutation can transform a functional gene in pseudogene. (B) If the gene is not destroyed, a point mutation can be seen as a transition in the triplet space (white dot before the transition, red dot after it), (C) such that $(x1, x2, x3)_{mutated} = (x1, x2, x3) \times M$, with M a diagonal matrix.

3.2. Overview of the artificial chemistry and the metabolic network

Each pearl of the genome represents a gene or a pseudogene. Following option (i) described in section 2.3, all active genes produce an implicit enzyme catalysing reactions as follows: each triplet describes a chemical reaction $s \rightarrow p$, such as $x_1 \equiv s$ (the substrate, or reactant), $x_2 \equiv p$ (the product). The third value x_3 is the Michaelis constant K_M of enzymes implicitly produced by the gene. If $K_M < 0$, the reaction is reversed. Thus, the genome defines the set of reactions rules R_{cell} to be potentially performed in the cell, each cell carrying its own set of reactions. Moreover, the whole triplet space $\mathbb{N}\times\mathbb{N}\times\mathbb{R}$ can be seen as an implicit definition of the set of reactions $R_{universe}$ reachable by any cell. Note that $R_{universe}$ is infinite. However, each cell will always contain a finite reaction network, since genome size will always be finite.

The model does not explicitly represent proteins regulation and concentrations. However, If we assume that our system is in compatible with Michaëlis-Menten kinetics (i.e. $d[ES]/dt = 0$ and $[ES] \rightarrow [E] + [P]$ non-reversible) and $[E] \ll [P]$, we can use Michaëlis-Menten equations to simulate enzymatic reactions in the metabolic network. For example, if a gene produces an enzyme catalysing the reaction $s \rightarrow p$, evolution of concentrations will be described by the ordinary differential equation (ODE):

$$\frac{d[p]}{dt} = \frac{(V_{max}\times[S])}{(K_m + [S])}$$

V_{max} is fixed (e.g. $V_{max} = 1$) and K_m is a parameter of the enzyme.

A cell can also interact with its environment by eating or excreting metabolites, thanks to special enzymes called protein pumps. A gene codes for a pump if the enzymatic reaction has the form $s \rightarrow s$ (i.e. $s = p$). If $K_m > 0$, the protein pumps the metabolite s into the cell ($s_{ext} \rightarrow s_{int}$), while if $K_m < 0$, the protein pumps the metabolite s out of the cell ($s_{int} \rightarrow s_{ext}$). Reaction rates are computed as previously.

Hence, for each cell, the whole genome produces enzymes defining a metabolic network, where nodes are species involved in chemical reactions, and edges are enzymatic reactions encoded in the genome. Reaction rates are computed following Michaëlis-Menten equations (figure 8).

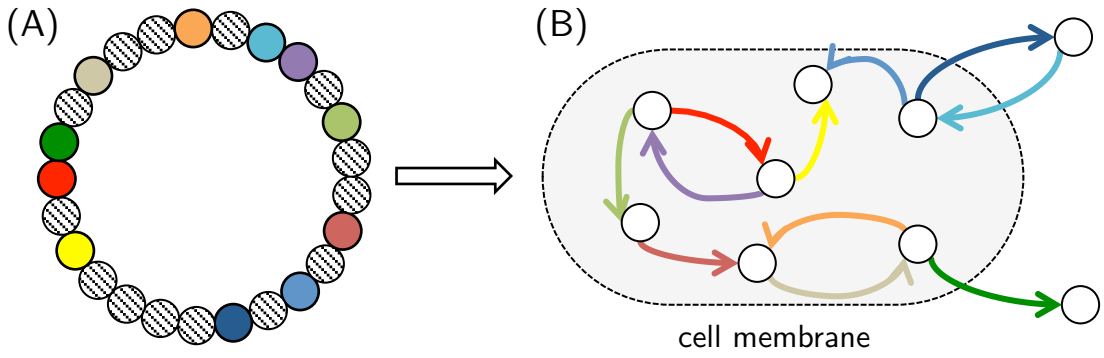

Figure 8 - The genome codes for a metabolic network. (A) Each active gene produces enzymes catalysing reactions of the form s → p, where enzyme activity is implicit. (B) The set of enzymes produced by the genome defines a metabolic network, with metabolic pathways, cycles and environmental exchanges.

When a cell divides itself, it shares its metabolic content with its daughter. Then, each new cell owns a state vector $X = \{x_1, x_2, ..., x_n\}$ of all present species and all potentially reachable ones during cell life. The set of reactions $R = \{r_1, r_2, ..., r_m\}$, the reaction rate vector $v = (v_1, v_2, ..., v_m)^T$ and the stoichiometry matrix S enable us to characterize the system equations:

$$\frac{dX_i}{dt} = \sum_j S_{ij}.v_j$$

for $i = 1, ..., n$, with X_i the concentration of the ith metabolite, v_j the reaction rate of the jth reaction and S_{ij} the stoichiometric coefficient of ith metabolite in the jth reaction.

This system of equations can be solved by several methods, stochastic or deterministic, discrete or continuous (see Gillespie, 2013, for an overview). Here, we chose to solve reaction rate equations (RRE) and to simulate the dynamic numerically.

To sum up, we defined an artificial chemistry {S, R, A} where:

- The set of molecules S is the integer space $\mathbb{N}$,
- Each cell carries its own set of reaction rules R_{cell}, defined by its genome. The triplet space $\mathbb{N}\times\mathbb{N}\times\mathbb{R}$ defining the whole (and infinite) set of reactions $R_{universe}$ reachable by any cell,
- For each network, the system of ordinary differential equations are integrated using a continuous and deterministic method.

3.3. Overview of the evolutionary process

Although the evolutionary process is not directly covered by this deliverable, interesting possibilities are worth noting for further development.

