# EvoEvo Deliverable 6.8

Guillaume Beslon, Jonas Abernot, Santiago F. Elena, Dominique Schneider, Paulien Hogeweg, Susan Stepney, Christophe Rigotti


# *EvoEvo Deliverable 6.8*

# *Final report*

Due date: M36 (October 2016)
Person in charge: Guillaume Beslon
Partner in charge: INRIA
Workpackage: WP6 (Project management)
Deliverable description: Final report: Final report of the EvoEvo project.

Revisions:

| Revision no. | Revision description | Date | Person in charge |
|---|---|---|---|
| 1.0 | Main structure of the document | 22/09/16 | G. Beslon (INRIA) |
| 1.1 | Section 2.2 | 21/10/16 | S. Elena (CSIC)<br>D. Schneider (UGA) |
| 1.2 | Section 3 | 24/10/16 | P. Hogeweg (UU) |
| 1.3 | Section 2.4 | 27/10/16 | P. Hogeweg (UU) |
| 1.4 | Section 2.10 | 7/11/16 | S. Stepney (UoY) |
| 1.5 | Section 2.3 | 14/11/16 | G. Beslon (INRIA) |
| 1.6 | Introduction of section 2, conclusion of section 2.2, project management | 15/11/16 | G. Beslon (Inria) |
| 1.7 | Section 2.6 | 17/11/16 | C. Rigotti (INRIA) |
| 1.8 | Section 4.3 | 18/11/16 | J. Abernot (Inria) |
| 1.9 | Management and list of deliverables | 22/11/16 | G. Beslon (Inria) |
| 1.10 | General introduction | 23/11/16 | G. Beslon (Inria) |
| 2.0 | Report completed | 24/11/16 | G. Beslon (Inria) |

Revisions (continued).

| | | | |
|---|---|---|---|
| 2.1 | Correction on bibliography | 24/11/16 | S. Stepney (UoY)<br>P. Hogeweg (UU)<br>G. Beslon (Inria) |
| 2.2 | Proofreading | 25/11/16 | S. Stepney (UoY) |
| 2.3 | Proofreading | 25/11/16 | S. Elena (CSIC) |
| 2.4 | Correction and reference update | 27/11/16 | P. Hogeweg (UU)<br>G. Beslon (Inria) |

# Table of Contents

## Publishable summary

Evolution is the major source of biological complexity on Earth, being at the origin of all the species we can observe, interact with or breed. On a smaller scale, evolution is at the heart of the adaptation process for many species, in particular microorganisms (*e.g.*, bacteria, viruses…). Microbial evolution results in the emergence of the species itself, and it also contributes to the microorganisms' adaptation to perturbations or environmental changes. These organisms are not only shaped by evolution, they are also build to evolve. In other words, they have evolved their evolutionary capabilities, a process we call "evolution of evolution" or EvoEvo.

The EvoEvo project has used this "evolution of evolution" property to develop new evolutionary approaches in information science; our ultimate goal being to address open-ended problems, where the specifications are either unknown or too complicated to be expressed, and to produce software able to operate in unpredictable, varying conditions. To achieve this ambitious goal, the project organized a 4-steps transfer of knowledge from life sciences to information technology:

1. Since the processes by which evolution evolves ("EvoEvo strategies") are not fully understood, we started from experimental observations of microorganism evolution in order to observe, quantify and characterize the EvoEvo strategies in microorganisms at the level of genomes, biological networks and populations (WP1). This has been achieved through experimental evolution and bioinformatics in order to gain a better understanding of this phenomenon and contribute evolutionary theory by allowing understanding of the surprisingly high pace of evolution of microorganisms.
2. These EvoEvo strategies have been simulated in a computational framework. The simulations use individual-based models to help analyse the results of the evolution experiments (WP2). They have helped us to propose hypotheses on the structural roots of EvoEvo at the levels of genetic sequences, regulation and metabolic networks, and cell populations (WP3). These models also constitute the basis of the computational evolutionary platform. EvoEvo has thus contributed to computational biology by the development of integrated computational evolutionary models that are available for the scientific community.
3. The computational models developed in WP2 have been used to design a computational evolution platform to exploit evoevo in application software (WP4). This platform is directly inspired from the *in silico* models, but simplifications and generalizations have been made. The former remove from the models all the biological specificities that are not useful to exploit EvoEvo. The later enable the framework to be used in different application contexts. EvoEvo has thus contributed to evolutionary computation by the development of a new framework that uses evolution of evolution at its heart.
4. We have applied EvoEvo to real ICT problems (WP5). Two applications of increasing difficulty have been explored: subspace clustering of WiFi signals ("EvoWave"), and a musical personal companion that follows a dancer and learns to play music according to the dancer's moves ("EvoMove"). The ability to effectively exploit EvoEvo in these applications constitutes the final proof of concept that evolution of evolution can drive future technologies in an efficient way.

The EvoEvo project impacts ICT, through the development of new technologies. It also impacts biology and public health, by providing a better understanding of microorganism adaptation (such as the emergence of new pathogens or the development of antibiotic resistances).

# 1. Introduction

EvoEvo ("Evolution of Evolution") was a three year FP7-ICT project (call ICT-2013.9.6 - FET Proactive: Evolving Living Technologies - EVLIT). This final report covers the three years of the EvoEvo project, from November 1st 2013 (beginning of the project) to October 31th 2016. The EvoEvo consortium is composed of five partners: INRIA (France), UGA (formerly UJF, France), Utrecht University (The Netherlands), University of York (UK), and CSIC (Spain). EvoEvo aims were to study the "Evolution of Evolution" processes (*i.e.*, the processes by which evolution modifies itself and influences its own pace) and to exploit these processes in ICT applications in order to develop new "living" technologies.

EvoEvo achieved its objectives through a work plan containing 6 workpackages: WP6 management; WP1 *in vivo* experiments; WP2 model design; WP3 *in silico* experiments; WP4 computational framework design; WP5 applications. More precisely, the EvoEvo workplan organization is based on three principles that guarantee that the biological foundations of EvoEvo are effectively and efficiently transmitted to the computation application through modelling and framework development steps. The three principles are the following:

- A route from evolutionary biology (WP1) to artificial evolution (WP4) through modelling (WP2).
- Parallelism between *in vivo* experimental evolution (WP1) and *in silico* experimental evolution (WP3).
- Applicative targets (WP5) that will make profit from both the computational framework designed in WP4 and from EvoEvo knowledge produced in WP3.

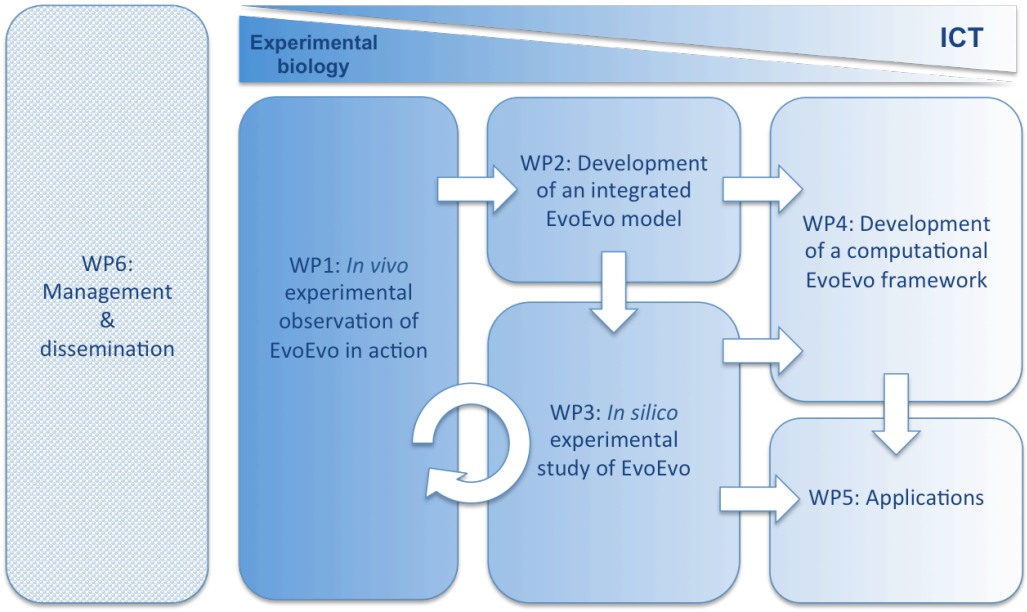


**Figure 1 –** Organization of the EvoEvo project. Arrows indicate dependencies between the workpackages. As the project will go on, it progressed from left to right and from top to bottom, i.e. from experimental biology to ICT applications.

This report describes the work done and the results gathered in all the aspects of the project, from life science to ICT applications. It is organized in three parts. In the first part (section 2, Description of the work), the work accomplished during the project is described one workpackage after another. This first part is complementary to the deliverables and publications produced during the project; the detailed information provided in those different deliverables is not repeated here. Pointers to the corresponding deliverables and publications are provided when necessary. The next two sections (section 3, General insights on the EvoEvo process; section 4, General insights on living technologies) focus specifically on the two central concepts of the project: EvoEvo and living technologies. These sections do not correspond to a specific WP or to a specific deliverable, but rather to general insights that the whole project has revealed on both concepts.

# 2. Description of the work

## 2.1. Introduction

This section presents the results of the project for each workpackage. For each WP the presentation of the results is organized by “contributions”, which sometimes diverge from the initial structure of the project (in which each WP was composed of “tasks”). This enables us to better emphasize the results. Note that this section does not replace the detailed description of the work presented in the project deliverables (available on the project website – see http://www.evoevo.eu/deliverables/ and section 2.7.2), nor the detailed results presented in the project publications (see http://www.evoevo.eu/publications/ and section 2.7.3)

## 2.2. Workpackage 1: Experimental observation of EvoEvo in action

### 2.2.1. Introduction

WP1 explored EvoEvo properties in two microorganisms (the model bacterium *Escherichia coli* and the RNA virus tobacco etch Potyvirus, TEV). Both organisms evolve at a high pace but their molecular structures are very different. Identification of traits that confer them their high evolutionary potential was the objective of WP1. These traits fed the computational models and were tested in the computational experiments, thus constituting the first step in the biological-to-application scheme of EvoEvo (Figure 1).

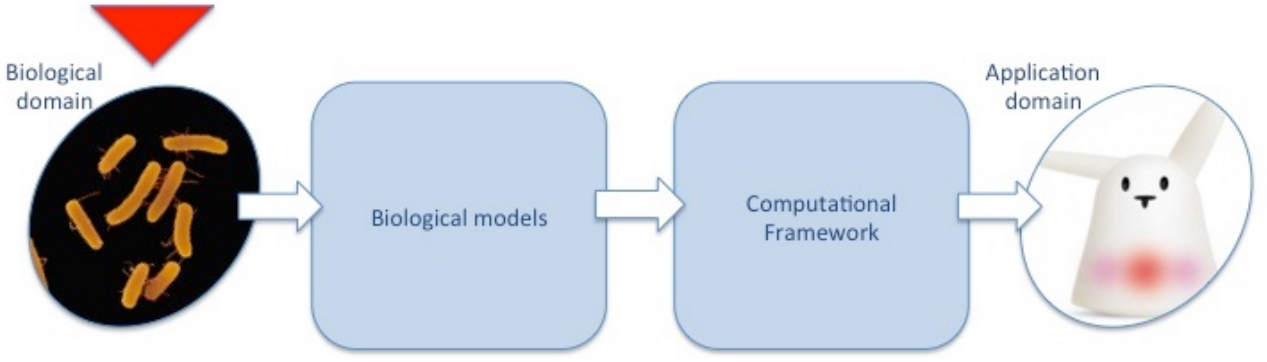


Figure 2: Position of WP1 in the biology-to-application scheme of EvoEvo

In WP1, we addressed experimentally the pace of evolution of microorganisms and related it to their robustness (task 1.1), evolvability (task 1.2) and open-endedness (task 1.3). In particular, we addressed the relationship between robustness and evolvability by directly testing whether more robust genotypes are also more evolvable or, by contrast, whether they adapt in a slower pace to new environmental conditions. We tackled these issues using *in vivo* experimental evolution, which consists in propagating living organisms for hundreds to tens of thousands of generations in defined environments. It provides a powerful methodology to analyse the molecular basis of adaptation and to draw a rigorous phenotype-to-genotype map. We used two different experimental systems, an RNA virus (TEV) and a bacterium (*E. coli*), both of which have become classic models in experimental evolution that allow detailed genetic manipulations and analyses. Furthermore, given their short generation times, large population sizes and, in the case of the RNA virus high mutation rates, relevant evolutionary changes take place after short periods of time, allowing us to observe evolution in action.

### 2.2.2. Contribution 1 (Task 1.1): Robustness at the population, regulatory network and genome levels

#### Section 1: Robustness at the population level in the TEV experimental system

Robustness at the population level was investigated by generating a collection of TEV strains and evolved populations that differed in their gene order and content (deliverable D1.1). Some of these engineered genomes contain additional genes (increased complexity), some containing fewer genes (reduced complexity), some containing multiple copies of the same gene (functional and genetic redundancy), and some containing additional genes that are functionally redundant with native genes (functional redundancy). All these engineered genotypes have been evolved under controlled environmental conditions (*i.e.*, host species, effective population size and mutation rates) and their fitness and robustness evaluated. Four experimental treatments were performed (described in Task 1.1 of Section 1 of the proposal) combining small and large population sizes with low and high mutation rates in a factorial manner. These experiments were performed in the reservoir host, *Nicotiana tabacum*. Multiple independent lineages were evolved under each environmental condition. Details are provided in deliverable D1.2.

We found that both main factors effective population size and mutation rate had significant effects on the robustness of the evolved lineages. Population size itself had a significant effect on robustness, with populations evolved at large effective population size being more robust than populations evolved under small effective population sizes. By contrast, evolving at high mutation rate had no net significant effect on robustness. However, interestingly, the effect of mutation rate was strongly dependent on the effective population size: viral populations increased robustness in the combination of high mutation rate and small effective population size; in contrast viral populations became more sensitive to mutational load when evolved at large effective population size and high mutation rates. Finally, independent lineages evolved under the same conditions showed differences in their robustness, as a result of the random effect of adaptive mutations fixed by each lineage.

In conclusion, we found that the theoretical predictions made by Krakauer and Plotkin (2002) for the evolution of population-level robustness were partially fulfilled for TEV: viral populations evolved under conditions of large effective population size become more robust than populations

evolved at small effective population sizes. However, the hypothesis of a positive synergism between mutation rate and effective population size was not fulfilled in our experimental system.

**Section 2: Robustness at the regulatory network level in the *E. coli* experimental model**

Robustness at the regulatory level was investigated by deleting the *crp* gene, encoding one of the biggest regulatory proteins in *E. coli*, in the ancestor of the long-term experimental evolution (LTEE) and in evolved clones sampled over evolutionary time in the 12 populations (see deliverable D1.2). First, the effect of the deletion was investigated in the glucose minimal medium that was used in the LTEE. The *crp* deletion had a much more dramatic effect, on both growth rates and global transcription profiles, in evolved clones than in the ancestor. Since *crp* itself was not affected by mutations during evolution, these differential effects of the *crp* deletion were diagnostic of epistatic interactions between the *crp* deletion and beneficial mutations that occurred during evolution. Therefore, these changes in epistatic interactions sustained the robustness of regulatory networks by rewiring them.

In one of the twelve populations, called Ara–1, we identified the mutation that affected the robustness of the CRP-controlled regulatory network during evolution. This mutation affected the *topA* gene, encoding topoisomerase I, which is involved in the control of DNA topology in bacterial cells. This mutation occurred early during evolution (before 2000 generations) and was shown to be beneficial during the LTEE. We thus demonstrated that the control of DNA topology in bacterial cells actively contributed to the robustness of regulatory networks. Changes in DNA topology, which were beneficial in the LTEE by increasing the fitness of bacterial cells, also had a dramatic side effect by decreasing the robustness of the cells after a genetic perturbation (here, deletion of *crp*). Therefore, increased evolvability by DNA topology changes was associated with decreased robustness. These data are included in a manuscript in preparation (Wielgoss *et al.*, 2016)

Second, we tested the effect of the *crp* deletion in alternative environments (different from the one prevailing during the LTEE). Perturbing the regulatory network by deleting *crp* affected growth more severely in the evolved clones than in the ancestor only in the minimal glucose medium in which the evolution occurred and not in alternative environments. This suggests that evolution in the glucose environment strongly selected a particular structure of the regulatory network and that disturbing this structure resulted in lower robustness of the evolved clones specifically in that environment.

**Section 3: Robustness at the genome level in both TEV and *E. coli* experimental models**

The mutational robustness of the engineered TEV strains generated for deliverable D1.1 was evaluated using a protocol specifically developed to this end. We exposed viral particles from each one of the strains to the chemical mutagen $HNO_2$ at increasing incubation periods, and evaluated the infectivity (as a proxy of viral fitness) after each exposition time. The logic of this assay was as follows: the more robust a genotype, the less affected would it be by the treatment with $HNO_2$. Details can be found in deliverable D1.2. TEV-alkB and TEV-2b genomes encode for additional functional genes: the alkB domain involved in removing alkylation damage from RNA and the 2b suppressor of RNA silencing from *Cucumber mosaic virus* (CMV). *A priori*, a fitness benefit was expected for these two genes, either in terms of reducing mutational load or in terms of better interfering with the host defenses. In addition, carrying the *2b* gene adds functional redundancy (*i.e.*, a second suppressor of RNA silencing in addition to the normal one, HC-Pro) without adding genetic redundancy. TEV-eGFP encodes for an additional gene, the eGFP marker which does not

provide any fitness benefit to the virus. Finally, TEV-2NIb2 carries a duplication of the viral replicase gene *NIb* cloned in the second proteolytic position of the genome. This duplication generates both genetic and functional redundancy. We found significant differences in robustness between all the 11 genetic architectures, being the wildtype virus the most robust one and the TEV-alkB the less robust one. In conclusion, TEV wildtype genome architecture is more robust to mutational effects than any alternative one we have engineered in the laboratory. Any possible benefit of genetic redundancy in viral RNA genomes does not pays off the cost of replicating a longer genome. Interestingly, we observed that the average effect of deleterious mutations affecting non-coding regulatory regions was weaker than for mutations affecting coding sequences. By contrast, the magnitude of positive epistasis was stronger in regulatory than in coding sequences due to the role of RNA folding in the regulation of transcription and translation (Bernet and Elena, 2015).

In *E. coli*, three strategies were used. First, we investigated the impact of large chromosomal rearrangements on the evolution of bacterial cells (these data were published during the EvoEvo project in (Raeside *et al*., 2014), see also Deliverable 1.2). More than 100 large chromosomal rearrangements (deletions, duplications, amplifications, inversions) were substituted in all 12 populations after 40,000 generations. We showed that some rearrangements were involved in the fitness increase during bacterial evolution.

Second, we investigated the effect of increased mutation rates on the robustness of the genomes during evolution. We showed that the mutation rate increase had a large impact on genome evolution. Indeed, we found deleterious mutations that were substituted due to the increased mutation rates. We showed that these deleterious effects were compensated by the activity of RNA chaperones and this was the first proof of this buffering effect of RNA chaperones. These results have been already published (Rudan *et al*., 2015).

Third, we sequenced the genome of 264 evolved clones that were sampled in all 12 populations from the LTEE at ten different time points during 50,000 generations of evolution. This allowed us to analyse the effects of the rates and interactions between mutations and to determine the proportion of beneficial mutations. These data have also been published (Tenaillon *et al*., 2016). We showed that the fraction of beneficial mutations declined as fitness rises, and neutral mutations accumulated at a constant rate. Nonsynonymous mutations, intergenic mutations, insertions and deletions are overrepresented in the long-term populations, further supporting the inference that most mutations that reached high frequency were favoured by selection. These results illuminate the shifting balance of forces that govern genome evolution in populations adapting to a new environment.

**Resources committed**

Partner 2 (University Grenoble Alpes):

- Gaffé, Joël: assistant professor
- Hindré, Thomas: assistant professor
- Lamrabet, Otmane: post-doctoral fellow
- Raeside, Colin: PhD student
- Schneider, Dominique: Professor

Partner 5 (CSIC):

- Bernet, Guillermo P.: post-doctoral fellow
- Carrasco, José L.: post-doctoral fellow
- De la Iglesia, Francisca: lab. engineer
- Elena, Santiago F.: Professor
- Willemsen, Anouk: PhD student
- Zwart, Mark P.: post-doctoral fellow

### 2.2.3. Contribution 2 (Task 1.2): Evolvability at the population and regulatory network levels

**Section 1: Evolvability at the population level in the TEV experimental model**

The evolvability of the different TEV genotypes constructed for deliverable D1.1 was assessed by performing evolution experiments with each one of them into a set of novel host species. The magnitude of fitness improvement observed in each case was taken as a proxy to evolvability. Ten lineages of each engineered genome organization were maintained by undiluted serial passages in each of three hosts: *N. tabacum* (the natural reservoir), *Nicotiana benthamiana* and *Datura stramonium* (two novel hosts). In all cases, three 9-weeks passages were performed. Under such demographic conditions, we had previously shown that the effect of selection in fixing beneficial alleles was maximized. To explore the evolution of viral genomes under relaxed selective pressures, wildtype viruses were evolved in transgenic plants expressing the TEV replicase *NIb* gene. Details on these experiments are provided in deliverables D1.3 and D1.5.

We found that the extent by which fitness was improved strongly depended on (*i*) the fitness of the starting clone and (*ii*) the novel host. The lower the fitness of the starting clone, the larger the magnitude of fitness improvement observed. While fitness improvements in the reservoir host were always associated to the removal of additional genes, except in the case of the 2b silencing suppressor from CMV, fitness in alternative hosts not always concurred with the removal of additional genetic material. We concluded from these experiments that increases in genome complexity (by horizontal gene transfer of new genes) only compensates if the added function provides a short-term benefit. Any second-order benefit, as for instance increased robustness due to genetic redundancy, did not paid for the cost of replicating longer genomes. These results have been published in (Zwart *et al*., 2014; Willemsen *et al*., 2016a; Willemsen *et al*., 2016b). If a viral gene was constitutively expressed from the host genome, viral populations evolved that have removed their own copy. This proves that selection for fast replication is the dominant selective force operating over viral populations. This result are published in (Tromas *et al*., 2014).

In the case of reorganized genomes (Majer *et al*., 2014), virus accumulation significantly improved in all cases to levels close to those observed for the wildtype virus. However, viruses with altered gen orders remained weaker competitors than the wildtype TEV during head-to-head competition experiments, meaning that the fitness cost associated to gene order alterations was large and difficult to compensate. These results are published in (Willemsen *et al.* 2016b; Willemsen *et al*., 2016c) and reviewed in (Elena, 2016a).

Finally, we explored the effect that past evolutionary events had on the evolvability of TEV populations. To do so, we generated a collection of mutants that occupied different positions in the

adaptive fitness landscape, at increasing distances from the local optimum, and evaluated the likelihood of these genotypes to reproduce the adaptive walk and reach this local optimum. We found that reproducibility of evolution was very low and that most of the evolved lineages moved to different regions in the adaptive landscape. These results are published in (Cervera *et al*., 2016a).

**Section 2: Evolvability at the regulatory network level in the *E. coli* experimental model**

A manuscript including the results of this section is currently in preparation (Lamrabet *et al*., 2016).

New evolution experiments were started with ancestors corresponding to *crp*-deleted strains chosen from the results obtained during Task 1.1 (contribution 1). Four different such ancestral strains with highly perturbed regulatory networks were chosen and propagated (five replicates each) by daily transfers in an environment in which they had very low fitness (see deliverable D1.5). The fitness defects were restored after only 100 generations of evolution in all genetic backgrounds (see deliverables D1.3 and D1.4). The genomes of evolved clones were sequenced to identify the mutations that restored fitness after the perturbation of the regulatory network.

A high level of parallelism was observed with mutations identified in identical genes in most evolved clones. They affected the transcriptional regulatory region of genes involved in the transport of the sugar present in the evolution environment. These mutations increased the transcription of these genes, thereby resulting in better sugar transport, which allowed restoration of the fitness defects (see Deliverable 1.5). Therefore, perturbed regulatory networks evolved toward phenotypic improvement by changing the regulation of target genes through transcriptional rewiring of their expression.

**Resources committed**

Partner 2 (University Grenoble Alpes):

- Hindré, Thomas: assistant professor
- Lamrabet, Otmane: post-doctoral fellow
- Schneider, Dominique: Professor

Partner 5 (CSIC):

- Carrasco, José L.: post-doctoral fellow
- Cervera, H.: PhD student
- De la Iglesia, Francisca: lab. engineer
- Elena, Santiago F.: Professor
- Tromas, Nicolas: PhD student/post-doctoral fellow
- Willemsen, Anouk: PhD student
- Zwart, Mark P.: post-doctoral fellow

### 2.2.4. Contribution 3 (Task 1.3): Phenotypic innovation at the population and regulatory network levels

**Section 1: Phenotypic innovation at the population level in the TEV experimental model**

The phenotype of a virus is a complex trait that can be defined in several ways. From a viro-centric perspective, the most relevant phenotype is viral fitness. Viral fitness itself is also a quite complex

trait, with many components. To better evaluate viral fitness, we have quantified two different components: infectivity, which is a proxy to between-host fitness and measures the efficiency of initiating a new infection, and viral load which is directly related to the efficiency replicating and accumulation within cells and moving from cell to cell and from the inoculation site to distal tissues.

We first sought to evaluate a first phenotypic innovation, namely, the ability to infect and replicate in novel hosts. We evaluated the fraction of all possible mutations that may confer TEV the ability to infect a set of new hosts. To do so, we had revisited the fitness and infectivity of a collection of single-nucleotide substitution mutants across a panel of susceptible hosts, including *N. tabacum*. *N. benthamiana*, *D. stramonium*, *Capsicum annuum*, *Solanum lycopersicum*, *Helianthus annuus*, *Gomphrena globosa*, and *Spinacia oleracea*. As a way to assess the extent of phenotypic innovation for infectivity, we counted the number of genotypes that had infectivity larger than the wildtype TEV on each host species. A highly significant positive correlation was observed between this frequency and the genetic distance between the reservoir host and the alternative host. Then we measured viral load. A significantly negative correlation between virus accumulation and the genetic distance between the reservoir host and the alternative host was observed. Together, these results suggest that genetic diversity exists within viral populations that allow for phenotypic innovation. Details of this study are provided in deliverable D1.4. Next, we sought to determine the effect that long-term evolution in the reservoir host (*N. tabacum*) and one of the above alternative hosts (*C. annuum*) had in phenotypic innovation and in the underlying dynamics of allele substitution. Parallel independent evolution lineages were evolved in both hosts, their fitness evaluated at the end of the experimental evolution phase and the genetic composition of each lineage at each passage and at different plant tissues evaluated by Illumina NGS. In full agreement with the above results, we found a large extent of phenotypic innovation in the novel host (measured as fitness increases and new symptoms). At the molecular level, new beneficial alleles swap and got sequentially fixed in the novel host as predicted by the clonal interference model. By contrast, lineages evolved in the reservoir host showed no phenotypic innovation and alleles changed in frequency along time as expected for a mutation-drift model. These results have been published in (Cuevas *et al*., 2015).

