HAL
open science

Evolutionary optimization of self-assembly in a swarm of bio-micro-robots

Nathanael Aubert-Kato, Charles Fosseprez, Guillaume Gines, Ibuki Kawamata, Huy Dinh, Leo Cazenille, Andre Estevez-Tores, Masami Hagiya, Yannick Rondelez, Nicolas Bredeche

To cite this version:

Nathanael Aubert-Kato, Charles Fosseprez, Guillaume Gines, Ibuki Kawamata, Huy Dinh, et al.. Evolutionary optimization of self-assembly in a swarm of bio-micro-robots. Genetic and Evolutionary Computation Conference (GECCO '17), Jul 2017, Berlin, Germany. 10.1145/3071178.3071289 . hal-03313837

HAL Id: hal-03313837

https://hal.sorbonne-universite.fr/hal-03313837

Submitted on 4 Aug 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Evolutionary Optimization of Self-Assembly in a Swarm of Bio-micro-robots

Nathanael Aubert-Kato*
ELSI, Tokyo Institute of Technology & Ochanomizu University
Tokyo, Japan

Charles Fosseprez
UMR 8231 (CBI)
CNRS/ESPCI & MNHN
Paris, France

Guillaume Gines
UMR 7083 (Gulliver)
CNRS/ESPCI
Paris, France

Ibuki Kawamata
Tohoku University
Sendai, Japan

Huy Dinh
Playnext Lab Inc
Tokyo, Japan

Leo Cazenille
Sorbonne Paris Cité & Sorbonne Universités, CNRS
Paris, France

Andre Estevez-Tores
Sorbonne Universités
UPMC Univ Paris 06, CNRS, LJP
F-75005 Paris, France

Masami Hagiya
University of Tokyo
Tokyo, Japan

Yannick Rondelez
UMR 7083 (Gulliver)
CNRS/ESPCI
Paris, France

Nicolas Bredeche†
Sorbonne Universités
UPMC Univ Paris 06, CNRS, ISIR
F-75005 Paris, France

ABSTRACT

This paper deals with the programmability of a swarm of bio-micro-robots in order to display self-assembling behaviors into specific shapes. We consider robots that are DNA-functionalized micro-beads capable of sensing and expressing signals as well as self-assembling. We describe an *in vitro* experimentation with a million of micro-beads conditionally aggregating into clusters. Using a realistic simulation, we then address the question of how to automatically design the reaction networks that define the micro-robots' behavior, to self-assemble into a specific shape at a specific location. We use bioNEAT, an instantiation of the famous NEAT algorithm capable of handling chemical reaction networks, and CMA-ES to optimize the behavior of each micro-bead. As in swarm robotics, each micro-bead shares the same behavioral rules and the general outcome depends on interactions between neighbors and with the environment. Results obtained on four different target functions show that solutions optimized with evolutionary algorithms display efficient self-assembling behaviors, improving over pure hand-designed networks provided by an expert after a week-long trials and errors search. In addition, we show that evolved solutions are able to self-repair after damage, which is a critical property for smart materials.

CCS CONCEPTS

•**Applied computing → Systems biology;** Chemistry; •**Computer systems organization → Robotics;** •**Computing methodologies → Artificial intelligence;**

KEYWORDS

Bio-micro-robots, molecular programming, evolutionary robotics, swarm robotics, collective behavior, self-assembly, sef-repair

ACM Reference format:
Nathanael Aubert-Kato[1], Charles Fosseprez, Guillaume Gines, Ibuki Kawamata, Huy Dinh, Leo Cazenille, Andre Estevez-Tores, Masami Hagiya, Yannick Rondelez, and Nicolas Bredeche[2]. 2017. Evolutionary Optimization of Self-Assembly in a Swarm of Bio-micro-robots. In *Proceedings of GECCO '17, Berlin, Germany, July 15-19, 2017,* 8 pages.
DOI: http://dx.doi.org/10.1145/3071178.3071289

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO '17, Berlin, Germany

© 2017 ACM. 978-1-4503-4920-8/17/07…$15.00

DOI: http://dx.doi.org/10.1145/3071178.3071289

1 INTRODUCTION

Swarm robotics considers large number of autonomous robots capable of achieving complex organization based solely on local interactions. These robots are usually endowed with limited sensing, acting and communication capabilities, with the same decision making

[1]Corresponding author 1: naubertkato@is.ocha.ac.jp
[2]Corresponding author 2: nicolas.bredeche@upmc.fr

rules being duplicated on each robot. Swarm robotics aim to achieve complex behaviors through self-organization: decisions that occur at the microscopic scale may have consequences observable at the macroscopic scale.