Individuals evolve on a lattice and share a common environment. Population and environment are initialized with certain characteristics (genome, species amount, population size), and **evolve together**.

In previous models, population size was fixed and cycles of mutation-selection were performed, either by renewing population with biased reproduction rates depending on relative fitness (generational model), either by killing some individuals at random and replacing them by a competitive reproduction depending on relative fitness and space (steady-state model). At each cycle, fitness was explicitly computed for all individuals, depending on some criteria: environmental target (function fitting, Boolean network state), functional target (homeostasis, capacity to recycle environment), and so on.

Here, **population size is variable**, **reproduction is asynchronous** and fitness is implicit.

Population size depends on population growth rate and environmental carrying capacity. Thus, simulations can possibly lead to oscillations or population extinction. Cell division depends on individual variables, and competition between individuals implicitly appears due to a common sharing of environmental resources and space. Here, environment and population are strongly linked since individuals constantly modify their surrounding by eating, transforming and excreting material, or dying. The environment is thus constantly modified; possibly resulting in the creation of new evolutionary niches in which new species can emerge, eventually creating a complex ecosystem and a complex trophic network.

Because cell division depends on individual criteria, no explicit fitness computation is needed. To survive and reproduce, cells have **to grow**, **maintain their structures**, and **respond to their environment**. Individual growth rate could be evaluated by individual biomass, structure maintenance or by measuring cell's energy production or metabolic fluxes. Interaction and detection of metabolic concentrations in the local environment is an obvious direction of the evolution.

The lattice allows for spatial interactions among individuals, and the possibility to model metabolite diffusion in the environment.

4. Note on the model development

Depending on modelling choices, one could prefer one formalism or another to model the genome data-structure (or even develop its own model). But at least, it should be possible to copy the genome, mutate it and extract from it a set of tuples.

Consequently, it will be interesting develop an **application programming interface** (API), to specify how the genome should interact with other components of the model (figure 9). This approach can be used more generally in the model development (e.g. to connect the metabolite model to different solvers). This question will be discussed later on with all the partners involved in software development.

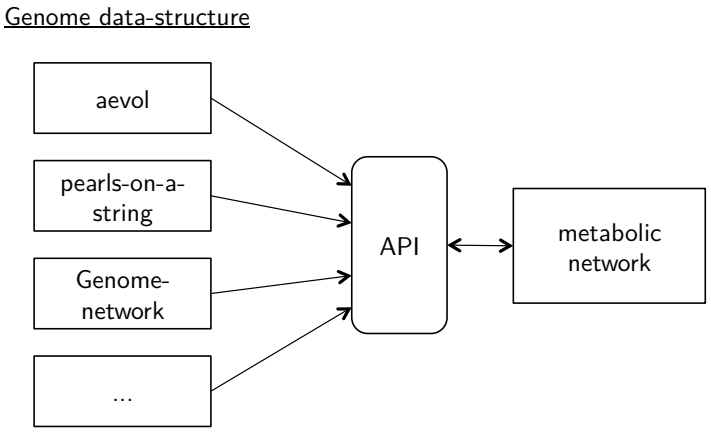

Figure 9 - Usage of APIs in the genome-network model. Depending on modelling choices, one could prefer one formalism or another to model the genome data-structure (or even develop its own model). Developing an application programming interface (API) will permit anyone to choose the right formalism or write it on purpose.

5. Conclusion

Merging INRIA and UU know-how allowed us to specify a genome-network model that will permit us to investigate the feasibility of integrating realistic genome structure and metabolic network. To date, a runnable prototype has been implemented and has provided some preliminary results.

However, developing a biological model for research purpose is just not the same as developing an application for a customer. Specifications can change over time, owing to experimental results, to parametric exploration, interaction with biologists and new insights in the model. Thus, the overview presented above may change slightly during model development and test. For example, we are currently working on a more complex artificial chemistry, allowing for chemical bonds and the existence of macromolecules. As presented in section 2.3, we should explore the feasibility of the different artificial chemistry paradigms, and this will possibly modify some of the model specifications. In such a case, new versions of the present deliverable will be written.

6. References

[Beslon *et al.,* 2010a] G. Beslon, D. P. Parsons, Y. Sanchez-Dehesa, J.-M. Peña, C. Knibbe, 2010, Scaling laws in bacterial genomes: A side-effect of selection of mutational robustness? *BioSystems,* 102:32-40.

[Beslon *et al.,* 2010b] G. Beslon, D. P. Parsons, J.-M. Peña, C. Rigotti, Y. Sanchez-Dehesa, 2010, From digital genetics to knowledge discovery: Perspectives in genetic network understanding. *Intelligent Data Analysis journal (IDAj)* 14:173-191.

[Crombach & Hogeweg, 2008] A. Crombach, P. Hogeweg, 2008, Evolution of evolvability in gene regulatory networks, *PLoS Computational Biology,* 4(7): e1000112.

[Dittrich *et al.,* 2001] P. Dittrich, J. Ziegler, W. Banzhaf, 2001, Artificial Chemistries - A Review, *Artificial Life,* 7(3):225-275.

[Knibbe *et al.,* 2007a] C. Knibbe, O. Mazet, F. Chaudier, J.-M. Fayard, G. Beslon, 2007, Evolutionary coupling between the deleteriousness of gene mutations and the amount of non-coding sequences, *Journal of Theoretical Biology,* 244(4):621-630.

[Knibbe *et al.,* 2007b] C. Knibbe, A. Coulon, O. Mazet, J.-M. Fayard, G. Beslon, 2007, A long-term evolutionary pressure on the amount of noncoding DNA, *Molecular Biology and Evolution,* 24(10):2344-2353.