Phenotypic innovation in terms of expansion of TEV host range was also explored in the context of fitness landscapes. We evaluated the ruggedness and other topological properties of a local fitness landscape involving the five mutations responsible for adaptation of TEV to the novel host *Arabidopsis thaliana* both in the novel and ancestral hosts and found that both landscapes were macroscopically similar (*i.e*., both were rugged and contained several holes due to lethal mutations) but differed in the details (*i.e*., antagonistic pleiotropic effects). These results have been published in (Lalić and Elena 2015; Cervera *et al*., 2016b), and reviewed in (Elena, 2016b).

Second, we had evaluated phenotypic innovation as a change in the way evolving TEV populations interact with the transcriptome of their hosts. In this case, “phenotype” was considered as the whole set of mRNAs that exist at a given time point in an infected plant. To tackle this issue, lineages of TEV adapted to five different ecotypes of the experimental host *A. thaliana* were used to inoculate each one of the ecotypes. The five ecotypes differed in their susceptibility to infection with the ancestral TEV strain (Hillung *et al*., 2014; 2015). We found that: (*i*) the extent of phenotypic innovation (that is, the magnitude of the difference in transcriptomic profiles between plants infected with the ancestral and evolved viral isolates) was dependent on the local host genotype, with lineages evolved in a common genotype showing more similarities than genotypes

adapted to other host ecotypes. (*ii*) Some host genes were altered by all viral lineages in all genotypes, whereas others where both lineage-specific and/or host ecotype-specific. The former should be considered as universal responses to TEV infection whereas the latter shall be considered as specific of each particular plant ecotype-virus genotype interaction. (*iii*) Phenotypic innovation comes with a cost: the more divergent a phenotype from the ancestral virus, the lower the fitness of the virus in the ancestral host. (*iv*) Generalist viruses, that is, those able of successfully infecting a broader range of host genotypes (more phenotypically diverse) altered the expression of similar sets of genes across all hosts tested; by contrast, specialist viruses (less phenotypically diverse) differed in the set of genes that were altered on each potential host tested. More details are provided in deliverable D1.6 and in the resulting publication by Hillung *et al*. (2016).

**Section 2: Phenotypic innovation at the regulatory network level in the *E. coli* experimental model**

We investigated the relationships between the structure of global regulatory networks and the bacterial ability to produce phenotypic innovation (see deliverable D1.6). In particular, clones with different regulatory network structures were propagated under conditions known to promote adaptive diversification and investigated for their ability to produce co-existing lineages of bacterial cells with differential phenotypic abilities. In addition, we investigated the involvement of global regulatory networks in the physiology and mechanism of the adaptive diversification event that was previously detected in the population Ara–2 of the LTEE (Plucain *et al*., 2014).

We propagated two types of strains in conditions known to promote adaptive diversification, *i.e.* the presence of two carbon sources (see deliverable D1.6). One set of strains harbored a typical regulatory network structure for *E. coli* and the other set was perturbed by deletion of *crp*, one of the central hubs of bacterial regulatory networks. While emergence of two different lineages of bacterial cells, specialized for the consumption of each of the two carbon sources, was readily detected for the first set of strains, no such polymorphism emerged when the regulatory network was perturbed.

In population Ara-2 of the LTEE, a stable and dynamic polymorphism was established and two lineages called S and L co-exist since more than 50,000 generations. Two lines of evidence demonstrated that the structure of regulatory networks was involved in the emergence and maintenance of this phenotypic innovation (see deliverable D1.6). First, perturbing the regulatory networks by deleting the *crp* gene in either of the two lineages completely abolished the ability of S and L clones to co-exist. Second, a combination of modeling and experimental approaches allowed us to characterize the physiological and molecular mechanisms of the emergence of the polymorphism. The L clones produced a new ecological niche, by secreting acetate, which provided the opportunity for the emergence of the S lineage that exploited this new niche. Moreover, each lineage became fitter and fitter in its own ecological niche over evolutionary time. Therefore, this combination of niche construction and character displacement was at the origin of this polymorphism. These results were recently published in (Großkopf *et al*., 2016). We further found the mutation that allowed the S clones to exploit the new ecological niche, namely to consume acetate. This mutation affected the *arcA* gene encoding one of the biggest global regulators of *E. coli*. We further showed that this mutation rewired the regulatory network in the S lineage, providing it with the opportunity to growth efficiently with acetate as a carbon source. These results are the subject of a manuscript that will soon be submitted (Consuegra *et al*., 2016).

**Resources committed**

Partner 2 (University Grenoble Alpes):

- Consuegra, Jessika: PhD student
- Gaffé, Joël: assistant professor
- Hindré, Thomas: assistant professor
- Lamrabet, Otmane: post-doctoral fellow
- Schneider, Dominique: Professor

Partner 5 (CSIC):

- Carrasco, José L.: post-doctoral fellow
- Cervera, H.: PhD student
- Cuevas, José M.: post-doctoral fellow
- De la Iglesia, Francisca: lab. engineer
- Elena, Santiago F.: Professor
- Hillung, Julia: PhD student
- Lalić, Jasna: PhD student/post-doctoral fellow
- Willemsen, Anouk: PhD student

### 2.2.5. Conclusion

The experiments performed in WP1 led us to identify some "EvoEvo strategies" that help organisms to improve their robustness, evolvability and innovative potential. As anticipated, these strategies are different in the two model organisms studied – a virus and a bacteria – owing to the "simplicity" of the former and to the "complexity" of the latter. Indeed, in the former EvoEvo strategies are based on the population and genome levels while in the latter, they are mainly – but not only – based on the cellular networks.

**EvoEvo strategies for the evolution of robustness:**

- Robustness in TEV populations:
  - A positive dependence of mutational robustness on effective population size has been observed in evolving populations of TEV.
  - The effect of mutation rate on the evolution of mutational robustness depends on effective population size: it is positive in small populations but negative in large ones.
- Robustness of regulatory networks in *E. coli*:
  - Evolution strongly selected a particular structure of the regulatory network. Disturbing this structure resulted in lower robustness of the evolved clones specifically in the evolution environment.
  - The physiological effect in the evolution environment of perturbing the *crp* network was larger in the evolved clones than in their ancestor. We identified the mutations responsible for this larger effect and showed that they affected the level of DNA supercoiling. A link between catabolic repression and DNA topology is therefore responsible for maintaining robustness at the level of regulatory networks in *E. coli*.
  - The level of DNA topology controls the balance between the robustness and evolvability of regulatory networks during long-term evolution in bacterial cells.

- Robustness at the genome level in TEV and *E. coli*:
  - Duplicating a gene in a viral RNA genome does not contribute to increase its mutational robustness. The cost associated with replicating an additional piece of RNA does not compensate for the possible beneficial effects in buffering the effect of deleterious mutations.
  - Addition of the alkB domain that repairs alkylation on RNA molecules was not beneficial for TEV in terms of increasing its mutational robustness, suggesting that virus' robustness is a population collective process rather than individual-based.
  - Large chromosomal rearrangements have been substituted during evolution of the 12 populations of the LTEE, heavily restructuring the chromosome but without dramatic effect on robustness. They even in some cases increased the fitness of the evolved bacterial clones.
  - RNA chaperones, like protein chaperones, can have a buffering effect on deleterious mutations, thereby increasing the robustness of bacterial cells.
  - We identified the shifting balance of forces that govern genome evolution in bacterial populations adapting to a new environment with beneficial mutations declining as fitness rises and neutral mutations accumulating at a constant rate.

**EvoEvo strategies for the evolution of evolvability:**

- Evolvability in TEV populations:
  - Viruses with genomic duplications readily remove the second copy, returning to the wildtype genome architecture. Genetic redundancy does not improve the evolvability of RNA virus by relaxing selective constraints operating on multifunctional proteins.
  - If a viral gene is transferred into the host genome, this relaxes selection on the viral copy and viruses with deletions are favored, resulting in a virus that lacks this gene.
  - Genomes with alternative gene orders are viable. They quickly evolve and improve in absolute fitness (*i.e.*, virus accumulation) and virulence. However, they are still inferior competitors than the wildtype TEV.
  - Adding a gene with a short-term benefit results in its evolutionary retention. The case of a successful increase in genome complexity was the addition of a second suppressor or RNA silencing, the 2b protein from CMV, that operates at a different level on the silencing pathway than the original HC-Pro protein.
  - Rates of adaptive evolution vary among genome architectures, with the wildtype virus showing an intermediate value. No correlation between TEV robustness and evolvability has been observed.
  - When considering the evolution of genome architecture, host species jumps might play a very important role, by allowing evolutionary intermediates to be competitive.
  - The reproducibility of adaptive evolution is minimal in TEV due to the ruggedness of the adaptive landscape. Only a fraction of populations starting at one mutational step away from the local optimum reached it, fixing the exact same set of mutations, while populations starting farther away from the local optimum end up in distant regions of the landscape, fixing a new set of mutations.
- Evolvability of regulatory networks in *E. coli*:

- Regulatory networks are highly evolvable in bacteria. Deletion of a central hub of regulatory networks resulted in strong growth alteration that was reversed after only 100 generations of evolution.
- The restoration occurred by increasing the transcription of target genes involved in sugar uptake.
- Rewiring of perturbed regulatory networks involves additional regulatory changes in bacteria that influence and restore the altered phenotype.

**EvoEvo strategies for the phenotypic innovation:**

- Phenotypic innovation at the population level in TEV:
  - Phenotypic innovation in terms of accessibility to novel hosts has been shown in TEV. Genetic variability exists that allow the virus to infect and successfully replicate in a panel of novel hosts.
  - The extent of TEV phenotypic innovation depends on the genetic relatedness between host species and the natural reservoir. In terms of infectivity, the fraction of possible beneficial mutations increases with genetic distance. By contrast, in terms of virus accumulation, the fraction of beneficial mutations decreases with genetic distance among host species.
  - Genes and functional categories differentially expressed by plants infected with locally-adapted TEV isolates have been identified, showing heterogeneous responses among ecotypes, although significant parallelism existed among lineages evolved in the same ecotype. Adaptation to novel host genotypes results in phenotypic innovations, which are host-specific.
  - Plant resistance genes are not the only drivers of viral adaptation, as functional groups related to secondary metabolism and responses to abiotic stress and senescence were also pervasively over-represented in infected plants.
  - The nature of transcriptomic perturbations varies among generalist and specialist viral lineages. Whilst the generalist induced very similar perturbations in the transcriptomes of the different ecotypes, the perturbations induced by the specialist were divergent. Plant defense mechanisms were activated when the infecting virus was specialist but they were down-regulated when infecting with generalist.
- Phenotypic innovation in *E. coli* regulation network:
  - Regulatory networks are essential for phenotypic innovation in bacteria. Divergence of bacterial lineages, which is an essential trait both in evolutionary biology and in medicine, relies on the structure of regulatory networks.
  - The structure of regulatory networks is an essential trait in bacterial cells. As was shown since decades, it allows a fast answer to environmental changes by transiently modifying the global expression profiles. We showed here, in the framework of EvoEvo, that the structure of regulatory networks are also essential for the establishment of stable adaptive diversification events in bacteria, which is considered as the first step of speciation.

## 2.3. Workpackage 2: Development of an integrated modelling platform

### 2.3.1. Introduction

One of the central objectives of the EvoEvo project was to enable simulation of "evolution of evolution" in a computational framework. Computational models have been used to study evolution since the beginning of the 1990s (Adami, 2006). However, since then, most computational models used a partial representation of the genotype-to-phenotype mapping, generally in a fixed, predefined fitness landscape. By simulating the evolution of some particular organisation level (the genome, the genetic regulation network, the metabolic network...), different authors have studied evolution of robustness, evolvability or variability of these specific levels (Wilke *et al*., 2001; Knibbe *et al.*, 2006; Crombach and Hogeweg, 2008; Cuypers and Hogeweg, 2012). Yet, "EvoEvo" is an integrative concept exactly as fitness is. Indeed fitness is the result of the interaction of all the organisation levels of the organism, including its interactions with its environment. Similarly, the robustness/evolvability/variability of the phenotype is the result of the interaction of robustness/evolvability/variability at all the organisation levels of the organism (including its interactions with its environment!). Furthermore, these properties are not independent and they may interact in a cooperative or competitive way (*e.g.*, evolving chaperone proteins reduces the phenotypic variability, thus increasing the robustness). That is why, in the context of the EvoEvo project, we developed an integrated computational model of EvoEvo, including the main organisation levels of the (evolvable) genotype-to-phenotype map (genome, transcription network, metabolic network, phenotype, fitness, population).

The development of such an integrated modelling platform was the main objective of WP2. However, being aware that the development of such a model was a difficult and risky objective, this objective was divided in three subtasks to sequentially incorporate more and more levels in the model. So WP2 was divided into tasks 2.1 (integration of sequence and network levels), 2.2 (modelling regulation and metabolism), 2.3 (modelling environment, population and trophic network) and 2.4 (development of the integrated model). The objective was to reduce the risk inherent in task 2.4 as well as giving an alternative plan in case of failure of task 2.4. Indeed, the development of this "model stack" was shown to be very hard, and WP2 ran slow in the first months of the project. However, it also proved efficient, as a global strategy: the time devoted to tasks 2.1, 2.2 and 2.3 enabled a speedy accomplishment of task 2.4 (which was completed on time) and the integrated model was released early enough and was efficient enough for WP3 to use it to model the LTEE (see section 2.3).

Yet, an unexpected difficulty arose: a complex model like the one we envisioned necessarily incorporates a large set of parameters, and complete exploration of the parameter space proved to be impossible. We overcame this difficulty by two means (1) a minimal parameter exploration to identify the effect of the main parameters on the evolutionary dynamics, and (2) a vast review of the literature in order to fix the order of magnitude of the model parameters (the exact values being fixed by the evolutionary process).

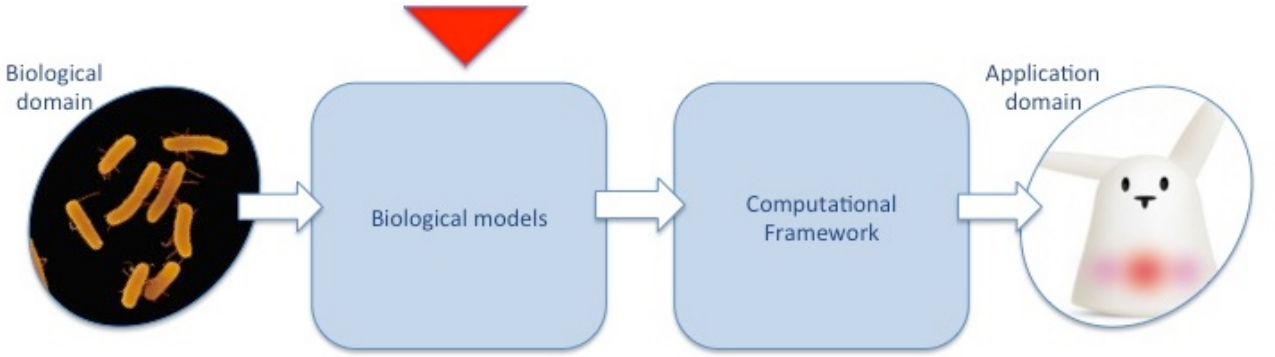


Figure 3: Position of WP2 in the biology-to-application scheme of EvoEvo

The rest of this section is organized as follow. Since we were successful in the development of the integrated model, the partial models developed in tasks 2.1, 2.2 and 2.3 were not used in the computational experiments (which used only the integrated model). We thus group the description of these models in a same contribution (section 2.3.2, Contribution 1: Development of the "model stack"). Then, we detail the development of the integrated model (section 2.3.3, Contribution 2: Development of "Evo$^2$Sim"). Finally, we discuss the choice of the model parameters (section 2.3.4, Contribution 3: Parameters choice for Evo$^2$Sim).

### 2.3.2. Contribution 1: Development of the "model stack" (tasks 2.1, 2.2 and 2.3)

Our initial objective was to develop the integrated model in four steps to test and validate all the submodels before integrating them all in "Evo$^2$Sim" (the name was chosen later). We thus released three submodels focusing on different aspects of evolution, namely biochemistry, networks and (minimal) ecology:

- **Submodel 1: Integrating sequences and metabolisms (task 2.1)**. The objective of this submodel was to choose and implement a coarse grained model of the genome encoding for a simplified metabolic network. At the genomic level, we wanted to combine some properties of the Aevol model developed in Lyon by INRIA and of the "Pearls on a String" (*aka* PoaS) formalism used in Utrecht. This was achieved by organizing the genome as a circular string of pearls, and allowing for "non-coding" pearls. One of our objectives was to allow evolution to play with the amount of non-coding sequences, since we suspected a strong effect of EvoEvo. At the metabolism level, our objective was again to gather properties of PoaS (the encoding of a metabolic network) and of Aevol (unlimited complexity of the phenotype). This was achieved by using an artificial chemistry in which metabolites are identified by an integer tag (infinite metabolite universe) and the "proteins" encoded on the genome encode for enzymes that transform metabolites (*i.e.*, integers) into one another. This submodel was implemented and tested, and released on the project website in December 2014 (deliverable D2.2).
- **Submodel 2: population model (task 2.2).** In submodel 1 all individuals behave independently and the only interaction between them is their fitness value. Submodel 2 adds the possibility for the individuals to exchange metabolites through diffusion in the environment. Such inter-individual interaction is proposed by Banzhaf *et al.* (2016) as a prerequisite for open-endedness, particularly when an explicit fitness criterion is used (which is the case here). To this aim we added basic but important properties to the model: pumps (enabling individuals to actively pump-in or pump-out metabolites), metabolic

diffusion and degradation, and cell membrane permeability. This submodel was implemented and tested, and released on the project website in May 2015 (deliverable D2.4).

- **Submodel 3: modelling regulation and metabolism (task 2.3).** While submodel 1 includes only a metabolic network, submodel 3 adds a genetic regulation network that controls the transcription rate of the enzymes (thus controlling the activity of the metabolic network). It is based on a more precise description of the genome that, in addition to non-coding and coding (enzymes) pearls, also contains promoter pearls, transcription-factor-coding pearls and binding sites. This enables the regulation network to evolve both in *cis* and in *trans* (as real regulation networks). The main difficulty of this submodel is the dual interaction between metabolic and regulation networks. While the regulation network naturally influences the metabolic network through modification of transcription rates (hence of enzymes concentrations), it also needs to sense the metabolic activity in order to react to it. We model this through the possibility of the transcription factors binding to coenzymes (like in the well-known Lac-operon). Coenzymes can change the binding property of transcription factors, hence enabling the regulation network to behave differently depending on the metabolic context. This submodel was implemented and tested, and released on the project website in May 2015 (deliverable D2.6)

The development and test of the "model stack" was fruitful on many points. First, it showed that all modelling choices were reasonable, both in terms of results (*i.e.*, efficient individuals evolved in all models – which fulfilled milestones MS2, MS4 and MS5) and in terms of computability (*i.e.*, all these models were runnable on classical computers in a reasonable time). As such, it efficiently paved the way to the integrated model. However, it also revealed an important property of the submodels: when individuals were not interacting (*i.e.*, in submodels 1 and 3), the evolutionary dynamic was considerably simplified. On the one hand, this was a confirmation of the theoretical discussions exposed in (Banzhaf *et al.*, 2016). On the other hand, this result quickly led us to stop using the submodels and to concentrate on the development and usage of the integrated model.

**Resources committed**

All submodels were developed by Charles Rocabert, Guillaume Beslon and Carole Knibbe (INRIA). All members of the consortium were involved in the discussions on modelling choices, particularly Paulien Hogeweg, Bram van Dijk, Thomas Cuypers and Sandro Colizzi (Utrecht University), Otmane Lamrabet and Dominique Schneider (Université Grenoble-Alpes) and Susan Stepney (University of York).

### 2.3.3. Contribution 2: Development of "Evo$^2$Sim"

Evo$^2$Sim is an integrated model that gathers all the submodels presented previously. It is extensively described in deliverable D2.7 (models specification) and D2.8 (model code – released on the project website in May 2015) as well as in (Rocabert *et al.*, 2015) and (Rocabert *et al.*, 2016b). Figure 4 recalls the main properties of the model.

Since all submodels had been previously tested and validated, the conception of the integrated model was not a major issue. However, its implementation was much more difficult than anticipated, for at least two interconnected reasons:

- The numerical solving of metabolic equations quickly appeared to be very difficult owing to the stiffness of the equations. The tuning of the ODE solver used in the model (GSL) had to be improved to take into account the specificities of the model (typically, the stiffness of the metabolic equations could not be anticipated since it is likely to change at every mutation).
- The computational load of the integrated model appeared to be much larger than anticipated due to three independent factors: (1) the complexity of the model itself (*i.e.*, the number of interacting elements). This was considered as a major risk in the project. However, two other factors quickly worsened the situation. (2) The stiffness of the metabolic equations led us to use highly demanding solving strategies and (3) although the model showed interesting evolutionary properties, it also showed that evolution within the model is rather slow (mainly due to the fact that individuals replicate slowly). Hence, observing interesting dynamics often needed some 100,000 simulation-steps or even 500,000 simulation-steps – see *e.g.*, (Rocabert *et al.*, 2016b).

To overcome this difficulty, we strongly optimized the simulator code and used parallel libraries to execute the model on multi-core computers. Compared to the initial versions of the model, we obtained a 10x faster implementation that made it possible to use the model in practice (which was not fully guaranteed at the model release).

From an algorithmic point of view, *in silico* experimental evolution is very similar to evolutionary computation. However, a strong difference is that the latter is mainly concerned by the last generation of the evolutionary process (*i.e.*, the best organism evolved so far) while the former is mainly concerned by the evolutionary process itself. That is why a model dedicated to *in silico* experimental evolution must include tools to *a posteriori* analyse the evolutionary dynamics. In Evo$^2$Sim the main analysis tool is a model viewer that enables the user to follow all the properties of the system along the evolution (concentration of metabolites in the environment, population n-evolution, fixed mutation rates, phylogenetic structure… see Figure 5).

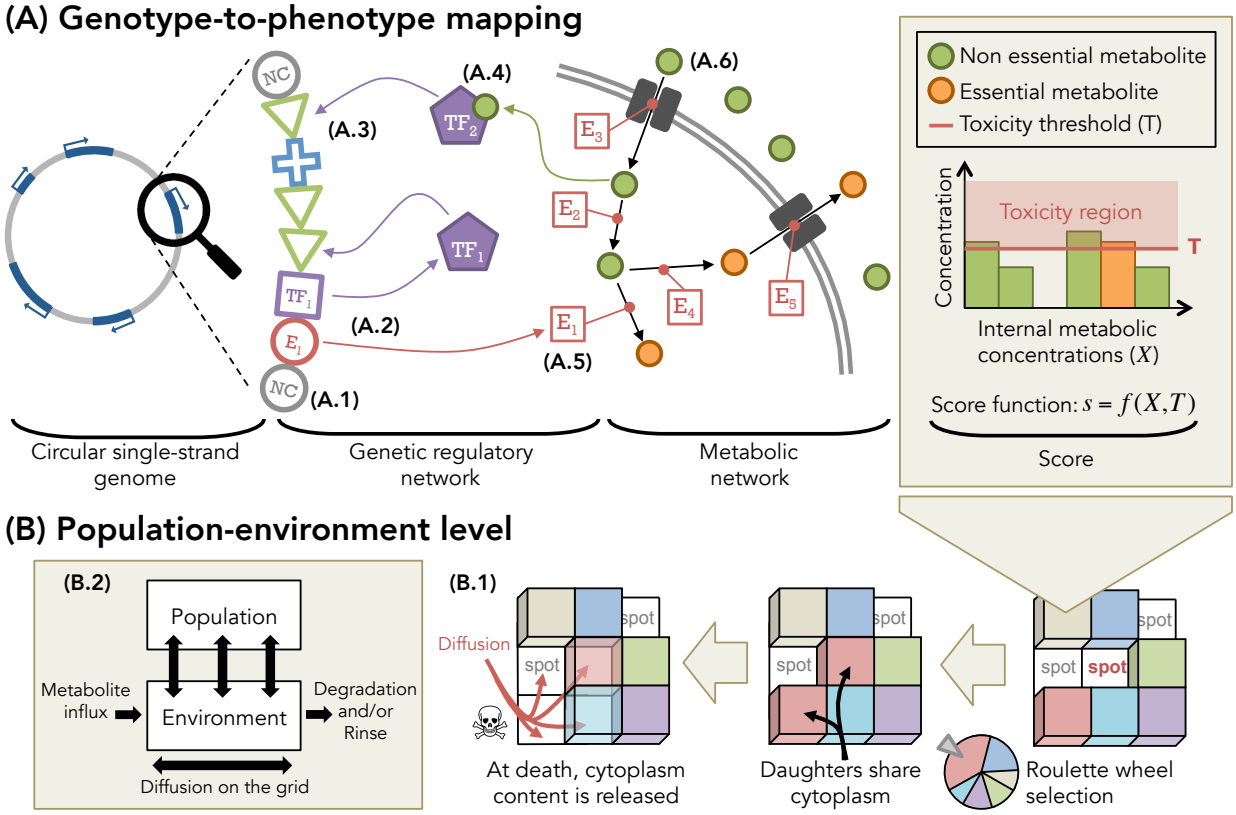


Figure 4: Presentation of Evo[2]Sim model. The genotype-to-phenotype mapping, as well as the population and environment, are represented here. **(A) Description of the genotype-to-phenotype mapping.** Organisms own a coarse-grained genome made of units. This genome is a circular single-strand sequence, with a unique reading frame. Non coding (**NC**) units are not functional **(A.1)**. The arrangement of the units on the sequence defines functional regions, where a promoter (**P**, blue cross) controls the expression of enzyme coding units (**E**, red circles) or transcription factor coding units (**TF**, purple squares), thereby allowing for operons (here, one E and one TF). When coding units are expressed **(A.2)**, they contribute to the genetic regulatory network (for TFs) and the metabolic network (for Es). Depending on their attributes, transcription factors bind on binding sites. **(A.3)** If they bind on the enhancer sequence (binding sites flanking the promoter upstream), the promoter activity is up-regulated. If they bind on the operator sequence (binding sites flanking the promoter downstream), the promoter activity is down-regulated. **(A.4)** Metabolites can bind on a transcription factor as co-enzymes, and activate or inhibit it, depending on transcription factor attributes. Enzymes perform metabolic reactions in the cytoplasm **(A.5)**, or pump metabolites in or out **(A.6)**. The score of an organism is computed from its "essential metabolites'" (usually the score is the sum of essential metabolite concentrations). Lethal toxicity thresholds are applied to each metabolic concentration and forbid organisms to accumulate resources. **(B) Description of the population and environment levels.** Organisms are placed on a 2D toroidal grid, and compete for resources and space. When an organism dies, it leaves its grid cell empty and organisms in the Moore neighborhood (if any) compete to divide in available space. The competition is based on scores, a minimal threshold being applied on scores to forbid worst organisms to divide. At division, daughters share cytoplasm content (enzymes and metabolites). At death, metabolites from the cytoplasm are released in the local environment, and diffuse on the grid **(B.1)**. On the largest scale, the population evolves on the environment by up-taking, transforming and releasing metabolites. Metabolites then diffuse and are degraded. This strong interaction between the population and the environment allows for the evolution of complex ecological situations, depending on environmental properties **(B.2)**.