We are interested in the process of self-assembly: the ability for a swarm of robots to assemble into a specific shape through self-organization principles [1–5]. In contrast to previous works in swarm robotics, we consider robots at the *microscopic* scale. Long-term applications for swarms of micro-robots capable of self-assembling are numerous, such as building smart materials (manufacturing of micro-structures in a distributed fashion or building a structure that is resilient with respect to damage or changing environmental conditions) or developing new functionalities for nanomedecine (space-filling, encapsulation of a specific target element, bio-sensing).

In this paper, we use a swarm of micro-beads of $1 - 10\mu$m in diameter. Each bead is functionalized by sticking bio-molecules (single strands DNA) to the bead's surface. This enables sensing, communication, and aggregation between micro-beads. Section 2 presents the full setup as well as an *in vitro* demonstration of approx. one million of micro-beads self-assembling to form clusters in a 2-dimensional environment. One major advantage of such a setup is that someone with the required expertise can engineer $20 - 100$ millions of micro-beads per milliliter within minutes using easily accessible chemical material, and let them run autonomously in a closed chamber of a few squared centimeters containing a few μL of solutions for hours.

However, designing the program that is to be sticked to each micro-bead poses a major challenge as the dynamics of chemical systems are *highly* non-linear. Single strands DNA interact with one another in a fashion that makes it extremely difficult, even for an expert, to program simple tasks. We address this challenge *in simulation* by using bioNEAT [6], an instantiation of the famous NEAT algorithm [7] modified to evolve chemical reaction networks with positive and negative feedbacks between chemical species. We show that bioNEAT is capable of finding reaction networks that are both non-intuitive and comparable in performance to reaction networks designed by human experts, but in much shorter time. We also show that both expert- and bioNEAT-designed solutions benefit greatly from additional automatic parameter tuning (using CMA-ES [8]), improving both performance and resilience, i.e. the ability to self-repair after damage.

The paper is organized as follows. Section 2 presents the methods and includes two contributions: the description of, as far as we know, the first *in vitro* experiment of a self-assembling swarm of approx. one million micro-beads, and a simulator capable of capturing the non-linear dynamics of such chemical systems, which is used in the following Sections. Section 3 then presents the experimental protocol, including the four targets used for evaluating self-assembly. Section 4 presents the results on performance, including a comparison between evolved and expert-designed reaction chemical networks, and an evaluation of the resilience of the micro-robot swarm to damage (i.e. self-repair). Finally, Section 5 provides a discussion with respect to both chemical and computational issues.

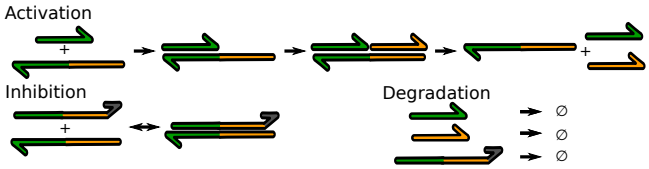

Figure 1: The three modules of the PEN toolbox: activation, where a signal strand attaches to a template, triggering the enzymatic production of another signal strand; inhibition, where a specific signal strand can prevent activity on a target template until release; and degradation, where all signal strands in the system are enzymatically destroyed over time.

2 METHODS

2.1 Molecular programming

Molecular programming [9] is the use of molecules (such as DNA, RNA and/or proteins) to process information. This paradigm relies on chemical concentrations to represent data and chemical reactions to transform data. *In vitro* demonstrations of molecular programming are numerous: computing a square root [10], emulating neural networks [11], assembling DNA nano-structures [12], encoding a toggle-switch [13], implementing a 2-dimensional predator-prey system [14], to cite a few.

In this paper, we use the PEN (Polymerase, Exonuclease, Nickase) toolbox [15], a molecular programming approach relying on the interaction between DNA molecules and enzymes to encode three basic operations (or modules): activation, inhibition and degradation (cf. Fig. 1). Specifically, the PEN toolbox distinguish between two types of DNA molecules: short (11 to 13 bases long) *signal* strands and longer (22 to 25 bases long) *template* strands. Signal strands can attach to complementary templates to produce (activation) other signal strand, or temporarily inhibit activity (inhibition). Moreover, signal strands are continuously degraded by one of the enzymes (degradation). Template strands are chemically protected against degradation. As such, the concentration of signal strands changes through time depending on their interactions with the template strands.