.

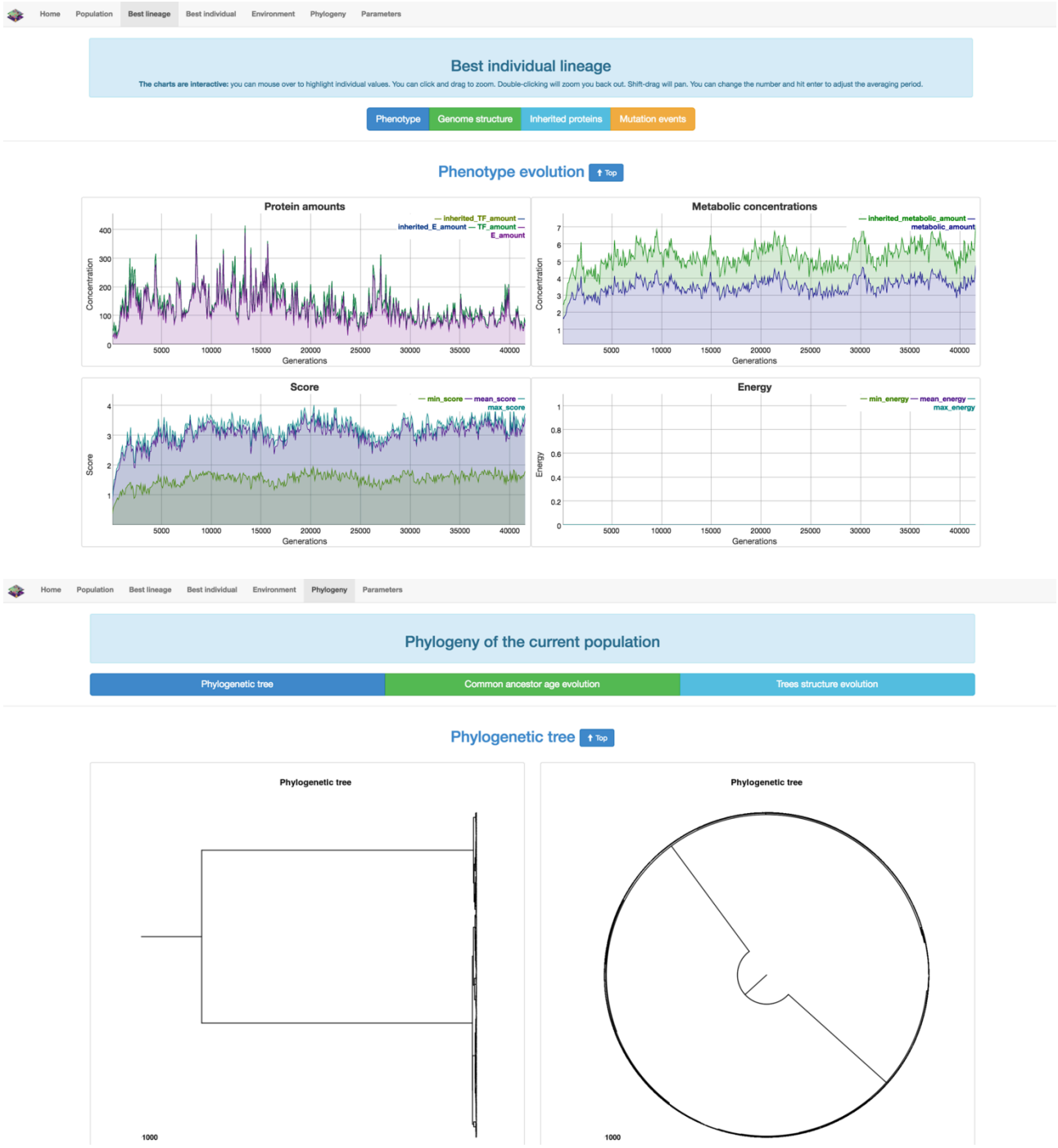


Figure 5: Evo$^2$Sim viewer. Different panels enable the user to observe the evolution in action at the individual, population and phylogenetic levels. Top: best lineage property. Bottom: phylogenetic tree. Here the population is structured in two groups co-evolving independently for more than 30,000 generations (~400,000 time steps).

## Resources committed

Evo$^2$Sim was developed by Charles Rocabert, Guillaume Beslon and Carole Knibbe (INRIA). The fine tuning of the ODE solver benefitted from Samuel Bernard and the code optimization and code parallelization was done by Charles Rocabert and Jonathan Rouzaud-Cornabas.

### 2.3.4. Contribution 3: Parameters choice for Evo²Sim

Individual-based models usually contain many inter-dependent parameters. This is the case with Evo²Sim. Since we want to be able to compare the evolutionary dynamics of Evo²Sim with what is observed *in vivo*, we need to tune the model parameters such that the dynamic regime of our "cells" is comparable with the dynamic regime of real cells (*e.g.*, the life-time of the cell, the amount of protein produced – or inherited, the enzymatic constants… must be in the same order of magnitude). More importantly, since the parameters are all interdependent, the correct orders of magnitude for all the parameters must be chosen, in order to avoid soundless behaviours *e.g.*, the metabolic reactions must be fast enough to enable the cell to react to environmental changes, but too fast reactions must be avoided since they would not be possible in practice. In particular, we have to carefully choose all time-related parameters such that the cells are able to react to changes in their environments (meaning that they are able to sense the changes, modify the enzyme production accordingly and stabilize on the new behaviour in a reasonable time).

Following the development of Evo²Sim and the first experiments, an important review of the literature was conducted in order to identify the correct dynamic regimes and the orders of magnitude of the most important parameters (note that only the orders of magnitude are important here since the exact values are fixed by the evolutionary process).

**Time unit:**

At the molecular level, we use a 1 minute time unit (*i.e.*, the ODE equations are updated each minute). The individual time unit is 100 minutes (*i.e.*, the population dynamics is updated each 100 minutes depending on the current molecular state of the cells). Given this time-scale, we fix the death rate at 0.07 (*i.e.*, a cell lives in mean 1400 minutes, ~24 hours).

**Protein degradation rate:**

In a cell, the half-life of the molecular components (proteins) varies between 5 and 70 hours depending on the proteins (the proteins with smallest half-life generally being mutant or badly folded ones). In the model the protein half-life is fixed by the degradation rate that is a non-evolvable parameter. We use a degradation rate of 0.0005 per protein per minute, corresponding to a protein half-life of approximately 24 hours (1440 minutes since $\ln(2)/1440 \cong 0.0005$).

**Concentration units:**

In the model we use arbitrary concentration units (denoted "Z" later) but all constants must be scaled to the production and degradation rates and the size of the cell. Since we consider that we model bacterial cells, the cell volume and the grid patch volume are fixed to 4 $\mu m^3$ (corresponding to the estimated volume of an *E. coli* cell[1]). We choose to use the production rate as a reference-rate so the maximum production rate is 1 (minimum production rate being 0). Given the degradation rate, the maximum concentration at equilibrium is 1/0.0005 = 2000 AU. Since enzymatic concentrations in *E. coli* vary between 5 and 500 nM (Hugues Berry, personal communication), the concentration unit (Z) is $10^{-10}$ M and the protein concentration varies between

[1] For a global reference on the biological values, see: http://bionumbers.hms.harvard.edu/

0 .. 2000 Z (0 .. 200 M). Below 1 Z, enzymes are considered to have disappeared from the cell (indeed, given a Z value and the cell volume, Z < 1 corresponds to less than 1 molecule per cell[2]).

In bacteria intracellular concentration of metabolites varies between $10^{-7}$ and $10^{-2}$ M (*i.e.*, $10^3$ to $10^8$ Z). Typical growing medium used to cultivate *E. coli* contain between 1 and 20 g/l of glucose (the minimal concentration under which *E. coli* does not grow being 4 to 5 g/l). Given Z this corresponds to a metabolite concentration of $5x10^7$ to $10^9$ Z in the medium (glucose molecular weight: ~180 g/mol). Since we intend to model the LTEE where glucose is the only carbon source (*i.e.*, in the model a single metabolite is provided in the external medium at regular time steps), we introduce a carbon source (*i.e.*, a metabolite which tag is not a prime number) at a concentration $2x10^8$ Z each ~14 time steps (*i.e.*, each 24h).

**Range of $K_m$ and $k_{cat}$ values:**

$K_m$ and $k_{cat}$ are the two parameters of the Michaelis-Menten enzymatic equation used to model the metabolic network. They are enzyme-specific (thus being coded in the genome) and they are free to evolve. One of the central difficulties is to restrict their evolution to a chemically-sounded-domain given the concentration and time units used in the model. In natural enzymes, the observed values are (Bar-Even *et al.*, 2011):

- $K_m$: between $10^{-7}$ and $10^{-1}$ M (*i.e.*, $10^3$ to $10^9$ Z),
- $k_{cat}$: between 6 and 60000 minute$^{-1}$ (median value: 600 minute$^{-1}$)

Given the range of variation of both values, they are encoded in logarithmic scale, resulting in a range of 3 to 9 for $K_m$ and 0.8 to 4.6 for $k_{cat}$ (both in $\log_{10}$). However, these values raise an unanticipated difficulty: independent mutations of $K_m$ and $k_{cat}$ could result in a ratio of $k_{cat}/K_m$ varying between $6x10^{-9}$ and $6x10^7$ min/Z which is a nonsense both mathematically (ODEs become strongly stiffened!) and biologically (in natural enzymes the $k_{cat}/K_m$ ratio would vary between $6x10^{-6}$ and $6x10^{-2}$ min/Z). Indeed, in natural enzymes there is a tradeoff between $K_m$ and $k_{cat}$ values (Bar-Even *et al*., 2011). We thus decided to parameterize the Michaelis-Menten reaction by two parameters: $k_{cat}$ and the $k_{cat}/K_m$ ratio (indirectly specifying the $K_m$ value).

**Minimum score:**

In Evo[2]Sim the fitness of the cell is given by a score that is the amount of essential metabolites (metabolites whose tag is a prime number) in its cytoplasm (being these essential metabolites produced or inherited). The parameter MINIMUM_SCORE indicates the minimal score below which a cell cannot replicate. Given the concentration unit and the size of the cell, the minimum score is fixed at 4, which is the threshold below which less than one molecule of any essential metabolite is present in the cytoplasm.

[2] one molecule per cell corresponds to $1.6x10^{-24}$ mol. Given the cell volume (4 $\mu m^3$), this gives a concentration of $4x10^{-10}$ M = 4 Z.

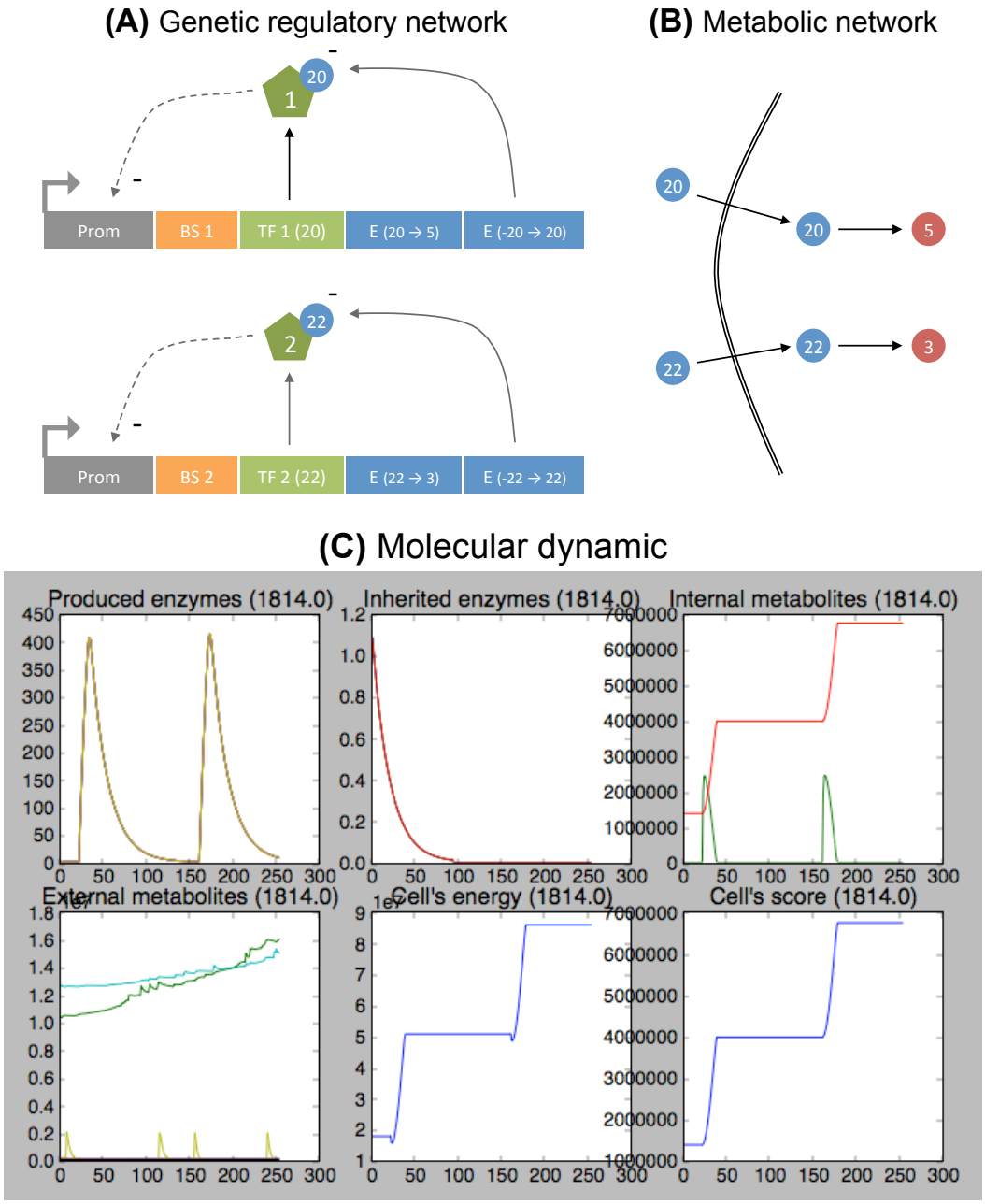


Figure 6: A simple model of operon regulation in Evo$^2$Sim. The genome contains two independent operons each producing an essential metabolite (the prime numbers 5 and 3) from two different carbon sources (respectively 20 and 22). **(A) Regulation network:** each operon self-inhibit itself unless it's primary metabolite is present in the environment. This ensures a minimal energy consumption in the absence of the energy source. **(B) metabolic network.** The metabolic network is rather simple: it contains two metabolic pathways, each dedicated to the production of an essential metabolite and each regulated by an operon. **(C) dynamic of the metabolic network.** The six graphs display the behaviour of a single cell in time (t = 0 corresponding to the cell birth). Each time one of the two carbon sources (20 ou 22) is present in the environment, the corresponding operon produced the corresponding pump and enzyme ("produced enzymes" panel). The cytoplasm then contains the carbon source (green curve on the "internal metabolites" panel) that is transformed into the corresponding essential metabolite (red curve on the same panel). Note the energy panel where the triggering of the operon is clearly visible (small energy drop corresponding to the operon transcription before the enzymes produce energy by degrading the carbon source).

**Results:**

Using this parameter setting, we tested a simple model of operon regulation (Figure 6) and immediately obtained a coherent (and viable) cell behaviour and evolution. Moreover, comparative evolutionary experiments between random parameters and the ones chosen here showed that evolution is more efficient and less unstable with the new setting. Even though this has to be confirmed experimentally, we hypothesise that this is mainly due to the new way to mutate the Michaelis-Menten parameters.

**Resources committed**

All bibliographic research was done by Carole Knibbe with the help of Hugues Berry. The Operon model was designed by Guillaume Beslon and Charles Rocabert, and tested by Charles Rocabert.

### 2.3.5. Conclusion

The development of the integrated model was more difficult than anticipated, as noticed by the project monitors during the first project review. However, we eventually developed a very promising model, finely tuned and available for conducting *in silico* experiments. In particular, it was used to replicate the LTEE evolutionary conditions and proved to replicate some of the results of the LTEE, thus facilitating the interpretation of this experiment (see (Rocabert *et al.*, 2016a,b) and section 2.4.5). Although WP2 ended at M18, the design of WP3 Evo$^2$Sim experiments triggered (and is still triggering) model evolution and optimization as well as new analysis and visualization tools. Thus Evo$^2$Sim was continually improved all along and the package distributed on the project website has been updated accordingly.

## 2.4. Workpackage 3: *In silico* experimental study of EvoEvo

### 2.4.1. Introduction

In WP3 we used the models developed in WP2 and the models developed previously by INRIA and UU to study the emergence of variability (plasticity) robustness, evolvability, and population level open-endedness in *in silico* evolutionary experiments. This enabled us to better understand and characterize the different "EvoEvo strategies" observed *in vivo* – or in some situations only *in silico* – and to understand how they could be simulated to be transferred to WP4 and WP5 (Figure 7). A detailed discussion of the most interesting results is given in Deliverables D3.1, D3.2, D3.3 and D3.4, following the narrative of the obtained insights. As discussed in these reports, one of the important insights obtained was the ways in which the crucial properties of EvoEvo are entangled. As stated in the conclusion of D3.2: "*Robustness and Evolvability are tightly linked: Robustness is an evolved property, one of the mechanisms for robustness is evolvability and one of the consequences of robustness is evolvability*". Moreover the mechanisms inducing variability, as well as the resulting variability in the population, are tightly linked to both the evolution of robustness and evolvability and are the result of these properties in the evolved population.

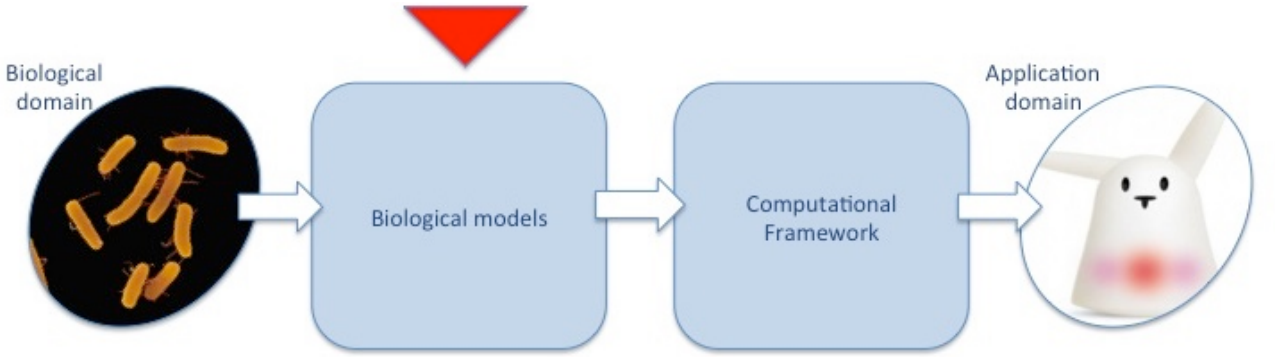


Figure 7: Position of WP3 in the biology-to-application scheme of EvoEvo

Despite this entanglement, in this section we follow the original proposal by summarizing the results on variability, robustness, evolvability and open-endedness separately, looking at each at the genomic, the phenotypic, the population and the ecosystem level.

### 2.4.2. Task 3.1: Evolution of variability

**Variability at the genome level**

Variability at the level of the genome is mediated by the genetic operators. We have elucidated the important role for genetic operators other than point mutations in evolution. In particular, large and small duplications and deletion, *i.e.*, genetic operators which modify genome size, are important mechanisms for EvoEvo. During the evolution of wild-types, genome size typically evolves in a predictable manner: initial genome expansion followed by streamlining (Cuypers and Hogeweg, 2012; 2014; Fischer *et al.*, 2014; Batut *et al.,* 2016a,b). "Typically" here means that it occurs especially in those runs which attain high fitness, and when there is no high penalty on genome size. Moreover, mutation rate should be proportional to genome size, as is the case in nature, but is not common practice in evolutionary computation. Evolution of genome size automatically tunes not only overall mutation rates per genome, but also the ratio of variability generated by different mutational operators, because this scales linearly for point mutations but super-linearly for the other operators in which not only the frequency but also the length scales with genome size.

**Variability at the phenotype level**

An arbitrary small amount of difference (*e.g.*, one point mutation) at the genome level can cause an arbitrary large amount of change at the phenotypic level, and an arbitrary large amount of genotypic difference can conserve the same phenotype. The mapping of genotype change (mutations) to phenotype change (GP map) is subject to evolution, and is in fact a primary mechanism of EvoEvo. We further discuss this below in terms of evolution of robustness and evolution of evolvability.

Moreover, given one particular genome, phenotypic variability (plasticity) can exist even simply by noise. Interestingly noise in the GP map enhances evolution of robustness and evolvability in developmental systems (Ten Tusscher and Hogeweg, 2011, Vroomans *et al.*, 2016). Phenotypic plasticity can evolve to increase functionality in constant environments, as we have shown in the case of alternative folding in RNA-based protocells (de Boer and Hogeweg, 2014), or to cope with variable environments, as we have shown for gene regulatory networks evolved in highly variable ecosystems, where an evolved stringent response allowed survival during periods of low resource

availability (van Dijk *et al.*, 2016a,b). When these evolved wild-types are brought into the experimental conditions of the LTEE, they tune by evolution this stringent response to the period of environmental change, a way which appears similar to what is seen in the LTEE experiments.

**Variability at the population level**

Maintaining sufficient population level variability is a prerequisite for effective evolution. It is a major concern in evolutionary computation. Embedding populations in space, and thereby limiting direct competition to the local neighbourhood, is an effective way to maintain variation and improve evolutionary adaptation. It is also biologically more realistic, and computationally more efficient than global competition. It is the default choice in all our experiments.

The relative amount of genotypic and phenotypic variation depends on the evolved GP map. High genotypic variation is maintained in the case of high neutrality, because the resulting low selection pressure does not weed out genetic variation, whereas conversely large phenotypic variability leads to low genotypic variability, because of high selection pressure. Evolution of the GP can lead to very counter-intuitive results, *e.g.* to much lower genetic variability at high mutation rates (Colizzi and Hogeweg, 2014).

**Variability at the ecosystem level**

How variability (diversity) at the level of ecosystems evolves and is maintained is a long-standing problem in ecology and evolution: why does not a “best” species evolve and out-compete all others owing to the principle of niche exclusion? Various mechanisms/conditions have been elucidated in our research. In Evo$^2$Sim we have shown that regular, *i.e.* seasonal, environmental variation, caused by periodic refreshing of the medium, can lead to speciation into two lineage which thrive in different seasons, where one grows on the primary resource, while the second one uses a secondary resource (produced by the first one) when the primary resource is exhausted (Rocabert *et al.*, 2016a,b). This speciation is very similar to that seen in a subset of the LTEE clones (Plucain *et al.*, 2014). In the *in silico* experiments it is stable only for regular fluctuations, as is indeed the case in the LTEE experiments.

Heterotrophic species can also evolve under conditions of spatial environmental variability generated by local variation of influx of different resources, which is however constant over time. Heterotrophy then allows the species to expand their range over the different environments. The spatial environmental variation is then not reflected anymore in the species composition as they occur intermingled in all environments (Meijer *et al.*, 2016).

Yet another mechanism to maintain ecosystem diversity is, counter-intuitively, horizontal gene transfer (HGT). By the evolution of differential rates of HGT for toxin and resistance genes high ecosystem diversity, with deep phylogenetic variation is maintained, despite the occurrence of super-killers (van Dijk and Hogeweg, 2016).

Multiple species ecosystems typically evolve in spatially embedded mutualistic systems, whereas these tend to go extinct in globally mixed systems due to the evolution of cheaters. Replicator systems are a preeminent example. We have shown, *e.g.* the evolution of rich multi-species ecosystems in spatially embedded version of the automata chemistry stringmol. The strains differ in replication (copying) mechanisms, and thereby the rates as well as the types of mutations

evolve. The result of the evolution is an increase in population size due to a decrease of parasitism.

These results are very relevant for ecological theory, providing examples that counter the default expectation of competitive exclusion. They are also very relevant for evolutionary computation because evolved ecosystem based diversity can also be exploited for problem solving by evolutionary computation (de Boer and Hogeweg, 2012).

### 2.4.3. Task 3.2. Evolution of Robustness

**Robustness at the genome level**

Organisms have evolved a plethora of repair mechanisms that protect genomes against mutations. Mutations (knock-out) of repair mechanisms lead to mutator strains. In the LTEE experiments such mutator strains evolved, which have a 100-fold higher mutation rate. The mutators were able to adapt faster to the LTEE conditions than non-mutator strains, but slowly lower their mutation rates again (Tenaillon *et al.*, 2016).

We studied the evolutionary dynamics of mutator strains by increasing (point) mutation rate 100-fold *in silico* evolved wild-type *E-coli*-like genomes (Rutten *et al.*, 2016a,b), as an interesting example system for EvoEvo. We showed that a lack of robustness at the genome level of the mutator strains leads to the evolution of even higher mutation rates, and especially large-scale duplications and deletions, through an increase of genome size, which nevertheless leads to the recovery of fitness. This is mediated by increased robustness and increased selection at the phenotypic level by mechanisms explained below.

Genome robustness and evolved repair mechanisms were also studied with respect to transcription-induced mutations in yeast (Colizzi and Hogeweg, 2016a,b). The evolved repair mechanism appears to decrease point mutation rate while increasing duplications and deletions. We showed that short-term selection pressures tend to lead to long-term genome deterioration, and thereby to loss in fitness without the repair mechanism. Even when the lack of repair increases overall mutation rate, the shift in type of mutations leads to more dynamic genome structure, but prevents the long-term deterioration.