PEN toolbox activation and inhibition modules can be combined in an arbitrary fashion, forming a program. As an example, a simple bistable system is shown in Fig. 2. A wide range of behaviors have been implemented so far, either *in vitro*, such as an oscillator [15] or a toggle switch [13] or in-silico, such as a two-bit counter [6] or a simple controller for an abstract molecular robot [16].

Gines *et al.* have recently demonstrated that PEN toolbox templates could be easily attached to agarose micro-beads, thus localizing PEN toolbox programs, while signal could still be exchanged through diffusion [17]. This approach is comparable to having communicating agents at the micro-scale, and inspired the work in this paper.

At the same time, other groups have shown that DNA strands are strong enough to bind together colloids [18] or large structures like oil-water emulsions [19]. Combining the PEN toolbox, the use of micro-beads, and the fact that DNA strands can be used as physical bond between larger objects, we have conducted additional *in vitro*

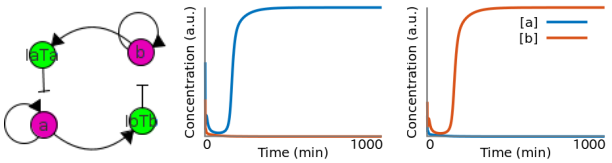

Figure 2: Left: graph representation of a bistable switch. Nodes are signal strands, arrows represent activation through template strands and bar-headed arrows represent inhibition. For convenience, activating signals are represented in magenta and inhibiting signals in green. Right: temporal evolution of concentration in a well-mixed system, based on two different initial conditions: low *a* (resp. *b*) concentration and high *b* (resp. *a*) concentration leads to a state where *b* (resp. *a*) is expressed.

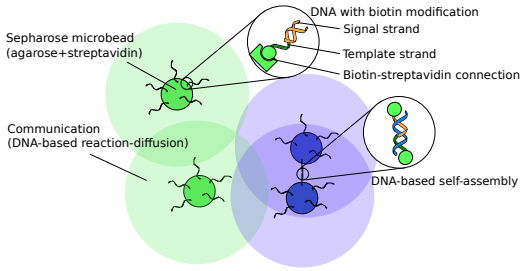

Figure 3: Implementation of control and self-assembly with micro-beads. DNA strands chemically bonded to a biotin attach to beads through biotin-streptavidin interactions. Those DNA strands retain their activity as part of the PEN toolbox, allowing signal sensing and expression, while special anchoring strands can assemble beads.

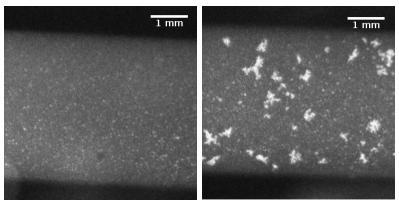

Figure 4: In vitro experiment with beads in water (control, left) or with signal activating the production of self-assembly strand (right). When self-assembly strands are produced in the environment, beads self-assemble in an unspecific way. Experiment was run for 13 hours.

experiment to provide a proof-of-concept of self-assembly at the microscopic scale.

We enrich the original design to add the possibility for beads to self-assemble by using a specific signal strand and specially designed strands also attached to the beads. This results in extending the PEN toolbox with an additional module: self-assembly. Fig. 3 illustrates the implementation of the self-assembly module: *anchoring* strands (around 40 bases long) are attached to the micro-beads, and micro-beads can anchor to one another through a signal strands that bind together two anchoring strands.

Fig. 4 shows the results of an *in vitro* experiment of self-assembling micro-beads. First, a batch of 1μm of micro-beads is placed in a 2-dimensional chamber filled with a solution containing the template strands that can produce the signal strands used for binding. Template strands are initially inactive, but adding a specific trigger (i.e. a signal strand) initiates the production of the signal strands used for binding. As micro-beads randomly move around due to Brownian motion, they attach to one another when close enough, gradually forming clusters. Appendix A provides details for this experiment.

Therefore, micro-beads can be seen as forming a robot swarm. Each micro-bead is embodied in the environment and can both:

- **sense** its environment, through interaction between signal strands from the environment and template strands on the micro-bead;
- **act**, either by producing signal strands through its template strands (which can be used to regulate the behavior of neighboring micro-beads) or modulating its movement by deciding to self-assemble. For the latter, micro-beads aggregates become heavier as the number of micro-beads increases, therefore nullifying random motion. Similarly, assembling may be canceled by breaking the bond between micro-beads, thus reverting to random motion.

The setup also includes other materials in the environment, mostly enzymes and dNTPs (generic fuel for DNA production). A typical *in vitro* experiment takes several hours (e.g. 11 hours in [20] and [13]). Due to the limited length of the experiments, it is safely assumed that those chemicals can be considered constant in concentration over the time scale of the systems. This hypothesis is realistic thanks to the slow denaturation process of enzymes and the possibility to have a large excess of dNTPs.

Finally, monitoring of those systems relies on confocal microscopy to (1) track fluorescent molecules attached to template strands, which conditionally change light emission based on template activation and (2) see physical beads. By combining both sets of data, it is possible to monitor both the state of the controller (template activation) and of the robot (position).

2.2 Simulation

One major challenge when designing molecular programming systems (including PEN toolbox systems) is the non-linearity of the chemical reaction response. A trial and error approach is thus necessary, but unrealistic to perform directly *in vitro* due to the time required for such experiments. As such, a reasonable solution is to rely on simulation to find first good candidate solutions before actual implementation.

Dynamics of PEN toolbox systems have been modeled by Aubert *et al.* [6]. Their approach relies on a domain level model for reactions: complementary DNA strands are either completely double-stranded or single-stranded, with no intermediate form, and with enzymes performing their operations, in a single step. The set of possible reactions is thus strongly constrained and can then be enumerated.

Reactions are modeled through mass action kinetics, which yields a system of ordinary differential equations. Solving those equations then gives us the behavior of a system in a well–mixed environment. Reaction rates were obtained from experimental results when available [13] or from the existing literature otherwise.

We extend this model to spatial, inhomogeneous environment, by using a reaction-diffusion model. This is achieved by adding a diffusion term to the 0D reaction fluxes:

$$\frac{\partial[s]}{\partial t}(x,y,t) = r_s(x,y,t) + D_s \cdot \triangle_{[s]}(x,y,t) \quad (1)$$

where s is a DNA strand from the system, r_s the 0D flux for s, D_s the diffusion rate of s and $\triangle_{[s]}$ the local Laplacian of concentration. We can then solve the system on a discretized grid.

We further enrich the model with micro-beads, considering both motion and DNA functionalization.

Micro-beads move around following Brownian motion, which simply results from interactions at work in a liquid solution at a microscopic scale. This is implemented by applying a Gaussian noise to their position at each time step with mean 0 and variance $\sigma^2 = 2D$ with

$$D = \frac{k_B T}{6\pi d \eta_{\text{water}}} \quad (2)$$

where k_B is the Boltzmann constant, T the temperature, d the bead's diameter and η_{water} the viscosity of water. For aggregates of N micro-beads, we consider the structure has N times the size of a single micro-bead. Finally, we ignore collisions between micro-beads, as it can be assumed that micro-beads can move above or below each other.

With respect to functionalization, micro-beads are considered to have a homogeneous concentration of template strands on their surface, so that all the 2D surface covered by a micro-bead receive an equal concentration of templates. Additionally, we ignore fabrication defects and template leaks. We also consider that activation and inhibition work with the same rates for both single and aggregated micro-beads. While this is currently an open question, it is not unreasonable if aggregates are loose enough (i.e. if anchoring strands are long enough). Moreover, this assumption keeps the model simple.

2.3 Automated design of reaction networks

Search space. In our setup, the challenge is to design a reaction network that solve a given problem, i.e. to pick the right signal strands (produced at the surface of beads and diffusing in the environment) and template strands (attached to the beads) to trigger self-assembly at the right location. Moreover, we consider that all beads have the same templates at the same concentrations, which is akin to give the same program to all robots in a swarm.

The search space thus spans (1) a graph structure: the types of template strands on the beads, which will determine how signal strands are regulating each other's production; (2) the concentration of each type of template strands; (3) the duplex stability of each signal strand: the ratio of the hybridization rate over the separation rate of the strand, which is a proxy for the physical properties of a strand sequence.