**Robustness at the phenotype level**

Mutational robustness at the phenotype level refers to insensitivity to mutations at the phenotypic level, *i.e.* to neutrality. A classic example of EvoEvo is the increase of neutrality in long-term evolution in a constant environment when mutation rate is high enough, as was first seen in RNA evolution (Huynen *et al.*, 1996; van Nimwegen *et al.*, 1999). Such phenotypic robustness leads to high genetic variability and therewith slows better exploration of genotype space, and therewith increased, population based evolvability (Huynen, 1996) – see below.

Alternatively, and seemingly contradictory, evolution at high mutation rates can evolve low phenotypic robustness, but can maintain high fitness by low phenotypic robustness, because this leads to high selection pressure (Krakauer and Plotkins, 2002; Elena *et al.*, 2007).

An important addition to this basic theory is our discovery that prolonged evolution not only increases neutrality, but also increases large phenotypic changes by single mutations. In other words the mutational neighbourhood, represented in terms of the frequency (Y axis) of the amount

of phenotypic change (X axis) becomes “U-shaped”. (Cuypers and Hogeweg, 2012; Rutten *et al.,* 2016a,b). This U shape combines neutrality and high selection pressure, and therewith combines the best of both worlds. Skewing the U-shape in either direction is an important mechanism and consequence of EvoEvo.

Indeed the evolutionary dynamics of the mutator strain, leading to recovery of fitness deepens the U shape, *i.e.* increases neutrality and highly deleterious mutations, while decreasing slightly deleterious mutations (Rutten *et al.*, 2016a,b).

Robustness to environmental variation of a phenotype can evolve by evolution of gene regulation. We have studied this in the form of evolution of internal homeostasis under strong external resource fluctuations, with only implicit sensing of the variation in the internal environment. Given frequent enough environmental changes and provided that early genome expansion occurs, high degrees of homeostasis do evolve.

**Robustness at the population level**

Robustness at the population level can be maintained despite or due to low robustness at the individual level. This is achieved by high (individual) evolvability, by which the phenotypic composition of the population varies over time maintaining a stable population despite with environmental varying environmental demands. This strategy appears to occur in the virtual microbe model in complex, time varying environments (Cuypers *et al*., preliminary results).

**Robustness at the ecosystem level**

Robustness at the ecosystem level is discussed below (section 2.4.5). It is mediated by robustness and/or evolvability at the other levels

### 2.4.4. Task 3.3 Evolution of Evolvability

**Evolvability at the genome level**

A general mechanism increasing evolvability at the genome level is the tuning of non-coding intergenic regions. This apparent junk can form the substrate for novel genes, as analysed in detail in Aevol (Knibbe and Parsons, 2014). Moreover, by tuning the amount of intergenic regions, the relative frequency as well as the impact of different type of mutations is tuned. More intergenic regions increases genome length, without (or barely) increasing the effect of point mutations, whereas the frequency and length of large duplication's/deletions is increased. This mechanism is used to cope with high point mutations rates, increasing robustness and evolvability at other levels (Rutten *et al.*, 2016a,b) – see above.

Tight genomes, with only very small intergenic regions, evolve at very high per base mutations rates as in viruses (Beslon *et al.*, 2016a,b). Wild-types adapted to a certain environment tend to have exhausted all available small (*e.g.* point) mutations, and they seem evolutionary stuck. In those case innovations tend to occur in cascades after rare, not too deleterious duplications (or deletion), which deform the fitness landscapes and open up novel dimensions of variability and evolvability (Beslon *et al.*, 2016a,b).

Evolution of Evolvability by genome structuring is also observed in response to recurrent environmental changes, through transposon dynamics. Inverted repeats left by transposons create

hotspots for large duplication/deletion, and together with genome rearrangement biases mutation to being beneficial, as shown by Crombach and Hogeweg (2007), and as appears to be the case in yeast adapting to new resource environments (Dunham *et al.*, 2002). Likewise the replication fork barriers in between ribosomal genes appear to be an evolved to increase bias transcription induced mutations to be duplication/deletion of entire genes. We have shown that this increases evolvability as well as long-term robustness (Colizzi and Hogeweg, 2016a,b).

**Evolvability at the phenotype level**

Evolution of the GP map not only evolves the amount of phenotype change due to mutations, but also the specific changes that occur. Subjected to recurring environmental changes, the mutational neighbourhood evolves in such a way the very few mutations change the phenotype to match the alternative environments. This was first shown by Crombach and Hogeweg (2007, 2008) and is achieved either by genome structuring via transposon dynamics, or by switching between attractors of gene regulatory networks. The bias to be beneficial in the other environment is true for all types of mutations allowed. Thus, very fast evolutionary adaptation evolves, *i.e.* evolution of evolvability.

We now have shown, moreover, that the ability to regulate achieves phenotypic robustness to some type environmental fluctuation (*i.e.* homeostasis) strongly increases phenotypic evolvability to novel environmental changes, never seen before (Cuypers *et al.*, 2016). Repeating these changes increases evolvability even further, even if they occur very infrequently (up to once per thousand generations tested). This means that very high fitness is maintained despite these (drastic) environmental changes. Strikingly the average fitness due to phenotypic evolvability is similar with the average fitness due to regulation (plasticity), while plasticity is much harder to evolve, if it evolves at all, which it does only after many relatively frequent switches (Cuypers *et al.*, 2016).

**Evolvability at the population level**

Phenotypic mutational robustness, leads to high genotypic variability, and therewith to high evolvability of the population, as mentioned above.

### 2.4.5. Task 3.4. Evolution of open-endedness at the population level

In D3.4 a comprehensive review is given of different concepts and different levels of open-endedness. Here, we summarize our results on evolutionary systems in which no fitness function is defined *a priori* (or only partially defined, imposing only weak constrains) but only represented as what in fact does survive, and in which the evolving, replicating entities can either directly interact, or interact by changing the (local) environment. These are minimum requirements for potential open-ended systems. We used four different implementations of such potential open-ended systems, to study EvoEvo dynamics. These are: (1) the Evo$^2$Sim model developed in WP2 (Rocabert *et al.*, 2016a,b), (2) the virtual microbe model (VM model) (Cuypers and Hogeweg, 2015), which an extension of Virtual cell model (Cuypers and Hogeweg, 2012; 2014) (3) the RNA world models (Takeuchi and Hogeweg, 2008; Colizzi and Hogeweg, 2014), (4) spatial embedded stringmol (Hickinbotham and Hogeweg, 2016). In the former two models evolving entities have a PoaS-like genome and a metabolism, can evolve gene regulation and use up resources from the environment, and interact via the resources. The Evo$^2$Sim implements potential open-endedness by allowing an unbounded set of metabolites (*i.e.,* integers, where prime numbers provide

functionality) and enzymes (converting metabolites), and allowing unconstraint mutation of enzymes. In contrast the VM model implements a restricted universe of potential metabolites and enzymes and constrains mutations of enzymes into each other. In the latter two systems minimal replicators interact via complementary binding, and evolve GP maps.

**Ecosystem robustness prerequisite for ongoing evolution, and is strongly facilitated by spatial self-organization**

For ongoing (open-ended) evolution a prerequisite is survival of the eco-evolutionary system. Spatial embedding, and thereby spatial pattern formation, is often a prerequisite for survival, i.e. for preventing the system to evolve to extinction. Such evolutionary extinction is common in non-spatial models requiring cooperation for survival. For example, only after embedding StringMol model in space, could its evolutionary potential unfold. As detailed in D3.4 it shows a rich spectrum of innovations through EvoEvo: new strategies of robustness against cheaters, evolving new types of mutations through copying errors, e.g. whole genome duplications, converting templates to primers, i.e. copying self rather than the template etc. (Hickinbotham and Hogeweg, 2016)

Moreover spatial pattern formation leads to multilevel evolution. Multilevel evolution, whether generated automatically through spatial pattern formation or through the prior definition of multiple levels as e.g. (proto)cells, does not only protect against extinction, or lead to a compromise of opposing selection pressures, but actually open up a gateway to the evolution of complexity, and ultimately open-endedness, as discussed in Takeuchi et al. (2016a,b). A pre-eminent example is our demonstration of the evolution DNA in the RNA world, *i.e.* the separation of information storage and usage, one of the major transitions in biological evolution (Takeuchi *et al.*, 2011).

**Evolvability mediates ecosystem robustness**

Stability (robustness) of diverse ecosystems is a long-standing problem on ecology. On-going evolution is an often-ignored stabilizing factor. We have shown in the RNA world models that too low mutation rates lead to extinction even in spatial embedded systems (Takeuchi and Hogeweg, 2008), and that the evolved mutational neighbourhood, in which there are extreme phenotypic difference between close mutants (*i.e.*, high phenotypic evolvability) cause ecosystem robustness at very high mutation rates (Colizzi and Hogeweg, 2014).

As we have seen above, evolvability is relatively easy to evolve. In the metabolic based ecosystems high levels of evolvability (and/or speciation) appear to prevail as strategy to cope with variable environments, over individual plasticity through gene regulation. No persistent gene regulatory networks were found in the Evo$^2$Sim model, possibly due to the inherent high evolvability. In the VM model plasticity did evolve, but this appears to require relatively simple environments, and strong energy limitation. Similar to what we saw above, evolved plasticity enhances evolvability to changed environmental conditions, *e.g.* the conditions of the LTEE experiment (van Dijk *et al.*, 2016a,b).

Our experiments have also alerted us to the fact that evolving (and maintaining) complex gene regulatory networks does not imply plasticity and physiological adaption to environmental variation. Its main functionality appears to be the suppression of gene expression, and thereby conveys robustness to addition of new genes by duplications or HGT. This would support the conjecture that such (initial) suppression is the common in genes required by HGT, and are retained in biological populations.

**Speciation mediates ecosystem robustness**

Speciation evolves readily in all the potential open-ended systems, and helps to maintain ecosystem robustness/persistence despite spatial and temporal imposed environmental variability, and/or against short-term selection pressures leading to extinction. For example the evolution of parasitic (cheater) lineage prevents evolutionary extinction in simple RNA systems (Hogeweg and Colizzi, 2016c).

Moreover speciation into a lineage that uses the leftovers of the other species evolved in the Evo$^2$Sim model to cope with the imposed LTEE-like seasonality (Rocabert *et al.*, 2016a,b), similarly to what is seen in (some of the) *E. coli* LTEE experiments. In the VM models, speciation does not occur when plasticity evolves (van Dijk *et al.*, 2016a,b), and the resulting phenotypic robustness is sufficient to cope with the imposed temporal environmental variation of LTEE-like condition. We have referred to speciation *vs* plasticity as ecosystem based *vs* individual based diversity or complexity (de Boer and Hogeweg, 2012), in the context of problem solving by evolutionary computation).

Speciation also occurs in spatially variable, but temporally constant environment. This is not surprising as the species can simply adapt to the different niches. More surprising is the evolution of obligatory heterotrophy in such system, *i.e.* the evolution of species that (mutually) need products produced by the other. This leads to evenly spread species over the whole environment in such a way that the environmental niches are not reflected in the ecosystem anymore (Meijer *et al.*, 2016), and a very stable ecosystem persists.

### 2.4.6. Conclusions

By using various approaches, and through many discussions, the groups at INRIA and UU working together in WP3 have been very successful in elucidating general principles of how Darwinian evolution shapes the variability, robustness and evolvability of genomes, individuals, populations and ecosystems, as well as the understanding of the entanglement of these processes. In collaboration with UGA and CSIC we developed digital equivalent of the laboratory experiments conducted in WP1. Preliminary analyses led to a generalization of the *in vivo* observation, in the context of the *E. coli* LTEE (Rocabert *et al.*, 2016a,b; van Dijk *et al.*, 2016a,b; Rutten *et al.*, 2016a,b) as well as in the context of the TEV (Beslon *et al.*, 2016a,b). Overall we have shown how EvoEvo arises even in relatively simple models, and often leads to counter-intuitive consequences that seem to contradict survival of the fittest while they simply indirectly result from it.

## 2.5. Workpackage 4: A computational EvoEvo framework

### 2.5.1. Introduction

While WPs 2 and 3 developed and produced models of EvoEvo mechanisms and approaches, focused on biological processes, the objectives of this WP4 were to take those biologically-oriented outputs, and develop suitable computational *analogues*, to form the basis of a novel route to open evolved computational and engineered systems (Figure 8).

The development route was via a computational meta-model. This is essential for developing a coherent bio-inspired computational approach (Andrews *et al.*, 2011). Developing a computational model directly from a biological model runs the risk of confusing biological-contingent detail (the

existence of three pathways, say) with the underlying principle (the existence of multiple pathways refined for particular purposes), leading to a rigid, over-constrained, and naïve implementation. The meta-model route instead exposes the underlying principles, abstracts away from the irrelevant details, and results in a more flexible computational analogue.

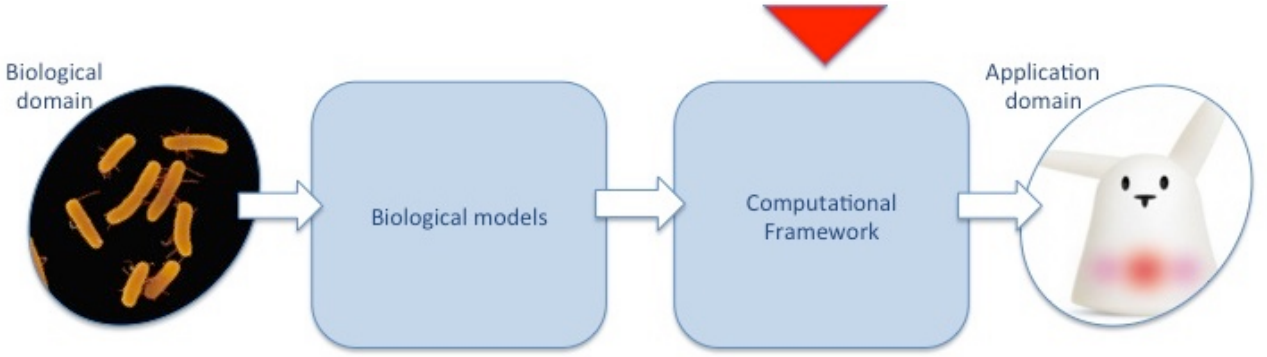


Figure 8: Position of WP4 in the biology-to-application scheme of EvoEvo

This section present the contributions of WP4 organized by tasks. Section 2.5.2 presents the modelling work that re-interpret the biological and computational models used/developed in WP1, WP2 and WP3 in the CoSMoS approach. Section 2.5.3 presents the development of the computational framework (EvoMachina) based on this model. Finally, section 2.5.4 goes one step further and proposes new directions for a *bio-reflective architecture* able to fully implement the EvoEvo concepts.

### 2.5.2. Contribution 1 (task 4.1): EvoMachina meta-model and model

We took input from WP2 and WP3 (code, literature, partner discussions) as our starting point. We applied the CoSMoS approach to this material, and reverse-engineered a CoSMoS Domain Model from the Lyon INRIA partners' Aevol code, and identified the key unifying concept of *machine* on which to base the modelling. See (Andrews and Stepney, 2014). We engineered a prototype python implementation of the core Aevol code, then *refactored* it to support the machine concept, as a means of validating the model. As a spin-off from this work, we also identified a further component needed in some uses of the full CoSMoS approach (Andrews and Stepney, 2015b).

We then applied the meta-model concept from the CoSMoS approach, using it to ensure an underlying coherence between specific models (particularly the biological domain and the computational platform). Here we refined the key machine concept, and clarified its central role in the architecture. We included the structures necessary to allow machines to be encoded on the genome, thereby allowing them to evolve, by generalising the PoaS genome model from the Utrecht partners. See (Andrews and Stepney, 2015a).

This computational meta-model is described fully in D4.1, and an extensive computational model requirements specification is given in D4.2. Together, these lay out the specification for a novel evolutionary algorithm, dubbed ***EvoMachina***, based on two main concepts from EvoEvo:

1. That *genomic reorganisation* is an important factor in the evolution of evolvability.
2. That the machinery of evolution (expression, replication, etc) is implemented by *machines* that are themselves encoded on the genome, and hence are themselves subject to evolution.

The conceptual model is summarised in Figure 9 (some components omitted for clarity).

The machine concept has proved valuable in structuring the development of EvoMachina. We have been guided by the biological inspiration, but have also, in the manner of all bio-inspired algorithms, deviated from emulation of biology where appropriate. The first key insight is to use machines (active entities) to implement the mechanisms of the system, in particular, to establish the relationships between the various passive information-bearing and evolvable components. The second key insight is to allow those machines to be encoded in the information-bearing components, so that they are subject to evolution.

**Resources committed**

Susan Stepney provided the project management. Paul Andrews (month 1-18) developed the model and specification.

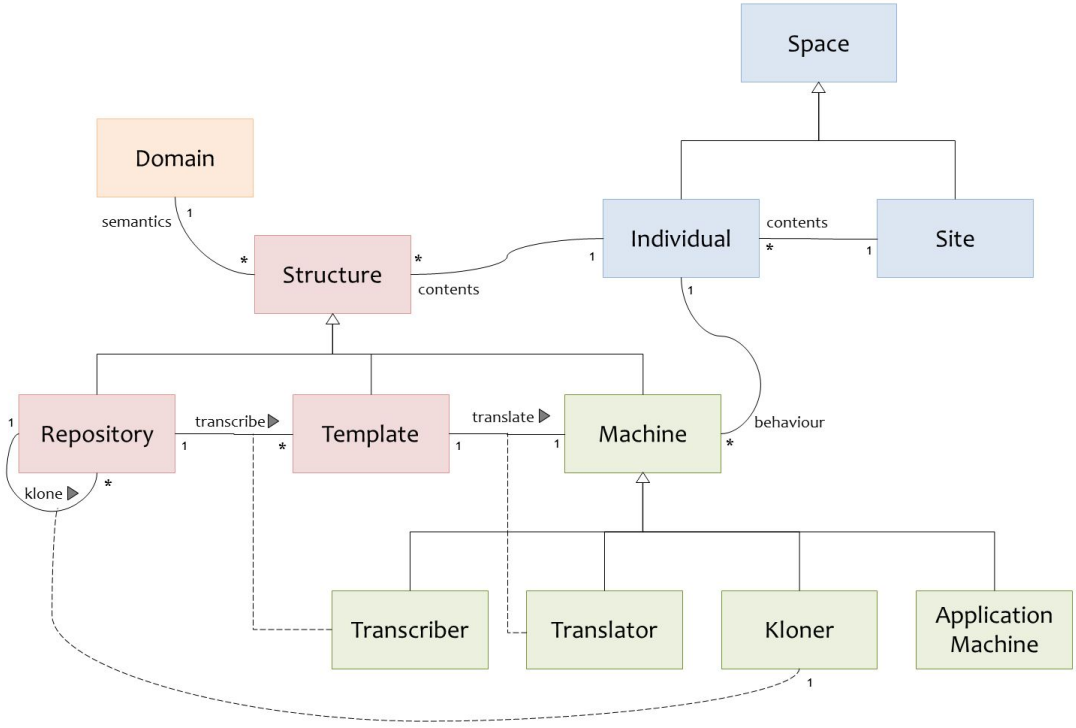


Figure 9: Conceptual model of EvoMachina: see D4.1 and D4.3 for details.

### 2.5.3. Contribution 2 (task 4.2): EvoMachina Java framework implementation

We used the meta-model and requirements specification from D4.1 and D4.2 to develop an object-oriented Java executable framework, implementing EvoMachina. See (Hoverd and Stepney, 2016) and D4.3. We extended the original specification to allow multiple “genomes” of different types (different Repository subclasses). This proved useful to separate concerns of different types of evolvable machines, in particular, the evolution of the “candidate solution” and the evolution of the Kloner “mutator” machine. D4.3 provides a description of **EvoMachina**, the Java framework that implements this specification.

EvoMachina code is available from github.com/evoevo-york/evomachina

The current release of EvoMachina (v2), includes two example applications, one of the classic TSP optimisation problem, and one of the ChameleoClust subspace clustering algorithm (where the genome "pearls" are tuples of numbers representing parts of cluster centres) as developed in EvoEvo for use in WP5 (see section 2.6.2 for details of the algorithm).

In these examples the mutation operators and mutation rates also themselves mutate, by having key components and parameters encoded in evolvable mutation genomes. (See Figure 10.)

- TSP
  - problem domain Repository: "pearls" are individual city locations
  - mutator domain Repository: "pearls" are values of k for k-opt operations
- ChameleoClust
  - problem domain Repository: "pearls" are tuples of numbers, representing (parts of) cluster core-point locations
  - mutator domain Repository: "pearls" are mutation, deletion, and duplication rates

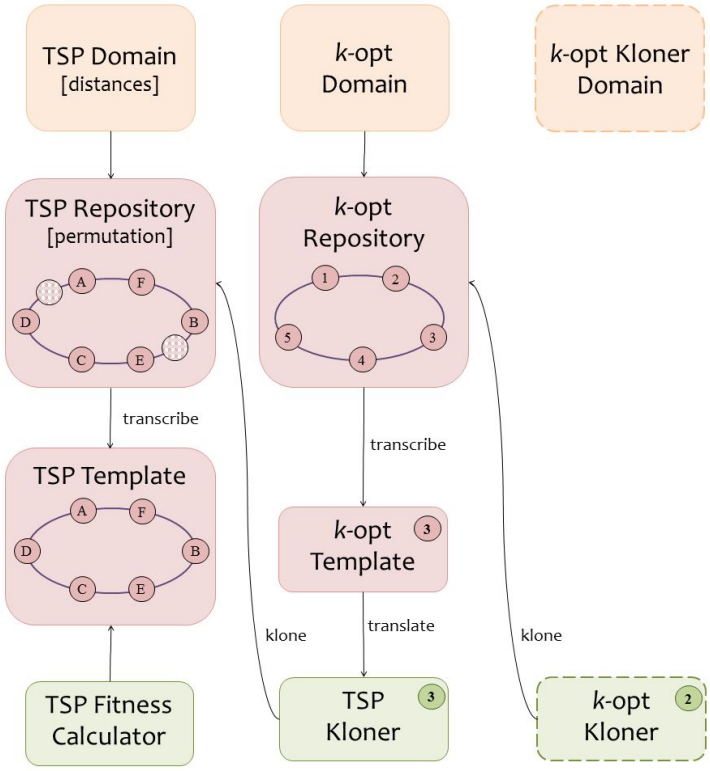


Figure 10: Instantiation of an EvoMachina Individual for the TSP: see D4.3 for details. (Some machines are omitted for clarity.)

EvoMachina uses several EvoEvo properties identified in WP1 and WP3, some of them having previously been tested in WP2. However, it also adds new features that enable the use of these EvoEvo properties in a broader application context:

1. EvoMachina allows multiple different types of genomes, allowing appropriate representations and machinery to be used for different parts of the application. For example, the mutation machinery can be encoded in a separate genome, allowing the mutation operators to evolve in a different manner, and at a different rate, from the application's candidate solutions.
2. An important realisation was that it is not necessary for EvoMachina to faithfully emulate biology by encoding the totality of a machine on the genome. EvoMachina allows some parts of each machine to be hard-coded, and hence un-evolvable, and other parts to be encoded on the genome, and hence evolvable, again in an application-dependent manner. For example, in the TSP application, the Kloner mutation machine implements a *k*-opt reorganisation of the city permutation: the value of *k* is encoded on the mutator genome allowing evolution, but the actual algorithm from reorganising the genome is hard-coded, and so protected from evolution. This allows a suitable separation of concerns, and supports more flexible mutation possibilities.

In addition to the two example applications, the EvoMachina framework comes with a variety of evolutionary variants: classic EA, microbial GA, as well as the EvoMachina specific operators, and with a variety of spatial options, including an aspatial (well-mixed) option and a 2D toroidal grid.

This implementation demonstrates that EvoMachina can support classic EAs, but also has more flexibility to support the *evolution of evolution*, by also providing concepts and mechanisms that support the evolution of the evolutionary machinery itself.

**Resources committed**

Susan Stepney provided the project management. Tim Taylor (month 19-24) provided validation of the models, and investigations of appropriate physics engines. Tim Hoverd (month 25-36) implemented the EvoMachina Java framework and developed the example applications.

### 2.5.4. Contribution 3 (task 4.3): Bio-reflective architecture and applications

We have argued that computational reflection is an essential component of computational novelty generation (Banzhaf *et al.*, 2016). In this task we developed a new **bio-reflective architecture** (Hickinbotham and Stepney, 2016a). It is a synthesis of concepts from:

1. von Neumann's Universal Constructor Architecture
2. procedural computational reflection
3. evolutionary algorithms
4. computational open-ended novelty mechanisms
5. the EvoMachina architecture of evolvable active machines and passive genomic structures

Parts of this architecture were realised in the stringmol automata chemistry, and various experiments were run to demonstrate its worth (Hickinbotham and Stepney, 2015a,b; Clark *et al.*, submitted), see D4.4 and D4.5.

Additionally, parts of the architecture were realised in other implementations, using Domain Specific Languages. In particular, a stand-alone evolutionary music application was developed (Hickinbotham and Stepney, 2016b), see D4.5, and was also integrated with the WP5 dance application (Abernot *et al.*, 2016).

This *bio-reflective architecture* (Hickinbotham and Stepney, 2016a), for the first time, incorporates concepts from computational reflection into an *evolving* computation system. Automata chemistries provide an ideal vehicle for implementing these at a low level. A higher-level approach, exploiting the EvoMachina architecture (D4.3), fits into the same model: EvoMachina's ability to separate the evolving and fixed parts of machines removes the fragility of attempting to evolve everything.

The main experiment of encoding a bio-reflective system into the stringmol language demonstrated *semantic closure* – the changing of the *meaning* of a genome without changing its *text* – for the first time in an evolving computational system (Clark et al., submitted).

Additionally, a combined research with Utrecht (WP3) based on this approach has demonstrated the clear role that space plays in the robustness of evolving systems, by permitting the system to survive parasites (see section 2.4.5).

**Resources committed**

Susan Stepney provided the project management. Simon Hickinbotham (month 13-36) designed the architecture and ran the experiments.

### 2.5.5. Conclusion

We have developed a novel evolutionary algorithm, ***EvoMachina***, by analysing and abstracting the relevant EvoEvo biological concepts, and have demonstrated its capabilities in two applications, TSP and subspace clustering. The new architecture supports the *evolution of evolvability* in an *evolutionary algorithm* for the first time, by allowing genome reorganisation, and by allowing the machines of evolution to evolve.

We have augmented EvoMachina's directly bio-inspired approach to evolutionary algorithms with another novel architecture: the ***bio-reflective architecture***. This is inspired not only by biology (compatible with the EvoMachina framework), but also by computational concepts, specifically that of computational reflection. We have realised this architecture on the stringmol platform, and have observed a form of ***semantic closure*** for the first time in an artificial system.