Parameter	Value
Target number of bioNEAT species	20
Population size	50
Individual re-evaluations	2
Maximum graph size	16
Mutation operator	**Probability**
Parameter mutation	0.96
Add template strand	0.01
Remove template strand	0.01
Add signal species	0.01
Add inhibition species	0.01

Table 1: Parameters for bioNEAT.

BioNEAT. We rely on bioNEAT, an evolutionary algorithm based on the famous NEAT (NeuroEvolution of Augmenting Topologies) algorithm [7]. NEAT is designed to increase progressively the complexity of neural networks while relying on species to protect solution diversity. BioNEAT works similarly, with different mutation operators to deal with PEN toolbox graphs rather than neural networks. Global parameters for evolution are summarized in Table 1.

When performing a mutation, we select and apply one of the following mutation operators: mutating a parameter (signal species stability or template strand concentration); adding/removing a template strand; adding an activating signal species; adding an inhibition species. The last two operations differ from NEAT and are explained further.

Signal species are added in one of two ways, either (a) by splitting an activation template, so that the structure $A \rightarrow B$ becomes $A \rightarrow C \rightarrow B$ (which is similar to NEAT), or (b) by adding a signal species with an auto-catalytic template, as well as a template from that species to a random signal species in the graph.

Inhibition species are added either (a) by adding an inhibition species targeting an existing (non-inhibited) template and adding a template that bind an arbitrarily chosen existing signal species to the newly created species, or (b) by adding an inhibited template, i.e. performing both a template addition operation as defined above and applying (a) to that template.

Table 1 gives the probabilities for each variation to be applied. A detailed description of the algorithm can be found in [21, 22].

3 EXPERIMENTAL SETTINGS

3.1 Target objectives for self-assembly

We define four targets for self-assembly, named *bottom-line*, *right-line*, *top-line* and *center-line*. Fig. 5 (right) shows for each target a white area where micro-beads should self-assemble, starting from initial random positions. In order to enable localization, we place two fixed sources each producing a particular type of signal strand (see Fig. 5 (left)). Through diffusion, these two sources produce gradients throughout the environment, which can in turn be used by micro-beads to trigger self-assembly. Due to the isotropic diffusion of gradients, self-assembling into straight patterns is not straightforward.

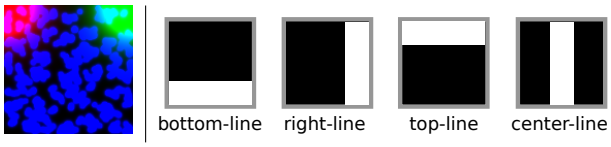

Figure 5: Left: the experimental setup, as seen from above. Two gradient sources are shown in red and green, respectively. Beads are shown in blue. Gradients diffuse isotropically throughout the environment. Right: targets for self-assembly, as seen from above. Micro-beads should aggregate in the white area.

Targets are meant to pose increasing challenges. The *bottom-line* target roughly corresponds to a non-linear repulsion from gradients. The *right-line* and *top-line* targets correspond to both non-linear attraction and repulsion from either a single or both gradient source(s). Finally, the *center-line* target requires a non-linear comparison of the concentration of gradient signals to establish two thresholds for self-assembly (positions of right *and* left edges).

Table 2 lists parameters and values for our experimental setup. In order to speed up evaluation time, we consider a small swarm of 500 micro-beads of 5μm in diameter within a chamber of $1mm * 1mm$, ie. preserving concentration within the same order of magnitude w.r.t. *in vitro* experiments.

3.2 Fitness function

In order to evaluate self-assembly with respect to target shapes, we first discretize the environment into a $N \times N$ matrix of cells ($N = 160$ in the following). Then we define the following fitness function, to compute a *match-nomatch* score:

$$f = \sum_{(x,y)\in target} r \times B(x,y) - \sum_{(x,y)\notin target} p * e^{a*d(x,y)} \times B(x,y) \quad (3)$$

where r is the reward, p the base penalty, $B(x,y)$ the indicator function of presence of a set of aggregated micro-beads at position (x,y), d the distance between a cell that is not within the target area and the closest cell within the target area, and a a scaling factor.

The first term of the equation provides a reward based on the number of cells *within* the target area that contains micro-beads producing anchoring strands. The second term is a penalization term that is based on the number of cells producing anchoring strands that are located *outside* the target area, with penalization increasing with distance to the target area.