By designing, implementing, and executing these novel architectures, we have demonstrated how close cooperation with the leading edge of wet-lab biological and computational biological research can inspire sophisticated next-generation computational algorithms.

## 2.6. Workpackage 5: EvoEvo applications

### 2.6.1. Introduction

WP5 constitutes the final step of EvoEvo. Its objective was to build proof-of-concept applicative software that use the theoretical and practical outcomes of the 4 previous WPs (Figure 11). Embryonic examples of living technologies have existed for more than 15 years – see, *e.g.* (Sims, 1994; Funes and Pollack, 1999; Lipson and Pollack, 2000) – but they have never really demonstrated usability or feasibility. Apart from the technological difficulties of building living technologies, this is due to the lack of real applications and to the limit of the toy-problems used to demonstrate the capacities of these technologies. In the EvoEvo project, we decided to directly concentrate on a real application, but we carefully designed it such that its difficulty should be manageable in the context of the project. Our objective here were to design living technologies

able to manage the complex, unstable and unstructured flux of information produced by smart sensors in order to enable intelligent agents (here personal companions) to adapt to their usage context.

The new wireless sensor technologies have been at the origin of a profound shift in the concept of smart houses, smart buildings and smart cities, since the sensing structure now evolves faster than the structured of the sensed system. On the one hand, this situation creates huge information management difficulties since the sensing system cannot be modelled before it is used. On the other hand, the generated flux of information enables the monitoring of the system with a quality of service never before achieved. The proof-of-concept applications of EvoEvo directly follow from these two points. Our aim was to create a software infrastructure able to manage this flux of information. This Infrastructure was tested in real situations (*i.e.*, while interacting with naïve users). The objectives were:

- To generate a stable model of the environment, despite the evolution of the sensing network. The biological analogy here is the notion of the circadian cycle, which is maintained and used in most living creatures despite the parallel and different evolution of their biological sensors. This application was tackled in task 5.1.
- To design intelligent agent(s) that are able to use the information flux produced by the sensors to learn and invent actions in accordance with usage and users. These intelligent agents can thus become personal companions of their users and progressively adapt to them so that their presence becomes first acceptable and second useful. These agents were designed in task 5.2. Note that while the general context is smart sensors embedded in houses, buildings or cities, the application developed in task 5.2 concentrated on body sensors as they enable easier testing sessions than smart buildings. This choice was proposed by the HiKoB company (our sensor provider) and was validated during the two first project reviews (see deliverable D6.4).

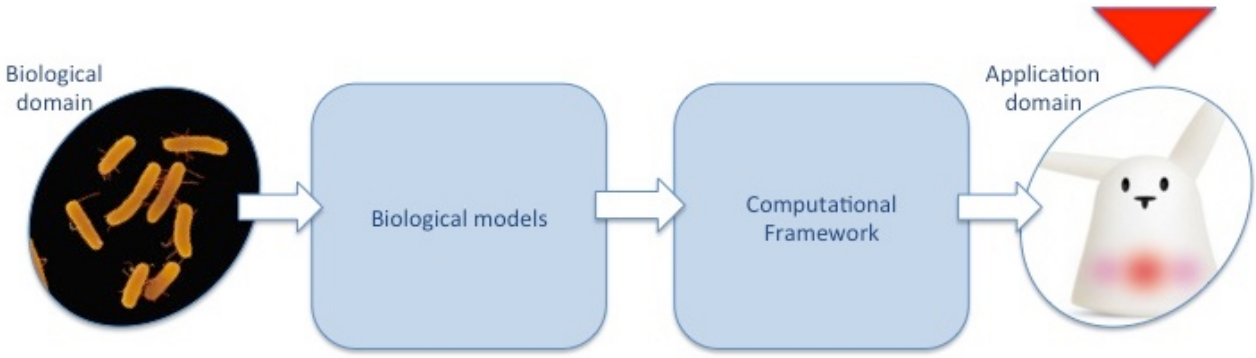


Figure 11: Position of WP5 in the biology-to-application scheme of EvoEvo

While WP5 was organised in two tasks, we organize this section in three contributions. The first (section 2.6.2: Contribution 1: An evolutionary subspace clustering algorithm exploiting EvoEvo strategies) corresponds directly to task D5.1, while the two remaining correspond to two different parts of task 5.2: the development of the adaptive software companion (section 2.6.3: Contribution 2: Applications) and its testing with real users (section 2.6.4: Contribution 3: Testing living technologies). Even though this last contribution was integrated in task 5.2 in the DoW, it resulted in a very specific work.

### 2.6.2. Contribution 1: An evolutionary subspace clustering algorithm exploiting EvoEvo strategies

Clustering is a data-mining task that aims to group objects sharing similar characteristics into a same cluster over the whole data space. Usually similarity between objects is determined using a distance function. Subspace clustering purpose not only implies identifying groups of similar objects, it also aims to detect the subspaces where similarity occurs. Subspace clustering can be conceived as "similarity examined under different representations" (Patrikainen and Meila, 2006). It is for this reason that subspace clustering is recognized as a more complicated and general task than standard clustering. Moreover, retrieving meaningful subspaces is particularly useful when dealing with high dimensional data (Kriegel *et al.*, 2009).

Many approaches have been investigated for subspace clustering in the literature using various clustering paradigms. The reader is referred for instance to (Kriegel *et al.*, 2009; Müller *et al.*, 2009; Parsons *et al.*, 2004) for detailed reviews and comparisons of the methods, including the main categories: Cell-based, density based, clustering-oriented approaches and also pattern-based clustering or bi-clustering approaches, *e.g.*, (Sim *et al.*, 2013). The two algorithms developed in EvoEvo belong to the clustering-oriented category; such approaches are based on parameters specifying properties of the targeted clustering such as the expected number of clusters or the cluster average dimensionality. According to these constraints, the objects are grouped together mainly using distance-based similarities. Most of these methods tend to build center-based hyperspherical shaped clusters.

Even though many evolutionary clustering approaches exist (Hruschka *et al.*, 2009), very few of them address the subspace-clustering problem. Two earlier approaches presented in (Sarafis *et al.*, 2003) and (Vahdat *et al.*, 2010) require non-evolutionary steps to identify clusters in lower dimensional spaces. The two algorithms developed in EvoEvo, called ChameleoClust+ and SubCMedians, are both a single stage fully evolutionary approach, without any preliminary stage to identify clusters in lower dimensional spaces. Both algorithms take advantage of EvoEvo by means of an evolvable genome structure to tackle the subspace-clustering problem. The key underlying principle is to use such an evolvable genome structure to find various numbers of clusters in subspaces of various dimensionalities.

The first algorithm, ChameleoClust+, has a coarse-grained genome defined as a list of tuples of numbers, and containing a variable proportion of non-functional elements (similar to the genome used in Evo$^2$Sim, see section 2.3.3). This genome is mapped at the phenotype level by using the genome tuples to denote core-point locations in different dimensions, which are then used to build the subspace clusters. During replications the genome undergoes both local mutations and large random rearrangements, namely: large deletions and duplications. Local mutations modify the genome elements and rearrangements modify the genome length and the proportion of non-functional elements. The key intuition in the design of the ChameleoClust+ algorithm is to take advantage of such an evolvable structure to detect various numbers of clusters in subspaces of various dimensions. In addition, ChameleoClust+ takes advantage of the genetic memory through evolution to evaluate the fitness over a sliding dataset sample, leading to an important reduction of the execution time, without effective degradation of the clustering quality. ChameleoClust+ was compared to state-of-the-art algorithms on both real and synthetic datasets using the evaluation framework presented in (Müller *et al.*, 2009). The experiments show that ChameleoClust+ obtains competitive results with a single parameter related to the domain, *i.e.*, the maximal number of

clusters. A sensitivity analysis carried out by varying the main parameters one-at-a-time, revealed that the impact of the parameters related to the evolution strategy (population size, mutation rate...) are low for a large portion of the parameter space. Further analysis revealed that the elitist reproduction method used in this algorithm ensured slightly better results for the majority of the experiments presented here. Finally a deeper analysis of the impact of the presence of the non-functional elements showed that the subspace clustering quality increased considerably when non-functional elements were incorporated in the algorithm. ChameleoClust+ is available on the project website at (http://evoevo.liris.cnrs.fr/chameleoclust/).

The second algorithm, SubCMedians, was inspired by ChameleoClust+. As before, each individual encodes a candidate subspace-clustering model and selected individuals are copied and mutated to produce the next generation. However, in SubCMedians, the phenotypic description (core-point locations) is explicitly encoded, and thus the algorithm requires fewer operations, since it does not need to decode genomes to produce the phenotypic description. Moreover, in order to achieve a more efficient exploration of the space of possible models, SubCMedians uses data objects themselves to build and adjust each core-point's coordinates so that they approximate the locations of the median of their corresponding cluster. The explicit representation of the phenotype and the use of a median based approach lead to significant reduction of the runtimes and allowed tackling the subspace clustering of data streams on-the-fly. In addition to the explicit representation at phenotypic level, the design of SubCMedians also incorporates an evolvable representation at the genotype level in order to allow for evolution-of-evolution. The genotypic description stores the number of genes that are involved in the construction of each core-point location along each dimension. In this representation, the number of genes used to encode each coordinate can vary without a direct impact on the subspace clustering model, and individuals can share the same phenotypic description yet have different genome structures (different number of genes associated to each core-point location). An important contribution of SubCMedians is to show that a median-based subspace clustering approach can exhibit satisfactory results when compared to well-established subspace clustering paradigms. Moreover, since no subspace clustering technique based on medians has been reported in subspace clustering techniques reviews, *e.g.*, (Kriegel *et al.*, 2009; Müller *et al.*, 2009), nor in recent subspace clustering proposals, *e.g.*, (Wang and Zhu, 2015; Liu *et al.*, 2013), the SubCMedians algorithm is a good candidate as a complementary tool, in particular for users interested in the properties of medians themselves (facility location, robustness to noise and outliers).

The interest of the flexible genome structure, controlled by evolution itself, to tackle the subspace-clustering problem are discussed in (Peignier *et al.*, 2015a), and a preliminary version of ChameleoClust+ received a best paper award in the category Evolutionary Machine Learning at the conference GECCO 2015 (Peignier *et al.*, 2015b). This preliminary version has been implemented in EvoMachina (see section 2.5.3).

**Resources committed:**

Sergio Peignier, Anthony Rossi, Christophe Rigotti, and Guillaume Beslon designed and implemented ChameleoClust+ and SubCMedians. They also ran the experiments to evaluate these algorithms.

### 2.6.3. Contribution 2: Applications

**EvoWave: classifying Wi-Fi context**

ChameloClust+ was tested with a proof-of-concept application on a dynamic data stream. Even though the problem addressed is still a subspace-clustering problem, new difficulties are added by the context of a dynamic stream. The number of classes may change over time; the location of the classes is also susceptible to change over time; the descriptors of the incoming objects are also susceptible to change (*i.e.*, features can appear or disappear). The real-world data stream used for these experiments is the Wi-Fi environment in which any ICT device is immersed: the strength of the signal from every Wi-Fi antenna in the neighbourhood. This environment depends especially on available routers and other computers, so it is linked to the context of use of the computer: work, teaching, house, etc. If the Wi-Fi signals from different contexts are dissimilar enough from each other, we expect that ChameleoClust+ should be able to discriminate different contexts from the data. This corresponds to a dynamic stream problem as new classes (*i.e.*, context of use of the computers) are appear or disappear at any time, and also because Wi-Fi antennas are never the same in different contexts (features appear and disappear). Moreover, this application is also challenging regarding the high dimensionality and noise level of the data.

The results achieved by ChameleoClust+ on this dataset suggest that the genome structure of the organisms was able to adapt to the changes in the data stream. The same applies to the subspace-clustering model of the best individual, indeed the number of clusters and the cluster dimensionalities evolve along the data stream. Despite the changes in the data stream, the quality of the subspace clusters produced by the individuals tended to remain interesting. The algorithm seems to be adaptable enough to cope with the changes in the data stream by adapting the subspace clustering models encoded in the genomes. Nevertheless we also observed that the genome structure often contained far more genes than those needed to encode the model, this phenomenon involved particularly core-points that are built but that corresponded to empty clusters. Such core-points often had high dimensionalities and many genes involved in their description and appeared to be inappropriate while dealing with such a dynamic environment. In order to limit the accumulation of many dimensions in empty (useless) clusters we proposed to limit the promotion of non-functional elements to functional ones. Further experiments were run to gather preliminary evidence regarding the effect of such a modification. These experiments showed that the modification lead to higher quality measures most of the time. This suggests that individuals were able to adapt more quickly to the changes in the stream because they had fewer core-points corresponding to empty clusters with high dimensionalities.

This application of ChameleoClust+, called EvoWave, is described in deliverable D5.1, and the corresponding software components are distributed as a companion archive, available at:

http://evoevo.liris.cnrs.fr/download/4_-_deliverables/wp5/Deliverable_D5.1_software_archive.zip

**EvoMove: a musical personal companion**

To explore further the potential of real-world applications of EvoEvo inspired technologies we developed a prototype of a musical system to produce music on-the-fly from the moves/dance of a performer. This system is named EvoMove, and its hardware is composed of wearable motion sensors, an acquisition gateway, a move recognition unit and a sound generator. The EvoMove system uses SubCMedians as a move categorization/recognition subsystem, and is fully described

in deliverable D5.2. Its use gave preliminary evidence regarding the capability of the system to handle real noisy data on-the-fly while continuing to evolve over this motion stream and constantly adapting to handle such an open-ended context. Real organisms have to constantly adapt their existing functions to new environments. For instance, predators have to adapt their capabilities (speed, teeth, claws, etc) to their co-evolving prey. In the EvoMove system, functions are the identification of clusters and typical changes are for instance a new cluster appearing when a new move is introduced by the user, or a cluster drift if the user decide to modify one of their moves. Thanks to its adaptable nature the system is able to follow such modifications while keeping in its genome memories of moves that are not seen during several generations so as to be able to identify them again when they reappear. This can be observed on videos of working sessions made available at

https://www.youtube.com/watch?v=p_eJFiQfW1E

and

https://www.youtube.com/watch?v=E85B1jJOiBQ

The use of the system as a core part to build a commensal architecture for evolving living instruments is introduced in (Abernot *et al.*, 2016).

**Resources committed:**

Sergio Peignier, Jonas Abernot, Leo Lefebvre, Anthony Rossi, Christophe Rigotti, and Guillaume Beslon designed and implemented the EvoWave and EvoMove systems.

### 2.6.4. Contribution 3: Testing living technologies

A third contribution of the WP5 was to organize real working sessions with the EvoMove system. These tests were performed with people having different backgrounds and different approaches to the system, ranging from people who were the developers of the project themselves, to professional dancers and musicians. Several different sets of sample sounds were created for these sessions, in order to have various musical atmospheres. At the end of the sessions we asked the performers about the way they perceived the system and also their interactions with it.

Two half-day sessions were organized with the dance company Anou Skan, and two of its dancers were filmed. As mentioned in the previous section, two videos recorded during these sessions are publicly available at https://www.youtube.com/watch?v=p_eJFiQfW1E and https://www.youtube.com/watch?v=E85B1jJOiBQ . About ten sessions were organized with Claire Lurin, an engineering student and semi-professional dancer, and a project of solo dance piece using the EvoMove system is currently in progress. Several other informal sessions also took place with users who are not professional dancers, including a two hour collective session with the members of the Beagle INRIA team. A session was also organized with a violinist, who gave very different feedback from what was given by the dancers.

What we learnt from the discussions we had with these different users, is that they found the system easy to use, but had very different representations of what the system was doing, and they thought/spoke about the system in very different ways. Here are three examples of these mental representations and feeling about the interactions: (The descriptions given here have been collected from different users after their first trial of the system)

- A musician: Felt like being involved in a teaching relationship with the system, trying to repeat gestures for the machine to memorize, trying to insist on some distinctions between gestures, as if he were teaching a trick to an animal.
- A dancer: Perceived the system as a good monitoring tool for her moves, that was noticing small move differences by playing a different sound, as if a specialist (a dance teacher) were checking the correctness of her moves.
- One of the developers of the system: As the system is based on a clustering algorithm dealing with sets of points and subspaces, he thought about it in a geometrical way. His perception, when using the system, was of placing points in a multidimensional space, so as to push and pull candidate cluster centers in this space.

Furthermore, they each imagined very different usages/applications of the system, ranging from its use as a dance improvisation companion, to its use as an electronic instrument that could be mastered to control what sound will be played by the system and when. An interesting point is that they all perceived the adaptation capability of this EvoEvo inspired technology, and were able to experiment with it, without the need to adopt a common cognitive representation or to have dedicated training.

**Resources committed:**

Jonas Abernot and Christophe Rigotti organized these EvoMove working sessions.

### 2.6.5. Conclusion

WP5 was clearly a bet when we proposed the EvoEvo project. Three years ago, nobody could be sure that the results of the project would produce efficient applications. Now, at the end of the project we can state that not only has it done so, but more than that, that the users who tested the EvoMove application were immediately captivated. It is probably too early to affirm that this interest will last long enough to see public applications, but our exchanges with musicians, dancers and physicians on this system are all very promising.

In a sense EvoMove is now an autonomous entity that is being used by dancers to co-create plays. We are all eager to see the results in a very near future.

## 2.7. Workpackage 6: Project management

### 2.7.1. Introduction

The project management was described in the DoW document (WP6 and section B2). WP6 was composed of four tasks:

- Task 6.1: Consortium Management and project monitoring
- Task 6.2: Administrative & Financial Management
- Task 6.3: Internal dissemination
- Task 6.4: Interdisciplinary dissemination

Partner 1 (INRIA) was in charge of these four tasks. The project coordinator (Guillaume Beslon) was directly in charge of tasks 6.1, 6.2, 6.3 and 6.4 and used the services of INRIA financial and legal affairs for task 6.2. Since administrative and financial management was reported in the

second review report, this document is structured as three contributions: project monitoring and management (section 2.7.2), dissemination (including internal dissemination – section 2.7.3) and the organization of the two EvoEvo interdisciplinary workshops (section 2.7.4). Although these workshops were dissemination activities, they are specific enough to deserve a dedicated section.

### 2.7.2. Contribution 1: Project monitoring and management

The project monitoring was based on regular exchanges between all project members during the general project meetings, bilateral exchanges between two members (sometimes three) for technical discussions, and email exchanges (*e.g.*, for deliverables coordination). In particular, we had seven "all hands" project meetings during project:

- **Kick-off meeting (Lyon, October 28-29 2013).** Attendees: Paul Andrews (Univ. of York), Laetitia Arnould (UJF, Legal affairs), Guillaume Beslon (INRIA), Cecile Cornu (INRIA, Legal Affairs), Santiago Elena (CSIC), Paulien Hogeweg (Utrecht Univ.), Carole Knibbe (INRIA), Otmane Lamrabet (UJF), Caroline Lothe (INRIA, Team assistant), Aurelia Mouton (INRIA, Human Resources Management), Christophe Rigotti (INRIA), Charles Rocabert (INRIA), Fanny Rossetti (INRIA, Financial affairs), Dominique Schneider (UJF), and Susan Stepney (Univ. of York)
- **General meeting (Utrecht, May 21-23 2014).** Attendees: Paul Andrews (Univ. of York), Guillaume Beslon (INRIA), José-Luis Carrasco (CSIC), Thomas Cuypers (Utrecht Univ.), Enrico Sandro Colizzi (Utrecht Univ.), Bram van Dijk (Utrecht Univ.), Santiago Elena (CSIC), Paulien Hogeweg (Utrecht Univ.), Carole Knibbe (INRIA), Otmane Lamrabet (UJF), Sergio Peignier (INRIA), Charles Rocabert (INRIA), Jaap Rutten (Utrecht Univ.), Dominique Schneider (UJF), Gÿs Schroder (Utrecht Univ.), Susan Stepney (Univ. of York), Yoram Vadée-le-Brun (INRIA), and Anouk Willemsen (CSIC). Note that this meeting has been preceded by a long stay of Charles Rocabert (Partner 1) in Utrecht University to discuss the design of the computational model (WP2).
- **General meeting (Lyon, October 27-29 2014).** Attendees: Paul Andrews (Univ. of York), Guillaume Beslon (INRIA), Thomas Cuypers (Utrecht Univ.), Bram van Dijk (Utrecht Univ.), Santiago Elena (CSIC), Paulien Hogeweg (Utrecht Univ.), Carole Knibbe (INRIA), Otmane Lamrabet (UJF), Vincent Liard (INRIA), David Parsons (INRIA), Sergio Peignier (INRIA), Christophe Rigotti (INRIA), Charles Rocabert (INRIA), Jonathan Rouzaud-Cornabas (INRIA), Susan Stepney (Univ. of York), and Yoram Vadée-le-Brun (INRIA).
- **General meeting (York, July 25-26 2015).** This meeting immediately followed the first EvoEvo workshop. Attendees: Guillaume Beslon (INRIA), Thomas Cuypers (Utrecht Univ.), Bram van Dijk (Utrecht Univ.), Santiago Elena (CSIC), Simon Hickinbotham (Univ. of York), Paulien Hogeweg (Utrecht Univ.), Tim Hoverd (Univ. of York), David Parsons (INRIA), Sergio Peignier (INRIA), Charles Rocabert (INRIA), Susan Stepney (Univ. of York), and Yoram Vadée-le-Brun (INRIA).
- **General meeting (Valencia, February 22-24 2016).** Attendees: Jonas Abernot (INRIA), Guillaume Beslon (INRIA), José-Luis Carrasco (CSIC), Thomas Cuypers (Utrecht Univ.), Bram van Dijk (Utrecht Univ.), Santiago Elena (CSIC), Simon Hickinbotham (Univ. of York), Paulien Hogeweg (Utrecht Univ.), Tim Hoverd (Univ. of York), Otmane Lamrabet (UJF), Sergio Peignier (INRIA), Charles Rocabert (INRIA), Susan Stepney (Univ. of York), and Anouk Willemsen (CSIC).

- **General meeting (Utrecht, June 1-3 2016).** Attendees: Jonas Abernot (INRIA), Guillaume Beslon (INRIA), Sandro Colizzi (Utrecht Univ.), Thomas Cuypers (Utrecht Univ.), Bram van Dijk (Utrecht Univ.), Simon Hickinbotham (Univ. of York), Thomas Hindré (UJF), Paulien Hogeweg (Utrecht Univ.), Tim Hoverd (Univ. of York), Otmane Lamrabet (UJF), Sergio Peignier (INRIA), Charles Rocabert (INRIA), and Susan Stepney (Univ. of York).
- **Final meeting (Amsterdam, September 21-22 2016).** The objectives of this meeting were to prepare the end of the project and to structure the present document. Partners 2 (UJF) and 5 (CSIC) were not physically present but contributed to the decisions by mail. The meeting immediately follows the second EvoEvo workshop. Attendees: Jonas Abernot (INRIA), Guillaume Beslon (INRIA), Sandro Colizzi (Utrecht Univ.), Thomas Cuypers (Utrecht Univ.), Bram van Dijk (Utrecht Univ.), Simon Hickinbotham (Univ. of York), Paulien Hogeweg (Utrecht Univ.), Tim Hoverd (Univ. of York), Sergio Peignier (INRIA), Charles Rocabert (INRIA), and Susan Stepney (Univ. of York).

All partners were represented at all the general meetings (except the third general meeting in York where partner 2 was unable to attend because of health concerns) and the people working full-time on the project attended most general meetings. Moreover, during the kick-off meeting in Lyon, members of INRIA financial affairs, legal affairs and human resources where attending to present the administrative rules of the project (see deliverable D6.3 for details). For all meetings, a meeting secretary was identified. The internal project communication was based on standard electronic tools: a mail diffusion list (evoevo@insa-lyon.fr), Skype, a Document Management System available on the project intranet, and development repositories (gforge.inria.fr and GitHub) used for software development. All these tools were available from November 2013 (*i.e.*, during the first month of the project).

The general project meetings were not the only project meetings held during the project. Many technical meetings were also held (generally between two partners, sometimes between three partners). Moreover, these technical meetings were often longer than the general project meetings in order to allow deeper technical/scientific discussions. The list of these technical meetings is too long to be presented here and we only present them globally:

- Several technical meetings between partners 1 (INRIA) and 3 (UU) were held at the beginning of the project, to discuss the specifications of the integrated model (WP2).
- A long visit of partner 3 (UU) and partner 1 (INRIA) to partner 2 (UGA) was held to discuss the modelling of the LTEE experiment in WP3. This long visit was followed by many shorter visits to discuss the implementation details.
- Two technical meetings between partner 1 (INRIA) and partner 5 (CSIC) were held to discuss the modelling of virus evolution with the Aevol model.
- Four technical meetings were held between partner 4 (UoY) and partner 1 (Inria) to discuss the concept of open-ended evolution.
- Partner 4 (UoY) visited partner 1 (INRIA) to discuss the implementation of the EvoMachina software.
- Partner 1 (INRIA) visited partner 4 (UoY) to coordinate the evolutionary music and dance demonstrations given at the Huddersfield conference.

Although the global workload of the project meetings cannot be precisely computed, we estimate that the general and technical meetings globally represent approximately 15 person-months of the

total workload of the EvoEvo project (*i.e.*, approximately 5% of the total workload). This clearly shows that the project management, monitoring and progress strongly relied on direct partner-to-partner communications (this estimate does not take into account the many meetings organized locally by each partner).

Globally we had no management difficulties during the project. All partners were very reactive to requests, and most deliverables and milestones were produced and met in time. Where deliverables or milestones were delayed, it was for technical and scientific reasons that were always discussed previously within the consortium. Given the size of the project and the number of collaborators, we had few management difficulties on that aspect. In particular, we are proud to announce that four collaborators of the project (Kirsten ten Tusscher, Carole Knibbe, Jonathan Rouzaud-Cornabas, and Thomas Cuypers) each had a baby during the EvoEvo project. To our eyes this shows that gender issues were correctly managed. Most partners hired all necessary personnel (PhD students, post-docs, engineers, and lab technicians) from the beginning of the project, the only exception being Utrecht University who experienced difficulties in hiring a postdoc. This resulted in a lower than expected workload in Utrecht during the first year of the project. However, these difficulties were overcome when Utrecht University hired a post-doc (Thomas Cuypers) and a PhD student (Bram van Dijk) in January 2015, thus correcting this effect. Finally, partner 4 (University of York) had management difficulty with two post-docs who successively found positions in private companies during their post-doc. This slightly delayed the development of EvoMachina, and we adapted the interactions between WP4 and WP5 accordingly. The main consequence was that the evolutionary clustering algorithms developed in WP5 was first developed in C++ before being implemented and tested in EvoMachina.

As explained in the DoW, the project work plan was organized in such a way that the workpackages dependencies were mainly based on knowledge exchanges (biological knowledge between WP1, WP2, and WP4; experimental design and results between WP1 and WP3; specifications of the computational model between WP2, WP3 and WP4; EvoEvo strategies between WP3 and WP5). These exchanges and represented as horizontal arrows on **Figure 1** (page 6). This was a guarantee of efficient project progress (providing that partners interact regularly to exchange the corresponding knowledge – which was ensured by the regular project meetings). The two exceptions were the dependencies between WP2 and 3 (model development and usage) and WP4 and 5 (computational framework development and usage) – vertical arrows on **Figure 1**. Interaction between WP2 and WP3 was relatively straightforward since partners 1 and 3 were both strongly involved in these two WPs. Moreover, as explained in the DoW, WP3 used the model developed in WP2 but also various models developed previously by partners 1 and 3 – the model used for each *in silico* experiment being chosen in close collaboration with WP1. This enabled us to design many more experiments than what would have been possible with the sole Evo$^2$Sim model (*e.g.*, *in silico* TEV-like experiments were conducted with the Aevol model since the fine genome structure has been shown to be a critical features during the wet experiments). As explained previously, due to human-resource management issues in UoY, the interactions between WP4 and WP5 had to be reorganized. Although the development of EvoMachina was delayed, we chose not to delay the development of the WP5 applications (this would have severely compromised the outcome of WP5) but to develop a prototype of WP5 applications independently from the EvoMachina package, and then to test the same algorithm within EvoMachina. This was done: ChameleoClust was developed and tested in C++

simultaneously with the development of EvoMachina, and the undelying Chameleoclust algorithm was then implemented successfully within this framework.

**List of project deliverables:**

All the project deliverables have been produced and all public deliverables are available on the project website (http://www.evoevo.eu/deliverables/). Most deliverables were produced on time; a few were delayed to follow the scientific development of the project.

| | | | |
|---|---|---|---|
| D1.1 | TEV and *E. coli* strains for robustness analysis | Published online on November 27th 2014 | http://www.evoevo.eu/deliverable-1-1-tev-and-e-coli-strains-for-robustness-analysis/ |
| D1.2 | Analysis of robustness in TEV and *E. coli* | Published online on March 25th 2016 | http://evoevo.liris.cnrs.fr/publication-of-the-deliverable-1-2-analysis-of-robustness-in-tev-and-e-coli-strains/ |
| D1.3 | Analysis of evolvability (part 1) | Published online on March 25th 2016 | http://evoevo.liris.cnrs.fr/publication-of-the-deliverable-1-3-analysis-of-evolvability-part-1/ |
| D1.4 | Analysis of phenotypic innovation (part 1) | Published online on March 25th 2016 | http://evoevo.liris.cnrs.fr/publication-of-deliverable-1-4-analysis-of-phenotypic-innovation-part-1/ |
| D1.5 | Analysis of evolvability (part 2) | Published online on October 11th 2016 | http://evoevo.liris.cnrs.fr/publication-of-deliverable-1-5-analysis-of-evolvability-part-2/ |
| D1.6 | Analysis of phenotypic innovation (part 2) | Published online on October 11th 2016 | http://evoevo.liris.cnrs.fr/publication-of-deliverable-1-6-analysis-of-phenotypic-innovation-part-2/ |
| D2.1 | Specifications of the genome-network model | Published online on July 4th 2014 | http://evoevo.liris.cnrs.fr/deliverable-2-1-specifications-of-the-genome-network-model/ |
| D2.2 | Genome-network model | Published online on December 10th 2014 | http://evoevo.liris.cnrs.fr/deliverable-2-2-genome-network-model-source-code/ |
| D2.3 | Specifications of the population model | Published online on July 24th 2014 | http://evoevo.liris.cnrs.fr/deliverable-2-3-specifications-of-the-population-model/ |

| | | | |
|---|---|---|---|
| D2.4 | Population model | Published online on May 23th 2015 | http://evoevo.liris.cnrs.fr/publication-of-the-deliverable-2-4-population-model-source-code/ |
| D2.5 | Specifications of the realistic network model | Published online on November 26th 2014 | http://evoevo.liris.cnrs.fr/deliverable-2-5-specifications-of-the-realistic-network-model/ |
| D2.6 | Realistic-network model | Published online on May 23th 2015 | http://evoevo.liris.cnrs.fr/publication-of-the-deliverable-2-6-realistic-network-model-source-code/ |
| D2.7 | Description of the modeling choices for the integrated model | Published online on February 19th 2015 | http://evoevo.liris.cnrs.fr/deliverable-2-7-specifications-of-the-integrated-evolutionary-model/ |
| D2.8 | Integrated evolutionary model | Published online on May 23th 2015 | http://evoevo.liris.cnrs.fr/publication-of-the-deliverable-2-8-integrated-evolutionary-model-source-code/ |
| D3.1 | Evolution of variability; Mechanisms and consequences | Published online on November 3rd 2015 | http://evoevo.liris.cnrs.fr/publication-of-the-deliverable-3-1-evolution-of-variability-mechanisms-and-consequences/ |
| D3.2 | Evolution of robustness; Mechanisms and consequences | Published online on October 8th 2016 | http://evoevo.liris.cnrs.fr/publication-of-deliverable-d3-2-evolution-of-robustness/ |
| D3.3 | Evolution of evolvability; Mechanisms and consequences | Published online on October 9th 2016 | http://evoevo.liris.cnrs.fr/publication-of-deliverable-d3-3-evolution-of-evolvability/ |
| D3.4 | Evolution of open-endedness; Mechanisms and consequences | Published online on October 28th 2016 | http://evoevo.liris.cnrs.fr/publication-of-deliverable-d3-4-evolution-of-open-endedness-mechanisms-and-consequences/ |
| D4.1 | Computational meta-model definition | Published online on May 11th 2016 | http://evoevo.liris.cnrs.fr/publication-of-the-deliverable-4-1-computational-meta-model-definition/ |

| | | | |
|---|---|---|---|
| D4.2 | Computational model requirements specification | Published online on May 11th 2016 | http://evoevo.liris.cnrs.fr/publication-of-the-deliverable-4-2-computational-model-requirements-specification/ |
| D4.3 | A calibrated, tested and documented implementation of the platform specification | Published online on September 15th 2016 | http://evoevo.liris.cnrs.fr/publication-of-deliverable-4-3-computational-run-time-platform/ |
| D4.4 | Computational reflective run-time platform | Published online on November 7th 2016 | http://evoevo.liris.cnrs.fr/publication-of-the-deliverable-4-4-computational-reflective-run-time-platform/ |
| D4.5 | Reflective application | Published online on November 7th 2016 | http://evoevo.liris.cnrs.fr/publication-of-the-deliverable-4-5-reflective-applications/ |
| D5.1 | Impact obtained from EvoEvo mechanisms on data stream cluster analysis | Published online on October 26th 2016 | http://evoevo.liris.cnrs.fr/publication-of-the-deliverable-5-1-impact-obtained-from-evoevo-mechanisms-on-data-stream-cluster-analysis/ |
| D5.2 | Impact obtained from EvoEvo mechanisms on evolution of a hardware personal companion | Published online on October 31th 2016 | http://evoevo.liris.cnrs.fr/publication-of-the-deliverable-5-1-impact-obtained-from-evoevo-mechanisms-on-evolution-of-a-hardware-personal-companion/ |
| D6.1 | Project website | Published online on July 8th 2014 | http://evoevo.liris.cnrs.fr/deliverable-6-1-project-website/ |
| D6.2 | Project communication media | Published online on July 8th 2014 | http://evoevo.liris.cnrs.fr/deliverable-6-2-project-communication-media/ |
| D6.5 | Mid-term dissemination report | Published online on May 23th 2015 | http://evoevo.liris.cnrs.fr/publication-of-deliverable-6-5-mid-term-dissemination-report/ |
| D6.6 | Program of the interdisciplinary workshop | Published online on September 7th 2016 | http://evoevo.liris.cnrs.fr/publication-of-deliverable-6-6-program-of-the-interdisciplinary-workshop/ |

### 2.7.3. Contribution 2: Dissemination

**Introduction**

EvoEvo is a concept and a term that was coined in a joint publication by partner 1 and 2 (Hindré *et al.*, 2012). One of the objectives of the project was to develop the concept, to verify its scientific soundness and, ultimately, to disseminate it in at least two scientific communities: evolutionary biology and computer science (evolutionary computation and artificial life). The dissemination strategy is presented in deliverable D6.5, and includes several dissemination actions. As explained in that deliverable, the dissemination actions can be split in two parts: dissemination of the results and dissemination of the concepts. Dissemination of the results is mainly a question of scientific publications. On that aspect, EvoEvo is a clear success, with 55 international publications in all the domains covered by the project, and many further in preparation. The dissemination of the concepts is less easy to measure and takes more time. As proposed in the project, our approach toward living technologies will be disseminated through videos of our demonstrators and availability of software. Videos of EvoMove sessions have been published, and the EvoWave software is available on the project website (see sections 2.6.3 and 2.6.4). The EvoMachina software is available on github (see section 2.5.3). Finally, to discuss and disseminate the EvoEvo concept, we organized two interdisciplinary workshops. Note that the EvoEvo concept proved to be more difficult to disseminate than initially expected. This was mainly due to the difficulty of distinguishing this concept from that of the concept of Evolvability that has become highly popular these last years. It was also due to those search engines that transform “EvoEvo” into “Evo Evo”, thus mixing the concept with a welter of irrelevant information.

**Awards and distinctions:**

Sergio Peignier, Christophe Rigotti, Guillaume Beslon (2015) Subspace clustering using evolvable genome structure. *In: proceedings of GECCO'15, Annual Conference on Genetic and Evolutionary Computation*. Madrid (Spain), July 2015, pp. 575-582 **[Best paper award]**

Jessica Plucain, Antonia Suau, Stéphane Cruveiller, Claudine Médigue, Dominique Schneider, Michael Le Gac (2016) Contrasting effects of historical contingency on phenotypic and genomic trajectories during a two-step evolution experiment with bacteria. *BMC Evolutionary Biology*, 16(1):86 **[Recommended by F1000]**

Bram van Dijk (2015) Evolution of differential gene mobility. BioSB 2015 Dutch Bioinformatics & Systems Biology conference **[Best presentation award]**

Simon Hickinbotham, Susan Stepney (2016) Augmenting Live Coding with Evolved Patterns. In: *Proceedings of EvoMusArt, 5th International Conference on Computational Intelligence in Music, Sound, Art and Design,* Porto, Portugal, March 2016, LNCS 9596:31-46 **[best paper award nominee]**

Tim Hoverd, Susan Stepney. (2016) EvoMachina: a novel evolutionary algorithm inspired by bacterial genome reorganisation (abstract). *Late Breaking Abstracts, UCNC 2016, Manchester*, UK, 2pp., **[highly commended poster award]**

**Publications of the project:**

1. Abernot J., Beslon G., Hickinbotham S., Peigner S., Rigotti C. (2016) A Commensal Architecture for Evolving Living Instruments. In: *proceedings of Conference on Computer Simulation of Musical Creativity*, Huddersfield, United Kingdom, June 2016, 8pp.

2. Andrews P. S., Stepney S. (2014) Using CoSMoS to Reverse Engineer a Domain Model for Aevol. In: *Proceedings of CoSMoS workshop*, New York, USA, July 2014, pp.61-79. Luniver Press

3. Andrews P. S., Stepney S. (2015) A Metamodel for the Evolution of Evolution. *In proceedings of ECAL 2015, York, UK,* July 2015, pp.621-628. MIT Press

4. Andrews P. S., Stepney S. (2015) The CoSMoS Domain Experiment Model. *CoSMoS workshop, York, UK,* July 2015 pp.1–8. Luniver Press

5. Banzhaf W., Baumgaertner B., Beslon G., Doursat R., Foster J. A., McMullin B., Veloso de Melo V., Miconi T., Spector L., Stepney S., White R. (2016) Defining and Simulating Open-Ended Novelty: Requirements, Guidelines, and Challenges. *Theory in Biosciences*, 135(3):131-161

6. Banzhaf W., Beslon G., Doursat R., Stepney S. (2016) Open-Endedness: Definitions and Shortcuts. *The Second Workshop on Open-Ended Evolution, at ALife 2016*, Cancun, Mexico, 2pp.

7. Batut B., Beslon G., Knibbe C. (2016). Unexpected genome inflation and streamlining in variable environments. *Journées ouvertes de Biologie Informatique & Mathématiques 2016*, June 2016, Lyon, France

8. Bernet G. P., Elena S. F. (2015) Distribution of mutational fitness effects and of epistasis in the 5'untranslated region of a plant RNA virus. *BMC Evolutionary Biology*, *BMC Evol. Biol*. 15(274)

9. Beslon G., Liard V., Elena S. F. (2016) Evolvability drives innovation in viral genomes. *2nd EvoEvo Workshop, satellite workshop of CCS2016*, Sep 2016, Amsterdam, NL, 6 p.

10. Biller P., L Guéguen, Knibbe C., Tannier E. (2016) Breaking good: accounting for fragility of genomic regions in rearrangement distance estimation. *Genome Biology and Evolution*, 8(5):1427-1439

11. Biller P., Knibbe C., Beslon G., Tannier E. (2016) Comparative genomics on artificial life. In: proceedings of CiE 2016 (Computability in Europe), Paris, France, Volume 9709 of the series Lecture Notes in Computer Science, pp 35-44

12. Biller P., Tannier E., Beslon G., Knibbe C. (2016) *In silico* experimental evolution provides independent and challenging benchmarks for comparative genomics. *Journées ouvertes de Biologie Informatique & Mathématiques 2016*, June 2016, Lyon, France, pp. 79-82

13. Cervera H., Lalić J., Elena S. F. (2016) Efficient escape from local optima in a highly rugged fitness landscape by evolving RNA virus populations. *Proceedings of the Royal Society, Biological Sciences*, 283: 20160984.

14. Cervera H., Elena S. F. (2016) Genetic variation in fitness within a clonal population of a plant RNA virus. *Virus Evolution*, 2(1): vew006.

15. Cervera H., Lalić J., Elena S. F. (2016) Effect of host species on topography of the fitness landscape for a plant RNA virus. *Journal of Virology,* 90(22):10160-10169

16. Colizzi E. S., Hogeweg P. (2016) Mutational load is ameliorated by increased transcriptional load-associated mutations, if these are biased towards duplications and deletions. *2nd EvoEvo Workshop, satellite workshop of CCS2016*, Sep 2016, Amsterdam, NL, 5 p.

17. Cuevas J. M., Willemsen A., Hillung J., Zwart M. P., Elena S. F. (2015) Temporal Dynamics of Intrahost Molecular Evolution for a Plant RNA Virus. *Molecular Biology and Evolution*, 32(5):1132-1147

18. Cuypers T. D., Hogeweg P. (2015) Endless evolutionary paths to Virtual Microbes.Workshop, *First EvoEvo Workshop, satellite workshop of ECAL2015*, July 2015, York, UK, 1 p.

19. Elena S. F. (2016) Evolutionary transitions during RNA virus experimental evolution. *Philosophical Transactions of the Royal Society, Biological Sciences*, 371(1701):20150441

20. Elena S. F. (2016) Local adaptation of plant viruses: lessons from experimental evolution. *Molecular Ecology*, [Epub ahead of print]. PMID: 27612225.

21. Fischer S., Bernard S., Beslon G., Knibbe C. (2014) A model for genome size evolution. *Bulletin of Mathematical Biology*, 76(9):2249-2291

22. Großkopf T., J Consuegra, J Gaffé, JC Willison, RE Lenski, OS Soyer, Schneider D. (2016) Metabolic modelling in a dynamic evolutionary framework predicts adaptive diversification of bacteria in a long-term evolution experiment. *BMC Evolutionary Biology*, 16(1):163

23. Hickinbotham S., Stepney S. (2015) Conservation of matter increases evolutionary activity. *In proceedings of ECAL 2015,* York, UK, July 2015, pp. 98-105

24. Hickinbotham S., Stepney S. (2015) Environmental bias forces parasitism in Tierra. *In proceedings of ECAL 2015*, York, UK, July 2015, pp. 294-301

25. Hickinbotham S., Stepney S. (2016) Bio-Reflective Architectures for Evolutionary Innovation. *In proceedings of ALife 2016*, Cancun, Mexico, July 2016, pp. 192-199

26. Hickinbotham S., Stepney S. (2016) Augmenting Live Coding with Evolved Patterns. In: *Proceedings of EvoMusArt, 5th International Conference on Computational Intelligence in Music, Sound, Art and Design,* Porto, Portugal, March 2016, LNCS 9596:31-46

27. Hickinbotham S., Hogeweg P. (2016) Evolution towards extinction in replicase models: inevitable unless... *2nd EvoEvo Workshop, satellite workshop of CCS2016*, Sep 2016, Amsterdam, NL, 5 p.

28. Hillung J., Cuevas J. M., Elena S. F. (2015) Evaluating the within-host fitess effects of mutations fixed during virus adaptatyion to different ecotypes of a new host. *Philosophical Transactions of the Royal Society, Biological Sciences*, 370(1675):20140292

29. Hillung J., García-García F., Dopazo J., Cuevas J. M., Elena S. F. (2016) The transcriptomics of an experimentally evolved plant-virus interaction. *Scientific Reports,* 6:24901

30. Hoverd T., Stepney S. (2016) EvoMachina: a novel evolutionary algorithm inspired by bacterial genome reorganisation. *Late Breaking Abstracts, UCNC 2016*, *Manchester, UK*, 2pp., 2016

31. Hoverd T., Stepney S. (2016) EvoMachina: a novel evolutionary algorithm inspired by bacterial genome reorganisation. *2nd EvoEvo workshop*, Amsterdam, Netherlands.

32. Knibbe C., Parsons D. (2014) What happened to my genes? Insights on gene family dynamics from digital genetics experiments. *In proceedings of ALIFE 14, 14th Intl. Conf. on the Synthesis and Simulation of Living Systems*. New-York (US), July 2014, pp. 33-40

33. Lalić J., Elena S. F. (2015) The impact of high-order epistasis in the within-host fitness of a positive-sense plant RNA virus. *Journal of Evolutionary Biology* 28(12):2236-2247

34. Peigner S., Rigotti C., Beslon G. (2015) Subspace Clustering for all Seasons. *First EvoEvo Workshop, satellite workshop of ECAL2015*, July 2015, York, UK, 1 p.

35. Peignier S., Rigotti C., Beslon G. (2015) Subspace clustering using evolvable genome structure. *In: proceedings of GECCO'15, Annual Conference on Genetic and Evolutionary Computation*. Madrid (Spain), July 2015, pp. 575-582

36. Plucain J., Suau A., Cruveiller S., Médigue C., Schneider D., Le Gac M. (2016) Contrasting effects of historical contingency on phenotypic and genomic trajectories during a two-step evolution experiment with bacteria. *BMC Evolutionary Biology*, 16(1):86

37. Raeside C., Gaffé J., Deatherage D. E., Tenaillon O., Briska A. M., Ptashkin R. N., Cruveiller S., Médigue C., Lenski R. E., Barrick J. E., Schneider D. (2014) Large chromosomal rearrangements during a long-term evolution experiment with *Escherichia coli*. *mBio* 5:e01377-14

38. Rocabert C., Knibbe C., Beslon G. (2015) Towards an Integrated Evolutionary Model to Study Evolution of Evolution. *First EvoEvo Workshop, satellite workshop of ECAL2015*, July 2015, York, UK, 15 p.

39. Rocabert C., Knibbe C., Consuegra J., Schneider D., Beslon G. (2016) Environmental Driving of Bacterial Diversification in *In Silico* Experimental Evolution. *Evolutionary systems biology: from model organisms to human disease workshop,* Cambridge (UK), March 2016, 1 p.

40. Rocabert C., Knibbe C., Consuegra J., Schneider D., Beslon G. (2016) In Silico Experimental Evolution Highlights the Influence of Environmental Seasonality on Bacterial Diversification. *2nd EvoEvo Workshop, satellite workshop of CCS2016*, Sep 2016, Amsterdam, NL, 4 p.

41. Rudan M., Schneider D., Warnecke T., Krisko A. (2015) RNA chaperones buffer deleterious mutations in *E. coli*. *eLife*, 4:e04745

42. Rutten J., Hogeweg P., Beslon G. (2016) Evolution of mutator populations in constant environments. *2nd EvoEvo Workshop, satellite workshop of CCS2016*, Sep 2016, Amsterdam, NL, 5pp.

43. Stepney S., Beslon G. (2016) Open-Endedness: Definitions and Shortcuts. *2nd EvoEvo Workshop, satellite workshop of CCS2016*, Sep 2016, Amsterdam, NL, 6pp.

44. Takeuchi N., Kaneko K., Hogeweg P. (2016) Evolutionarily stable disequilibrium: endless dynamics of evolution in a stationary population. *Proceedings of the Royal Society*, Biological Sciences, 283(1830):20153109

45. Taylor T., Bedau M., Channon A., Ackley D., Banzhaf W., Beslon G., Dolson E., Froese T., Hickinbotham S., Ikegami T., McMullin B., Packard N., Rasmussen S., Virgo N., Agmon E,. Clark E., McGregor S., Ofria C., Ropella G., Spector L., Stanley K. O., Stanton A., Timperley C., Vostinar A., Wiser M. (2016) Open-Ended Evolution: Perspectives from the OEE1 Workshop in York. *Artificial Life*, 22(3):408-423

46. Taylor T., Auerbach J. E., Bongard J., Clune J., Hickinbotham S., Ofria C., Oka M., Risi S., Stanley K. O., Yosinski J. (2016) WebAL Comes of Age: A review of the first 21 years of Artificial Life on the Web. *Artificial Life*, 22(3):364-407

47. Tenaillon O., Barrick J. E., Ribeck N., Deatherage D. E., Blanchard J. L., Dasgupta A., Wu G. C., Wielgoss S., Cruveiller S., Médigue C., Schneider D., Lenski R. E. (2016) Tempo and mode of genome evolution in a 50,000-generation experiment. *Nature*, 536(7615):165-170

48. Tromas N., Zwart M. P., Forment J., Elena S. F. (2014) Shrinkage of genome size in a plant RNA virus upon transfer of an essential viral gene into the host genome. *Genome Biology and Evolution*, 6(3):538-550

49. Vadée-Le-Brun Y., Rouzaud-Cornabas J., Beslon G. (2015) Epigenetic inheritance speeds up evolution of artificial organisms. *In proceedings of ECAL 2015*, York, UK, July 2015, pp. 439-446

50. Vadée-Le-Brun Y., Rouzaud-Cornabas J., Beslon G. (2015) In Silico Experimental Evolution suggests a complex intertwining of selection, robustness and drift in the evolution of genetic networks complexity. *In proceedings of ALife 2016*, Cancun, Mexico, July 2016, pp. 172-179

51. van Dijk B., Hogeweg P. (2016) In silico gene-level evolution explains microbial population diversity through differential gene mobility. *Genome Biology and Evolution*, 28;8(1):176-188

52. van Dijk B., Cuypers T. D., Hogeweg P. (2016) Evolution of r- and K-selected species of Virtual Microbes: a case study in a simple fluctuating 2-resource environment . 2nd EvoEvo Workshop, satellite workshop of CCS2016, Sep 2016, Amsterdam, NL, 5 p.

53. Willemsen A., Zwart M. P., Elena S. F. (2016) High virulence does not necessarily impede viral adaptation to a new host: A case study using a plant RNA virus. *bioRxiv* 060137; doi: http://dx.doi.org/10.1101/060137

54. Willemsen A., Zwart M. P., Higueras P., Sardanyés J., Elena S. F. (2016) Predicting the stability of homologous gene duplications in a plant RNA virus. *Genome Biology and Evolution*, 8(9):3065-3082.

55. Willemsen A., Zwart M. P., Ambrós S., Carrasco J. L., Elena S. F. (2016) 2b or not 2b: Experimental evolution of functional exogenous sequences in a plant RNA virus. *bioRxiv* 079970; doi: http://dx.doi.org/10.1101/079970

56. Willemsen A., Zwart M. P., Tromas N., Majer E., Daròs J. A., Elena S. F. (2016) Multiple barriers to the evolution of alternative gene orders in a positive-strand RNA virus. *Genetics*, 2(4):1503-1521

57. Wu B., Zwart M. P., Sanchez-Navarro J. A., Elena S. F. (2016) Within-host evolution of segments ratio for the tripartite genome of *Alfalfa mosaic virus*. *bioRxiv* 066084; doi: http://dx.doi.org/10.1101/066084

**Publications in preparation (non-exhaustive list):**

Batut B., Beslon G., Knibbe.: Genome inflation and streamlining in variable environments.

Beslon G., Liard V., Elena S. F.: Evolvability drives innovation in viral genomes

Clark E. B., Hickinbotham S., Stepney S.: Semantic closure demonstrated by the evolution of universal constructor in stringmol

Consuegra J., Plucain J., Gaffé J., Lenski R. E., Hindré T., Schneider D.: Molecular genetics of a new ecological opportunity exploitation during long-term bacterial sympatric adaptive diversification

Cuypers T. D., Rutten J. P., Hogeweg P.: Mutate or Regulate: evolutionary strategies along a continuum of ecological time scales

Hickinbotham S., Hogeweg P.: Evolution towards extinction in replicase models: inevitable unless...

Hoverd T., Stepney S.: EvoMachina: a novel evolutionary algorithm based on a machine meta-model

Lamrabet O., Plumbridge J., Lenski R. E., Hindré T., Schneider D.: Dynamics of altered regulatory networks in bacteria

Knibbe C, Schneider D., Beslon G.: Evolution without (point) mutations

Meijer J., van Dijk B., Cuypers T., Hogeweg P.: The role of HGT in the evolution of genomes, transcriptomes, metabolomes and ecosystems

Peignier S., Rigotti C., Beslon G.: Weight-based search to find clusters around medians in subspaces (submitted to SIAM DM)

Rocabert C., Knibbe C., Consuegra J., Schneider D., Beslon G.: Beware Batch Culture: Seasonality and Niche Construction Predicted to Favor Bacterial Adaptive Diversification (submitted to PLoS Computational Biology)

Rocabert C., Bernard S., Knibbe C., Beslon G.: Applying phenotypic noise to the Fisher's geometric model

Rutten J. P., Hogeweg P., Beslon G.: Adapting the engine to the fuel: hypermutator populations can escape the mutational burden by reorganizing their genome structure

van Dijk B., Cuypers T. D., Hogeweg P.: Evolution of the stringent response in virtual microbes

Wielgoss S., Hindré T., Lenski R. E., Schneider D.: Long-term evolution of global gene expression through pervasive changes in epistatic interactions within regulatory networks.

### 2.7.4. Organization of two interdisciplinary workshops in 2015 and 2016

One of the central claims of the project is that EvoEvo is a multidisciplinary concept that could interest both biology (evolutionary biology, microbiology, virology…) and also computer science. In the EvoEvo project we aimed at proposing new evolutionary algorithms that integrate evoevo properties. This interdisciplinary nature of the concept makes it difficult to disseminate widely, which is why we proposed to organize two interdisciplinary workshops during the project. The aim of the EvoEvo workshops was to seek for a unified theory of Evolution of Evolution by studying its biological mechanisms, evolutionary consequences and possible applications to bioinspired computation. In order to facilitate interdisciplinary dissemination, we chose to organize both workshops as satellite events of highly interdisciplinary conferences, rather that as independent events.

**First EvoEvo workshop (York, July 2015)**

The first EvoEvo workshop was held on 24 July 2015 as a satellite workshop of the European Conference on Artificial Life (ECAL 2015) in York UK. We welcomed approximately 35 attendees from various fields. The program was composed of invited speakers, long talks (from long communications) and short talk (from extended abstracts). The invited speakers were Lee Altenberg (Vienna University, Austria) who gave a talk on the origin of evolvability[3] and Jean-Baptiste Mouret (Nancy University, France). The contributions were given by Anton Crombach (Barcelona, Spain), Charles Rocabert (INRIA, France), Ben Kovitz (Indiana University, Bloomington, USA), Yifei Wang (University of Bath, UK), Thomas Cuypers (Utrecht University, NL), Tim Taylor (University of York, UK), and Sergio Peignier (INRIA, France). More information and material is available from the workshop website (http://evoevo.liris.cnrs.fr/evoevo-2015/).

The workshop was a real success with very interesting discussions and initiation of possible collaborations. As one of the participants noted, the workshop was the occasion to discuss concepts (*e.g.*, “cascading design”) that could not be discussed elsewhere or not in the same way.

**Second EvoEvo workshop (Amsterdam, September 2016)**

The second EvoEvo workshop was held on 20 September 2016 as a satellite workshop of the Conference on Complex Systems (CCS 2016) in Amsterdam. Compared to the first EvoEvo workshop, which aimed at discussing the EvoEvo concept with the community, this second EvoEvo workshop mainly aimed at presenting the results of the EvoEvo project to the scientific community. The workshop was organized a month before the end of the project, and the CCS conference offered us the opportunity to disseminate our results to a large interdisciplinary audience including computer scientists, physicists, mathematicians and biologists. Consequently, the workshop

[3] Note that Lee Altenberg is one of authors who popularized the concept of evolvability.

program was mainly composed of invited speakers and EvoEvo speakers presenting the interdisciplinary results of the project.

The three invited speakers were:

- Eörs Szathmary[4] (Parmenides Center for the Conceptual Foundations of Science, Pullach/Munich, Germany): *Evolution of Evolvable Systems*
- François Blanquart (Imperial College): *Epistasis and the structure of fitness landscapes: are experimental fitness landscapes compatible with Fisher's geometric model?*
- Nobuto Takeuchi (University of Tokyo): *The origin of genes through spontaneous symmetry breaking*

Paulien Hogeweg (Utrecht University) gave an introduction talk to present the concept of "Evolution of Evolution" ("What's EvoEvo?") and the results of the EvoEvo projects were presented in 9 contributions (speaker underlined):

1. Guillaume Beslon (INRIA), Vincent Liard (INRIA), Santiago Elena (CSIC): Evolvability drives innovation in viral genomes
2. Enrico Sandro Colizzi (Utrecht University), Paulien Hogeweg (Utrecht University): Mutational load is ameliorated by increased transcriptional load-associated mutations, if these are biased towards duplications and deletions
3. Jacob Pieter Rutten (INRIA), Paulien Hogeweg (Utrecht University), Guillaume Beslon (INARI): Evolution of mutator populations in constant environments
4. Susan Stepney (University of York), Guillaume Beslon (INRIA): Open-Endedness: Definitions and Shortcuts
5. Charles Rocabert (INRIA), Carole Knibbe (INRIA), Jessica Consuegra (UGA), Dominique Schneider (UGA), Guillaume Beslon (INRIA): In-silico experimental evolution highlights the influence of environmental seasonality on bacterial diversification
6. Bram van Dijk (Utrecht University), Thomas Cuypers (Utrecht University), Paulien Hogeweg (Utrecht University): Evolution of r- and K-selected species of Virtual Microbes: a case study in a simple fluctuating 2-resource environment
7. Tim Hoverd (University of York), Susan Stepney (University of York): EvoMachina: a novel evolutionary algorithm inspired by bacterial genome reorganisation
8. Simon Hickinbotham (University of York), Paulien Hogeweg (Utrecht University): Evolution towards extinction in replicase models: inevitable unless…
9. Jonas Abernot (INRIA), Simon Hickinbotham (University of York): Physical interaction with automated music composition platforms

The extending abstracts of these contributions are all available on the workshop website (http://evoevo.liris.cnrs.fr/evoevo-2016/). Of these 9 contributions, 6 were direct collaborations between two partners of the project (see Figure 12). The full paper of one of these abstracts (4) has been published (Banzhaf *et al* 2016), and all the other abstracts are currently being extended for submission to scientific journals.

[4] Eörs Szathmary Co-leads the FP7-ideas-ERC project EvoEvo (Evolution of Evolvable Systems)

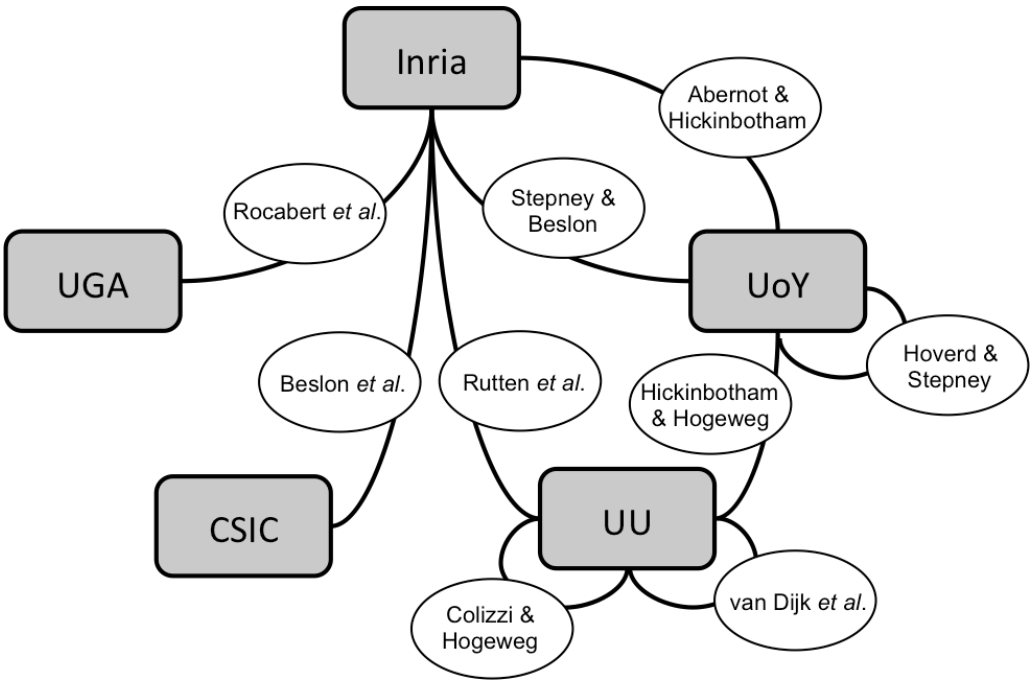


Figure 12: graph of the EvoEvo contributions to the second EvoEvo workshop

### 2.7.5. Internal dissemination

As planned in the DoW we organized frequent tutorials within the consortium to facilitate interdisciplinary exchanges. The objective of these tutorials was to ensure that all members of the consortium had a basic knowledge in all the domains covered by the project and were able to understand (to some degree) the experiments, developments and results of the other members.

These tutorials took place during the general project meetings. During the project we organized the following sessions:

- October 28th 2013: Susan Stepney and Paul Andrews (York) gave a 1.5h lecture on the CoSMoS approach.
- May 22th 2014: Santiago Elena (CSIC) gave a 3h lecture on experimental evolution and virus evolution.
- May 23th 2014: Kirsten ten Tusscher (Utrecht) and Guillaume Beslon (INRIA) gave a 1h lecture on the “Pearls-on-a-String” (PoaS) and “sequence” formalisms in digital genetics.
- October 27th 2014: Susan Stepney (York) gave a 1.5h lecture on unconventional computing
- October 28th 2014: Carole Knibbe (INRIA) gave a 1.5h lecture on the aevol platform and its use for *in silico* experimental evolution
- July 25th 2015: Santiago Elena (CSIC) gave a 3h lecture on the biochemical structure of the Tobacco etch virus (TEV)
- February 23th 2016: CSIS organized a visit of the greenhouse facilities

## 2.8. Conclusion

Given the ambitions of the EvoEvo project (develop new knowledge in life science, computational biology and ICT, and link these three domains), we are proud to say that the project is a clear success: All the technical objectives of the project have been fulfilled (as shown by section 2.2 to

2.6) and the interdisciplinary collaboration has been particularly efficient. The applications developed in WP5 (EvoWave and EvoMove) directly benefit from the knowledge produced by WP1 and WP3 (the so-called "EvoEvo strategies") while WP3 itself has benefited from the model development of WP2. WP4 has successfully produced a general algorithmic framework (EvoMachina) that makes it possible to use EvoEvo strategies in other applications. WP3 and WP1 interacted efficiently and we have been able to replicate *in silico* some of the experimental designs of WP1. On all these aspects, many of the collaborations are still active – and will continue – with numerous publications in preparation (see section 2.7.3).

A hallmark of the quality of the exchange during the project is that, although we did not plan explicit interactions between WP3 and WP4, in practice, some of the EvoEvo strategies identified in WP3 were directly injected in task 4.3 with an exciting result: while self-modifying codes systematically went to extinction in the original stringmol design, the introduction of spatial interactions (an evoevo strategy identified in WP3) permitted a stabilization of the population, and hence a powerful evolution on the long term.

# 3. General insights on the EvoEvo process

## 3.1. Introduction

Variability, robustness and evolvability were considered as independent concepts in the beginning of the project. Our research has elucidated that these properties are strongly entangled in evolving systems. Indeed, they mutually serve as mechanism and consequence of each other.

In this final summary of the results we focus on the most important mechanisms for EvoEvo that we discovered and elucidated in our research. These are: (i) genome structuring, (ii) shaping the genotype-phenotype mapping, and (iii) long term information integration in (multi-level) evolution.

## 3.2. Evolution of genome structure

The first mechanism for EvoEvo is the evolution of the size and the structure of the genome. The size of the genome determines the dimensionality of the evolutionary search space, the overall mutation rate as well as the ratio of different mutational operators. Moreover, the structure of the genome determines the genomic changes these mutational operators can achieve.

We first look at these entangled processes from the point of the genetic operators, and then from the point of genome structure.

### 3.2.1. Roles of genetic operators in evolution of evolution

One of the far-reaching fundamental insights emerging from our research is that different mutational operators, *i.e.* different ways of generating variation, have different roles in evolution, as well as in evolution of evolution.

While most evolutionary theory in biology, as well as in computer science, has focused on point mutations and crossovers, we have highlighted the role of mutations which structure the genome, in particular large and small duplications and deletions, which change the size of the genome, and thereby the dimensionality of the fitness landscape and search space. We have shown that this increases the effectiveness of evolutionary search in several ways. Typically, successful

evolutionary adaptation involves early genome expansion, followed by streamlining, whereas in the absence of genome expansion less fitness is obtained in the end: the gradual reduction of the dimensionality of the search space facilitates optimization. Likewise, these mutational operators open up new dimensions and thereby mediate innovation to escape local optima.

Apart from improving evolutionary optimization, these mutational operators are also instrumental in evolution of evolution in several ways. First, the relative frequency of genomic changes by various mutational operators is modified automatically through their effect on genome size: whereas frequency of changes by point mutations scale linearly with genome size, the frequency scales super-linearly for structural mutations. The effect on fitness of point mutations and duplications and deletions change the structure of the fitness landscape.

Strikingly, through these processes the deleterious effects of high point mutation rates can be compensated by an increase of duplication-deletion mutations, *i.e.* by making total mutation rate even *higher*. Evolutionary response to increased point-mutation rates is to increase (especially large) duplications and deletions and thereby recover, or even improve fitness. This is a very clear demonstration of EvoEvo, shedding new light on the classical problem of the information threshold. This result is independent of a particular model: apart from the *in silico* model of *E. coli*-like mutator strains, it also emerged in a quite different model of transcription-induced mutations in yeast. Both these examples demonstrate the effect of duplication and deletion genetic operators on all hallmarks of EvoEvo: variability, robustness and evolvability.

In conclusion, different genetic operators have quite different effects on the evolutionary process. Only recently has the prevalence of duplications and deletions in biological evolution become known: our research has explained their important role in evolution.

A strong recommendation derived from our research for both biological evolutionary theory as well as for the practice of evolutionary computation is to incorporate a rich set of mutational operators, and to allow for a flexible genome size. To this end, EvoMachina provides a range of evolutionary operators, and supports user-inclusion of more. It also supports a flexible genome size.

### 3.2.2. Evolution of genome structure and the role of non-coding sequences

Structuring of genomes goes beyond the size structuring discussed above, and so does its impact on evolution. First of all, apart from point mutations and duplication and deletions, translocations and inversions play a role in arranging the parts of the genome and thereby the DNA fragments which can be duplicated/deleted. Long-term selection can then lead to genome structuring, thereby biasing mutation to increase robustness and/or evolvability. For example, even given a certain genome size, and given a certain phenotype (fitness), the ratio between the size of genome coding for proteins, regulatory sequences and non-coding junk sequences can vary, and this impacts on the evolutionary process. For example, the *in silico E. coli*-like mutator strain not only increased its genome size but also structured the genome so as to decrease the length of the coding sequences and increase the length of the non-coding sequences. Despite the increased mutation rate, this structuring allowed neutrality (robustness) to remain the same, whilst allowing phenotypic variability, and thereby selection pressure, to increase, by an increase of large duplications and deletion; thereby high fitness could be maintained. Increasing the amount of non-coding sequences also increases the chance that mutational operators other than point mutations destroy coding sequences.

In the Aevol platform and in the Evo$^2$Sim platform, non-coding sequences are explicitly implemented. This is not the case in *e.g.* the virtual cell and virtual microbe platforms. Nevertheless similar effects can be observed, mediated e.g. by genes of minimal effect. Further genome structuring occurs, as more details of the mechanism of the genetic operators are included. For example, when duplications/deletions are mediated by transposon-like mechanisms, hotspots for double stranded breaks are generated, which localize the occurrence of subsequent mutations. In this way, mutations can be biased to particular effects, and thereby increase evolvability.

In conclusion, genome structuring, mediated by a plethora of mutational operators, is a powerful mechanism for EvoEvo. It helps explaining the recent observations of very fast adaptation to novel environments in experimental evolution.

## 3.3. Evolution of the genotype-to-phenotype (GP) map

The second import mechanism for evolution of evolution is the evolution of the GP map. Whereas the evolution of genome structuring influences the amount and types of mutations that occur, the evolution of the GP map influences what fitness effect these mutations have. In other words the evolution of the GP map shapes the mutational neighbourhood at the phenotype level.

A profound general insight derived from our research is the propensity to evolve to a U shaped mutational neighbourhood. That is, there is an overrepresentation of neutral mutants and an overrepresentation of strongly deleterious mutations, whereas slightly deleterious mutations are minimized. This U shape increases robustness in two ways, i.e. by high neutrality (which increases evolvability) and by high selection. It goes beyond the more traditional conceptualization of robustness in terms of flat or steep landscapes, by combining both. The U shape can be skewed in either direction, and tuning the shape is a powerful mechanism of EvoEvo. Note that the U shape can also be reinforced by the genome structuring, discussed above.

Moreover, the evolution of the mutational neighbourhood goes beyond this general, qualitative effect. The particular phenotypic/functional properties of mutants, distinct from those of the focal (master) phenotype, evolve through the evolution of the GP map. A striking example of this emerged in the evolution of RNA sequences at high mutation rates, where a whole ecosystem of different functional types emerged from the close mutational neighbourhood of one particular master sequence. In that case only point mutations were incorporated, demonstrating in that case only evolution of the GP map was involved. In other words, the evolution of the mutational neighbourhood can result in non-random random mutations, in the sense that there is a bias to particular phenotypic changes, despite the randomness of the mutations at the genome level.

More generally, both the evolved GP map and the evolved genome structure shape the mutational neighbourhood. As mentioned above, long-term evolution can thereby enable fast evolutionary adaptation to novel environments.

A powerful mechanism for shaping the mutational neighbourhood is through gene regulation. Gene regulatory networks have multiple attractors, *i.e.* the GP map is a one-to-many map. Mutations can lead to changes in the domain of attraction of these, and/or the loss or gain of attractors, and thereby mediate fast, particular change as well as innovations. Moreover, gene regulation allows for plasticity, i.e. adaptation to changing environmental conditions without mutations, operating on a fast timescale. Obviously such plasticity is an evolved property of the GP map. We found that evolved regulation relative to one type of environmental change greatly enhanced the evolvability

to novel other types of environmental changes. Repeated occurrence of the other types of changes further enhanced evolvability. If the latter type of environmental changes occurred frequently enough, eventually plasticity relative to these also evolved. Strikingly, evolution of evolvability was much faster than evolution of plasticity for intermediate and low frequency changes, and led to very similar average fitness over time. Thus, evolvability is a viable alternative to regulation to cope with variable environments.

We conclude that evolution of the GP map mediates the evolution of robustness and evolvability in various ways, where evolved robustness to mutations or to environmental changes enhances evolvability, which in itself enhances robustness. A prerequisite for the evolution of the GP map is redundant coding.

A strong recommendation derived from this project for biology is to include evolutionary consideration in the study of e.g. gene regulatory network on top of direct functional considerations, and for the practice of evolutionary computation to use such redundant mapping.

The current release of EvoMachina (v2) does not provide direct support for the inclusion of gene regulatory networks. However, its conceptual model and architecture have been designed to facilitate the incorporation of such facilities in future versions.

## 3.4. Multilevel evolution and long-term information integration

Long-term information integration has been a long-term taboo for explaining what has evolved, because without explicit modelling it invites just so stories. Moreover, classical evolutionary models in which evolution is limited to modifying allele frequencies, or few parameters, do not allow long-term information integration.

Nevertheless EvoEvo can only emerge when the evolutionary process integrates information over time, and the models we studied indeed do so. The processes described above, of genome structuring and redundant coding, allow for long-term retention of evolved structures beyond what is functional at the moment. Spatial embedding facilitates long-term information integration, because multiple local sub-populations compete, introducing multiple evolutionary timescales.

Eco-evolutionary models in which interacting replicating entities co-evolve, self-organize into spatial patterns with a dynamics of their own, which behave as higher level evolving entities. Selection pressures on the basic replicators and the higher-level entities can be in conflict, and in order to survive some ’compromise’ has to be reached, often leading to counter-intuitive results, like *e.g.* the evolution of early death. Even more interestingly, it can lead to more complex replication mechanisms, *e.g.* to the evolution of DNA in the RNA world. In other words, the automatic generation of higher levels of evolution is instrumental for major transitions in evolution, and can be seen as stepping-stones for open-ended evolution.

A strong recommendation derived from our studies for evolutionary computation is to embed populations in space, as it improves even simple evolutionary search. Moreover, exploiting long term information integration by using sparse fitness evaluation, in which subproblems co-evolve with solutions, can improve evolutionary search.

To this end, EvoMachina provides support for a range of spatial models, including the traditional 2D toroidal grid. Its model is sufficiently general to support any user-defined hierarchical arrangements of space.

# 4. General insights on living technologies

## 4.1. Introduction

The aim of EvoEvo was to addresses target outcome (a) of the EVLIT objectives:

> *Empirical, theoretical and synthetic approaches that define the key bio-inspired principles that drive future living technologies and the environment to use them in a controlled way.*

In the first part of this report, we described how we studied – and used – EvoEvo strategies to develop software systems able to change their own process and conditions of application through indirect evolution. We focused on the ability of evolution to evolve not only the phenotype of an individual (through mutations at the genome level) but also to evolve the genotype-to-phenotype mapping of this individual, leading to second-order evolution that later-on favours evolution.

In the DoW of the EvoEvo project, we acknowledged that the roadmap toward "Living Technologies" still contains many *terrae incognita* and that, "*nobody really knows what these technologies will look like, how they will be implemented (will it be software, hardware, "wetware" technologies?) nor what they will be useful for (will they complement extant technologies or replace them?)*". That is why we here want to address specifically the question of what such technologies could be – at least in the context of software systems, since in EvoEvo we are not interested in wet-ware technologies. In this context, the present section describes two proposals that emerged during the EvoEvo project. Both proposals address the questions of interaction with living technologies: In section 4.2 (The commensal architecture) we propose a way to organize software systems such that some elements use living technologies while others are more classical – specified – systems. In section 4.3 (Evolving a personal "living" companion) we claim that the performances of our EvoMove system is mainly due to its close integration with the system's user and that such a close integration is a key property for the development of useful living technologies.

## 4.2. The commensal architecture

One of the core universal properties of living beings is their autonomy. Even if some forms of cooperation or altruism can be observed in nature, every biological system is fundamentally selfish and cooperation can only emerge when multiple levels co-evolve, the selfishness of some constraining the cooperativeness to others (like for instance in kin selection in the RNA-model – see section 3.4). Now, one of the core universal properties of technology is its controllability – non-controllable technologies being only allowed in multi-level technologies, the controllability of some level constraining the non-controllability of others (like, for instance, in an internal-combustion engine).

These two antagonistic properties immediately conflict when one wants to design living technologies. As discussed in deliverable D3.4, they also immediately conflict when one wants to

*design* open-ended technologies: if open-ended systems are to continuously produce novelty, how can they be designed? In other words, when designing living technologies, one of the central problems is to design a system that is autonomous enough to surprise its user (by producing novelties) and, at the same time, is rational enough to serve the goals it has been built for (as a technology). Since the very beginning of this project, this tension has been at the heart of EvoEvo: if autonomy is one of the core properties of life, how can a technology be simultaneously *alive* and *controllable*?

As said above, biological systems can be cooperative or altruistic providing they are embedded in higher/lower levels of evolution that constrain them. We propose here a bio-inspired approach to resolve the autonomy *v.* controllability conundrum. We called this approach "commensal computation" (Abernot *et al.*, 2016). In biology a commensal (from the Latin *cum mensa* – at the same table) interaction is a form of mutualism between two organisms where the association is not detrimental but not obviously beneficial to the partners (Hooper and Gordon, 2001). The idea of commensal computation is based on one of the main functions of the gut microbiota: nutrient processing. Gut microbes degrade ingested substances that would otherwise be non-digestible or even harmful to the gut (Hooper *et al.*, 2002). This role enables the organism to uptake nutrients originating from a wider variety of sources than would otherwise be the case. Gut microbes *pre-process* the complex flow of nutrients and transfer the results to their host organism, helping it to regulate its feeding and to extract specific nutrients.

Importantly, while doing so, the microbiota lives their own lives, and change and evolve according to their environment, *i.e.*, according to what the host eats. In addition, the nutrient processing by the microbiota enables the host organism to gain resources it uses to survive. The commensal association of the microbiota and the host contains a part of autonomy (the microbes) and a part of control (the host).

We propose to organise living computational system following the manner in which host and microbiota are engaged in a mutualistic association. In commensal computation, the complex data (*e.g.*, data generated by the sensor networks) are pre-processed by a virtual microbiome that transforms them in digestible data that the processing system can use. Such an architecture differs from classical pre-processing-processing in that here the pre-processing is performed by an *evolving* community of virtual bacteria that uptake data, transform them in recognisable objects (symbols, clusters, classes...) and feed them to the main processing system. In the context of the EvoEvo project, we used a subspace-clustering layer to implement the commensal level: virtual bacteria evolve subspace classifiers and send the result to the processing layer (see section 2.6.2). The interest of subspace classification here is that it enables the sensor network (or more generally the source of data) to change its dimensionality (*e.g.*, adding/removing sensors) without causing a complete failure of the classification: new dimensions can be dynamically added to the system and will (or will not) be integrated to the clustering depending on their pertinence with regards to the existing clusters and to the data.

## 4.3. Evolving a personal "living" companion

When applied to our music generation system based on motion sensors, the commensal architecture results in a host fed by motion data and producing music, and a bacterial community that pre-processes the motion data, helping the host to interpret the moves. Both organisms thus "eat at the same table" (the motion) and co-evolve. The music produced by the host depends on

the command objects produced by the virtual bacteria. The motion fed to the bacteria depends on the movements the user makes in reaction to the music they hear.

Thus, this system creates a feedback loop including the human user. This principle is illustrated in Figure 13. One iteration of the loop is run approximately every second. This timing is short enough to allow interaction. Contrary to most software where the human is acting *on* a system, here the user is acting *in* a system. They do not have full freedom about what sounds will be produced, but they can influence it. They have to decide how they react to what could be called “proposals” from the system, and this decision changes the shape of what the system produces next. And contrary to most of music software, the output of the system is not only the sound produced, but what is produced at each step of the loop and especially what is visible: music and moves.

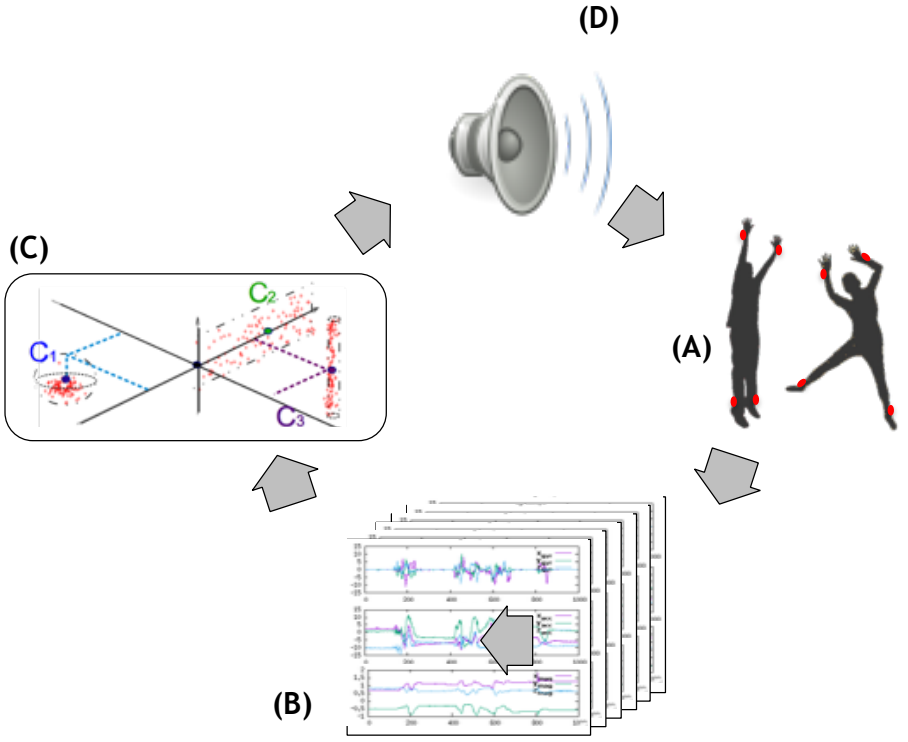


Figure 13: The EvoMove feed-back loop. **(A)** Dancer moves are captured by body sensors (Inertial Measurement Unit) that can be placed on arms, legs, center of mass… (red dots). **(B)** The sensors produce a high-dimensional data-stream. **(C)** This data-stream is clustered by SubCMedian algorithm that outputs a set of clusters. **(D)** The sound system outputs sounds that are immediately perceived by the dancers who can adapt their dance, leading to reciprocal adaptation of the clusters, hence of the music. This feedback loop produces coherent music due to the close integration of dancers and clustering algorithm: the duration of the loop is less than 1 second, enabling real-time response of the system.

These specificities made it impossible for us to anticipate the result of this work before the first test. We had no idea about what could happen, how controllable, or even understandable, the system would be. We thus were sincerely surprised after our first trials by the fact it works so well. Then we had to define what we understand by the statement “it works”, and we found out that what made us say that is the feeling of *interaction* when using it, or even when watching someone using it. Even though this interaction is sometimes hard to describe explicitly, or sometimes could look messy, we were still able to build a representation of the coaction of the human and the system.

This is also surprising compared to what is done in other state-of-the-art move recognition systems used in music control. A common practice is the use of many specific sensors and pre-processing steps, combined with a closely supervised learning algorithm. On the contrary, the EvoMove system uses a few generic sensors, that can be placed anywhere on the body, without any pre-processing or normalization. And yet we reached this “it works” feeling.

Our supposition about what makes this system that efficient, besides the possibilities offered by the SubCMedians algorithm, is the integration of the human user in the feedback loop. As a consequence, the human is always adapting their own moves and actions to fit what they understands of the state of the system. Thus, even though the machine part of the system is deviating from what would be seen as interaction, the human is able to follow it so as to keep this interaction alive. This process does not have to be conscious from the user perspective. Just by investing effort into being understood by the system, the user adapts their actions alongside the system state changes.

### 4.4. Conclusion

As the 2011 FET Consultation Report[5] “Living Technology, Artificial Systems, Embodied Evolution” shows, many approaches have been proposed to create living technologies. Now at the end of the EvoEvo project, and having created what we think *is* a living technology (“EvoMove”), we claim that the key insight into building living technologies is to go back to a fundamental property of living systems. Living systems are in essence strongly *integrated* systems, while technological systems are, by construction, strongly modular systems. Living technologies will only be efficient if they are strongly integrated with their users, be it a real person (as in EvoMove), or a software entity (as in commensal architecture).

In some sense, this proposal is not a total surprise, since it is a similar mindshift as the one that happened at the end of the 1980s in robotics. The development of Behavior-Based Robotics by *e.g.* Rodney Brooks was nothing other than the close integration of robots with their environments and of robots’ components one with the others. We now propose that software systems themselves, although they are not physical entities, follow the same path in order to be able to dynamically react and adapt to their users. This will enable living software systems to co-construct their behaviour with a user who, in that same moment, will become a *partner* of this behaviour.

## 5. Conclusion

In three years, the EvoEvo project produced numerous outcomes, as this report shows: more than 50 international publications and communications have been produced and more than a dozen are still in preparation. However, maybe more than the number of results, the wide disciplinary and interdisciplinary range of these results is a clear mark of success for this project that, from its very beginning to its end, placed a bet on interdisciplinary collaborations.

It is well known that interdisciplinary research is difficult. In the case of EvoEvo, we would like to emphasize the fact that all partners collaborated throughout the project whatever their scientific background was. They all learnt to discuss together, and all were highly tolerant of the repeated questions and misunderstanding that necessarily occurs throughout such a project.

[5] http://cordis.europa.eu/fp7/ict/fet-proactive/docs/shapefetip-wp2013-03_en.pdf

We are pleased to announce that all the objectives of the project have been fully met during these three years:

- Many EvoEvo strategies have been identified in the two biological models used in WP1 (TEV and *E. coli*). All these strategies have been published in high-ranked journals.
- An integrated model dedicated to the computational study of EvoEvo has been developed and tested in WP2 (Evo$^2$Sim). This model is available for the scientific community on the project website (http://www.evoevo.eu/evo2sim-software/).
- Using Evo$^2$Sim and other models available in the different teams collaborating on the project (Aevol, R-Aevol, virtual cell, virtual microbe…) we performed different *in silico* evolutionary experiments (including models of the experiments performed in WP1). Results of these experiments enabled us to better characterize the EvoEvo strategies identified in WP1 and, in some cases, to help understanding the results of the wet experiments (WP3).
- In WP4 the main EvoEvo strategies have been used to design and implement a computational platform EvoMachina that enables EvoEvo exploitation for Artificial Evolution applications. This platform has been tested on two different applications (TSP and subspace clustering) and is available for the scientific community on the project website (http://www.evoevo.eu/evomachina/). Additionally, WP4 developed a novel "bio-reflective architecture", and used it in experiments of evolving machines encoded on genomes, thereby demonstrating semantic closure of the integrated system.
- Finally, in WP5 the EvoEvo strategies have been implemented in two demonstrator applications. Both applications are based on evolutionary subspace clustering algorithms developed during the project (http://evoevo.liris.cnrs.fr/chameleoclust/). The first application (EvoWave) uses an evolutionary clustering algorithm to identify the working context from the WiFi signals received by a machine. The second application (EvoMove) uses it to classify the moves of a dancer and to learn to play music according to the choreography.

Given the ambitious goals of the EvoEvo project, a three years project could only superficially dig into the innovative concepts that were at the core of the project (evolution of evolution, living technologies, open-endedness, bio-reflective architectures…). EvoEvo has successfully opened many opportunities for future researches and technical developments. The question is now opened of the future of this consortium and of the continuation of this research. Two options are possible: going on with the same interdisciplinary consortium – or with a very similar one – or expand the consortium to launch different projects in the different disciplines covered by EvoEvo. These two options are not exclusive, but the latter is simpler than the former. Many smaller projects could immediately start to continue the collaborations. Indeed, most of the collaborations that started within the EvoEvo project are going on. In this view, we have applied for an H2020 FET grant, the Innovation Launchpad. The objective of this project (Evo2Move) is to continue the collaboration between INRIA and University of York on the development of the musical personal companion (the "living instrument") and to enlarge its domain of application (we have been contacted by physiotherapists who are strongly interested in EvoMove as a potential rehabilitative tool). Other similar projects will probably be launched in a near future, *e.g.*, to go on with the modelling of laboratory experiments. Discussions are currently underway about the opportunity to submit another project to the next Innovation Launchpad call (September 2017) or to the EIT Health – *e.g.*, on the development of evolutionary models that could be used for education. But these smaller projects must not hide the fact that, given the success of the EvoEvo project and the

sustained quality of the interactions between the partners, we are all ready to repeat the EvoEvo experience and start an EvoEvo2 project.

# 6. References

Abernot J., Beslon G., Hickinbotham S., Peigner S., Rigotti C. (2016) A Commensal Architecture for Evolving Living Instruments. In: *proceedings of Conference on Computer Simulation of Musical Creativity*, Huddersfield, United Kingdom, June 2016, 8 p.

Adami, C. (2006) Digital genetics: unravelling the genetic basis of evolution. *Nature Reviews Genetics*, 7:109

Andrews P. S., Stepney S. (2014) Using CoSMoS to Reverse Engineer a Domain Model for Aevol. *CoSMoS workshop*, New York, USA, July 2014, pp.61-79. Luniver Press

Andrews P. S., Stepney S. (2015a) A Metamodel for the Evolution of Evolution. *In proceedings of ECAL 2015*, York, UK, July 2015, pp. 621-628. MIT Press

Andrews P. S., Stepney S. (2015b) The CoSMoS Domain Experiment Model. *CoSMoS workshop*, York, UK, July 2015 pp.1–8. Luniver Press

Banzhaf W., Baumgaertner B., Beslon G., Doursat R., Foster J. A., McMullin B., Veloso de Melo V., Miconi T., Spector L., Stepney S., White R. (2016) Defining and Simulating Open-Ended Novelty: Requirements, Guidelines, and Challenges. *Theory in Biosciences*, 135(3):131-161

Bar-Even A., Noor E., Savir Y., Liebermeister W., Davidi D., Tawfik D. S., Milo R. (2011) The Moderately Efficient Enzyme: Evolutionary and Physicochemical Trends Shaping Enzyme Parameters. *Biochemistry,* 50(21):4402-4410

Batut B., Beslon G., Knibbe C. (2016a). Unexpected genome inflation and streamlining in variable environments. Journées ouvertes de Biologie Informatique & Mathématiques 2016, June 2016, Lyon, France

Batut B., Beslon G., Knibbe. (2016b): Genome inflation and streamlining in variable environments. *In prep*

Bernet G. P., Elena S. F. (2015) Distribution of mutational fitness effects and of epistasis in the 5' untranslated region of a plant RNA virus. *BMC Evol. Biol*. 15(274)

Beslon G., Liard V., Elena SF (2016a) Evolvability drives innovation in viral genomes. *2nd EvoEvo Workshop, satellite workshop of CCS2016*, Sep 2016, Amsterdam, NL, 6 p.

Beslon G., Liard V., Elena S. F. (2016b) Evolvability drives innovation in viral genomes. *In prep*

Cervera H., Lalić J., Elena S. F. (2016a). Efficient escape from local optima in a highly rugged fitness landscape by evolving RNA virus populations. *Proc. R. Soc. B*. 283: 20160984.

Cervera H., Lalić J., Elena S. F. (2016b) Effect of host species on the topography of fitness landscape for a plant RNA virus. *J. Virol*. PMID: 27581976.

Clark E. B., Hickinbotham S., Stepney S. (2016) Semantic closure demonstrated by the evolution of universal constructor in stringmol. *submitted to Royal Society Interface*.

Cuevas J. M., Willemsen A., Hillung J., Zwart M. P., Elena S. F. (2015) Temporal dynamics of intrahost molecular evolution for a plant RNA virus. *Mol. Biol. Evol*. 32:1132-1147

Cuypers T. D., Hogeweg P. (2015) Endless evolutionary paths to Virtual Microbes.Workshop, *First EvoEvo Workshop, satellite workshop of ECAL2015*, July 2015, York, UK, 1 p.

Colizzi E. S., Hogeweg P. (2014) Evolution of functional diversification within quasispecies. *Genome biology and evolution*, 6(8):1990-2007

Colizzi E. S., Hogeweg P. (2016a) Mutational load is ameliorated by increased transcriptional load-associated mutations, if these are biased towards duplications and deletions. *2nd EvoEvo Workshop, satellite workshop of CCS2016*, Sep 2016, Amsterdam, NL, 5 p.

Colizzi E. S., Hogeweg P. (2016b) Increased rate of duplications and deletions prevents evolutionary deterioration; Understanding the mutational dynamics of yeast rDNA. In: *Colizzi E. S., Multilevel Evolution and the Emergence of Function. PhD thesis* Utrecht University 2016 ISBN 978 90 393 6692 9

Colizzi E. S., Hogeweg P. (2016c) High cost enhances cooperation through the interplay between evolution and self-organization. *BMC evolutionary biology*, 16:31

Consuegra J., Plucain J., Gaffé J., Lenski R. E., Hindré T., Schneider D. (2016) Molecular genetics of a new ecological opportunity exploitation during long-term bacterial sympatric adaptive diversification. *In prep*

Crombach A., Hogeweg P. (2007) Chromosome rearrangements and the evolution of genome structuring and adaptability, *Molecular Biology and Evolution,* 24(5):1130-1139.

Crombach A., Hogeweg P. (2008) Evolution of evolvability in gene regulatory networks. *PLoS computational biology*, 4(7):e1000112

Cuypers T. D., Hogeweg P. (2012) Virtual genomes in flux: an interplay of neutrality and adaptability explains genome expansion and streamlining. *Genome biology and evolution*, 4(3): 212-229

Cuypers T. D., Hogeweg P. (2014) A Synergism between Adaptive Effects and Evolvability Drives Whole Genome Duplication to Fixation. *PLoS computational biology*, 10(4):e1003547

Cuypers T. D., Rutten J. P., Hogeweg P. (2016) Mutate or Regulate: evolutionary strategies along a continuum of ecological time scales. *Submitted.*

de Boer F. K., Hogeweg P. (2012) Co-evolution and ecosystem based problem solving. *Ecological Informatics* 9:47-58

de Boer F. K., Hogeweg P. (2014) Mutation rates and evolution of multiple coding in RNA-based protocells. *Journal of molecular evolution,* 79(5-6):193-203

Dunham M. J., Badrane H., Ferea T., Adams J., Brown P. O., Rosenzweig F., Botstein D. (2002) Characteristic genome rearrangements in experimental evolution of Saccharomyces cerevisiae, *Proc. Natl. Acad. Sci. USA*., 99(25):16144-16149

Elena S. F., Wilke C. O., Ofria C., Lenski R. E. (2007) Effects of population size and mutation rate on the evolution of mutational robustness. *Evolution*, 61(3):666-674.

Elena S. F. (2016a) Evolutionary transitions during RNA virus experimental evolution. *Phil. Trans. R. Soc. B.* 371: 20150441.

Elena S. F. (2016b) Local adaptation of plant viruses: lessons from experimental evolution. *Mol. Ecol*. PMID 27612225.

Fischer S., Bernard S., Beslon G., Knibbe C. (2014) A model for genome size evolution. *Bulletin of mathematical biology*, 76(9): 2249–2291

Funes P., Pollack J. (1999) Computer Evolution of Buildable Objects. *In Evolutionary Design by Computers*. P. Bentley (editor). Morgan Kaufmann, San Francisco. pp. 387-403.

Großkopf T, Consuegra J, Gaffé J, Willison JC, Lenski RE, Soyer OS, Schneider D. (2016) Metabolic modelling in a dynamic evolutionary framework predicts adaptive diversification of bacteria in a long-term evolution experiment. *BMC Evol. Biol*. 16(1)**:**163.

Hickinbotham S., Hogeweg P. (2016) Evolution towards extinction in replicase models: inevitable unless... *2nd EvoEvo Workshop, satellite workshop of CCS2016*, Sep 2016, Amsterdam, NL, 5 p.

Hickinbotham S., Stepney S. (2015a) Conservation of matter increases evolutionary activity. *In proceedings of ECAL 2015,* York, UK, July 2015, pp. 98-105

Hickinbotham S., Stepney S. (2015b) Environmental bias forces parasitism in Tierra. *In proceedings of ECAL 2015*, York, UK, July 2015, pp. 294-301

Hickinbotham S., Stepney S. (2016a) Bio-Reflective Architectures for Evolutionary Innovation. *In proceedings of ALife 2016*, Cancun, Mexico, July 2016, pp. 192-199

Hickinbotham S., Stepney S. (2016b) Augmenting Live Coding with Evolved Patterns. *In proceedings of EvoMusArt*, Porto, Portugal, March 2016, LNCS 9596:31-46

Hillung J., Cuevas J. M., Elena S. F. (2015) Evaluating the within-host fitess effects of mutations fixed during virus adaptatyion to different ecotypes of a new host. *Phil. Trans. R. Soc. B*. 370 : 20140292.

Hillung J., Cuevas J. M., Valverde S., Elena S. F. (2014) Experimental evolution of an emerging plant virus in host genotypes that differ in their susceptibility to infection. *Evolution* 68: 2467-2480.

Hillung J., García-García F., Dopazo J., Cuevas J. M., Elena S.F. (2016) The transcriptomics of an experimentally evolved plant-virus interaction. *Sci. Rep*. 6: 24901.

Hooper L. V., Gordon J. I. (2001). Commensal host-bacterial relationships in the gut. *Science*, 292:1115-1118.

Hooper L. V., Midtvedt T., Gordon J. I. (2002) How host-microbial interactions shape the nutrient environment of the mammalian intestine. *Annual review of nutrition*, 22(1):283-307

Hoverd T., Stepney S. (2016) EvoMachina: a novel evolutionary algorithm inspired by bacterial genome reorganisation. *2nd EvoEvo workshop*, Amsterdam, Netherlands.

Hruschka E. R., Barreto Campello R. J. G., Freitas A. A., Ferreira de Carvalho A. C. P. L. (2009) A survey of evolutionary algorithms for clustering. *IEEE Transactions on Systems, Man, and Cybernetics*, 39(2):133-155

Huynen M. A. (1996) Exploring phenotype space through neutral evolution. *Journal of molecular evolution*, 43(3):165-169

Huynen M. A., Stadler P. F., Fontana W. (1996) Smoothness within ruggedness: the role of neutrality in adaptation. *PNAS*, 93(1):397-401

Knibbe, C., Coulon, A., Mazet, O., Fayard, J. M., Beslon, G. (2007) A long-term evolutionary pressure on the amount of noncoding DNA. *Molecular biology and evolution*, 24(10):2344-2353

Knibbe C., Parsons D. (2014) What happened to my genes? Insights on gene family dynamics from digital genetics experiments. *In proceedings of ALIFE 14, 14th Intl. Conf. on the Synthesis and Simulation of Living Systems*. New-York (US), July 2014, pp. 33-40

Kriegel H.-P., Kröger P., Zimek A. (2009) Clustering highdimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data*, 3(1), Article No. 1, pp. 1-58

Krakauer D. C., Plotkin J. B. (2002) Redundancy, antiredundancy, and the robustness of genomes. *Proc. Natl. Acad. Sci. USA,* 99(3):1405-1409

Lalić J., Elena S. F. (2015) The impact of high-order epistasis in the within-host fitness of a positive-sense plant RNA virus. *J. Evol. Biol*. 28: 2236-2247.

Lamrabet O., Plumbridge J., Lenski R. E., Hindré T., Schneider D. (2016) Dynamics of altered regulatory networks in bacteria. *In prep*

Lipson H., Pollack J. B. (2000) Automatic design and Manufacture of Robotic Lifeforms, *Nature*, 406(6799):974-978.

Liu Y., Jiao L. C., Shang F. (2013) An efficient matrix factorization based low-rank representation for subspace clustering. *Pattern Recognition*, 46(1):284-292

Majer E., Salvador Z., Zwart M. P., Willemsen A., Elena S. F., Daròs J. A. (2014) Relocation of the *NIb* gene in the tobacco etch potyvirus genome. *J. Virol*. 88:4586-4590

Meijer J., van Dijk B., Cuypers T., Hogeweg P. (2016) The role of HGT in the evolution of genomes, transcriptomes, metabolomes and ecosystems. *In prep*

Müller E., Günnemann S., Assent I., Seidl T. (2009) Evaluating clustering in subspace projections of high dimensional data. *In Proceedings of the 35th Int. Conf. on Very Large Data Bases (VLDB 2009)*, Lyon, France, pp. 1270-1281

Parsons L., Haque E., Liu H. (2004) Subspace clustering for high dimensional data: A review. *SIGKDD Explorations Newsletter*, 6(1):90-105

Patrikainen A., Meila M. (2006) Comparing subspace clusterings. *IEEE Transactions on Knowledge and Data Engineering*, 18(7), 902-916

Peigner S., Rigotti C., Beslon G. (2015a) Subspace Clustering for all Seasons. *First EvoEvo Workshop, satellite workshop of ECAL2015*, July 2015, York, UK, 1 p.

Peignier S., Rigotti C., Beslon G. (2015b) Subspace clustering using evolvable genome structure. *In: proceedings of GECCO'15, Annual Conference on Genetic and Evolutionary Computation*. Madrid (Spain), July 2015, pp. 575-582

Plucain J., Hindré T., Le Gac M., Tenaillon O., Cruveiller S., Médigue C., Leiby N., Harcombe W. R., Marx C. J., Lenski R. E., Schneider D. (2014) Epistasis and allele specificity in the emergence of a stable polymorphism in *Escherichia coli*. *Science* 343:1366-1369.

Raeside C., Gaffé J., Deatherage D. E., Tenaillon O., Briska A. M., Ptashkin R. N., Cruveiller S., Médigue C., Lenski R. E., Barrick J. E., Schneider D. (2014) Large chromosomal rearrangements during a long-term evolution experiment with *Escherichia coli*. *mBio* ***5***:e01377-14

Rocabert C., Knibbe C., Beslon G. (2015) Towards an Integrated Evolutionary Model to Study Evolution of Evolution. *First EvoEvo Workshop, satellite workshop of ECAL2015*, July 2015, York, UK, 15 p.

Rocabert C., Knibbe C., Consuegra J., Schneider D., Beslon G. (2016a) In Silico Experimental Evolution Highlights the Influence of Environmental Seasonality on Bacterial Diversification. *2nd EvoEvo Workshop, satellite workshop of CCS2016*, Sep 2016, Amsterdam, NL, 4 p.

Rocabert C., Knibbe C., Consuegra J., Schneider D., Beslon G. (2016b) Beware Batch Culture: Seasonality and Niche Construction Predicted to Favor Bacterial Adaptive Diversification. *Submitted to PLoS Computational Biology*

Rudan M., Schneider D., Warnecke T., Krisko A. (2015) RNA chaperones buffer deleterious mutations in *E. coli*. *eLife* 4:e04745

Rutten J., Hogeweg P., Beslon G. (2016a) Evolution of mutator populations in constant environments. *2nd EvoEvo Workshop, satellite workshop of CCS2016*, Sep 2016, Amsterdam, NL, 5 p.

Rutten J. P., Hogeweg P., Beslon G. (2016b) Adapting the engine to the fuel: hypermutator populations can escape the mutational burden by reorganizing their genome structure. *In prep*

Sarafis I. A., Trinder P. W., Zalzala A. M. S. (2003) Towards effective subspace clustering with an evolutionary algorithm. *In Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2003)*, pp 797-806

Sim K., Gopalkrishnan V., Zimek A., Cong G. (2013) A survey on enhanced subspace clustering. *Data Mining and Knowledge Discovery*, 26(2):332-397

Sims K. (1994) Evolving 3D Morphology and Behavior by Competition, *In Proc. Of Artificial Life IV*, MIT Press, pp. 28-39

Takeuchi N., Hogeweg P. (2008) Evolution of complexity in RNA-like replicator systems. *Biology Direct*, 3:11

Takeuchi N., Hogeweg P., Koonin E. V. (2011) On the Origin of DNA Genomes: Evolution of the Division of Labor between Template and Catalyst in Model Replicator Systems. *PLoS Computational Biology*, 7(3): e1002024

Takeuchi N., Hogeweg P., Kaneko K., (2016a) Evolution as the “means” and “end” of the origin of life. *Invited paper for The Royal Society Philosophical Transactions A*, submitted.

Takeuchi N., Kaneko K., Hogeweg P. (2016b) Evolutionarily stable disequilibrium: endless dynamics of evolution in a stationary population. *Proceedings of the Royal Society B: Biological Sciences*, 283(1830). pii: 20153109

Tenaillon O., Barrick J. E., Ribeck N., Deatherage D. E., Blanchard J. L., Dasgupta A., Wu G. C., Wielgoss S., Cruveiller S., Médigue C., Schneider D., Lenski R. E. (2016) Tempo and mode of genome evolution in a 50,000-generation experiment. *Nature* 536:165-170

Ten Tusscher K. H., Hogeweg P. (2011) Evolution of networks for body plan patterning; interplay of modularity, robustness and evolvability. *PLoS Computational Biology*. 7(10):e1002208

Tromas N., Zwart M. P., Forment J., Elena S. F. (2014) Shrinkage of genome size in a plant virus upon transfer of an essential viral gene into the host genome. *Genome Biol. Evol*. 6:538-550

Vahdat A., Heywood M. I., Zincir-Heywood A. N. (2010) Bottom-up evolutionary subspace clustering. *In Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2010)*, pp 1-8

van Dijk B., Hogeweg P. (2016) In silico gene-level evolution explains microbial population diversity through differential gene mobility. *Genome biology and evolution*, 28;8(1):176-188

van Dijk B., Cuypers T. D., Hogeweg P. (2016a) Evolution of r- and K-selected species of Virtual Microbes: a case study in a simple fluctuating 2-resource environment . 2nd EvoEvo Workshop, satellite workshop of CCS2016, Sep 2016, Amsterdam, NL, 5 p.

van Dijk B., Cuypers T. D., Hogeweg P. (2016b) Evolution of the stringent response in virtual microbes, *in prep*

van Nimwegen E., Crutchfield J. P., Huynen M. A. (1999) Neutral evolution of mutational robustness. *PNAS*, 96(17):9716-9720

Vroomans R. M., Hogeweg P., Ten Tusscher K. H. (2016) In silico evo-devo: reconstructing stages in the evolution of animal segmentation. *Evodevo*, 1;7:14.

Wang Y., Zhu J. (2015) Dp-space: Bayesian nonparametric subspace clustering with small-variance asymptotics. *In Proceedings of International Conference on Machine Learning*, pp 1-9

Wielgoss S., Hindré T., Lenski R. E., Schneider D. (2016) Long-term evolution of global gene expression through pervasive changes in epistatic interactions within regulatory networks. *In prep.*

Wilke C. O., Wang J. L., Ofria C., Lenski R. E., Adami C. (2001) Evolution of digital organisms at high mutation rates leads to survival of the flattest. *Nature*, 412(6844):331-333

Willemsen A., Zwart M. P., Higueras P., Sardanyés J., Elena S. F. (2016a) Predicting the stability of homologous gene duplications in a plant RNA virus. *Genome Biol. Evol*. 8:3065-3082.

Willemsen A., Zwart M. P., Tromas N., Majer E., Daròs J. A., Elena S. F. (2016b) Multiple barriers to the evolution of alternative gene orders in a positive-strand RNA virus. *Genetics* 202:1503-1521.

Willemsen A., Zwart M. P., Elena S. F. (2016c) High virulence does not necessarily impede viral adaptation to a new host: a case study using a plant RNA virus. *bioRxiv* doi:10.1101/060137.

Willemsen A., Zwart M. P., Ambrós S., Carrasco J. L., Elena S. F. (2016d) *2b* or not *2b*: experimental evolution of functional exogeneous sequences in a plant RNA virus. *bioRxiv* doi:10.1101/079970.

Zwart M. P., Willemsen A., Daròs J. A., Elena S. F. (2014) Experimental evolution of pseudogenization and gene loss in a plant RNA virus. *Mol. Biol. Evol*. 31:121-134.