In order to reduce the number of iterations for evaluating a candidate solution, we consider the production of anchoring strands (as a proxy for self-assembling) to estimate the position of aggregated micro-beads. In other words, $B(x,y)$ is equal to 1.0 if the concentration of anchoring strands in (x,y) is above a given threshold (ie. 5 nanoMolar), and 0 otherwise. Considering the production of anchoring strands makes it possible to avoid waiting for micro-beads to actually self-assemble, which can be quite slow with Brownian motion.

In addition to the fitness score, we provide a qualitative estimate by setting a threshold for discriminating between successful and failed attempts. This criterion for successful self-assembly is defined by considering solutions covering more than 50% of the target area, with more signal expressed in the target area than outside.

Simulation parameters

Arena size	1mm × 1mm
Beads	500
Bead size (aggregation)	50 μm
Bead size (Brownian motion)	5 μm
Temperature	43°C
Grid size	160 × 160
Time discretization	0.1 min per step
Duration of an evaluation	1000 steps (i.e. 100 min)

Fitness parameters

r (reward)	1.0
p (penalty)	0.1
a (scaling)	0.1

Table 2: Parameters for simulation and fitness evaluation.

4 RESULTS

4.1 Optimization

For each target, we perform 12 replicates of 2500 evaluations for the right-line, top-line and bottom-line targets and 7500 for the more challenging center-line target. To reduce evaluation noise, each evaluation comprises two independent simulations, with the final fitness of the individual being the worse of the two. Each replicate takes less than a day of computation time using 24-core computer running Ubuntu 14.04.

Fig. 6 shows the results for each of the target. Each graph uses boxplots to display the best fitnesses from each of the 12 replicates throughout evolution. It can be observed that performance of the best individual increases over time, with the supposedly easier targets displaying higher fitness values. However, minimal fitness values are always very close to zero (except for the right-line target), implying that some runs struggle to bootstrap evolution.

Fig. 7 shows the evaluation of the best solutions of the last generation of each replicate. For clarity, only the anchoring signal is shown, that is: the position were micro-beads are self-assembling into larger clusters. In this figure, runs that produce successful self-assembly (cf. Section 3.2) are highlighted in green. Successful self-assembly is as follows: bottom-line: 6/12, right-line: 5/12, top-line: 4/12, and center-line: 2/12 (with two additional candidate solutions failing very short from the threshold). As expected, the more difficult the target, the lower the success rate.

Network complexity increases as target are more challenging, with the best solutions for each setup featuring 4 (bottom-line), 6 (right-line), 8 (top-line) and 15 nodes (center-line). It should also be noted that in all but the most simple case, evolved candidate solutions are not symmetric, even when the target is, and sometimes feature nodes that could easily be removed (see next sub-Section). we can also note some counter-intuitive or non-trivial features such as using incoherent feed-forward[3] instead of inhibition. While the simplest networks may be interpreted by an expert, candidate solutions for the center-line target are more difficult to analyze as

[3]I.e. a signal species both activates and inhibits the production of an output.

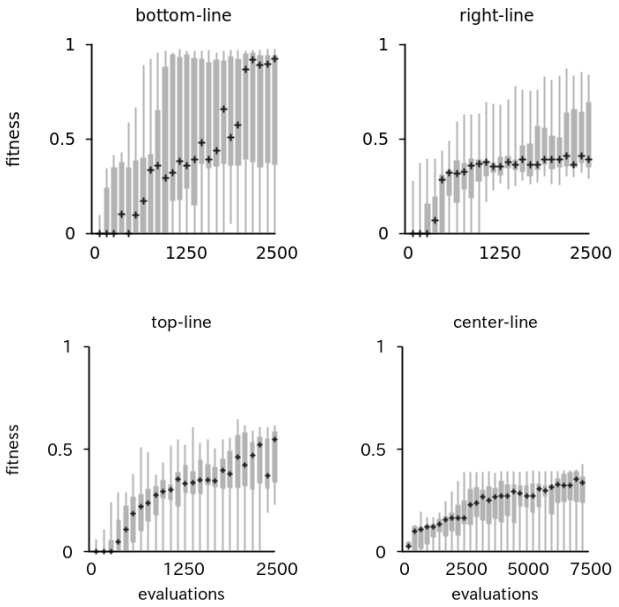

Figure 6: Performance results for each target. Each graph displays results obtained by picking the best individuals of each generation of the $N = 12$ independent runs. A value of 1.0 represents the maximum fitness value.

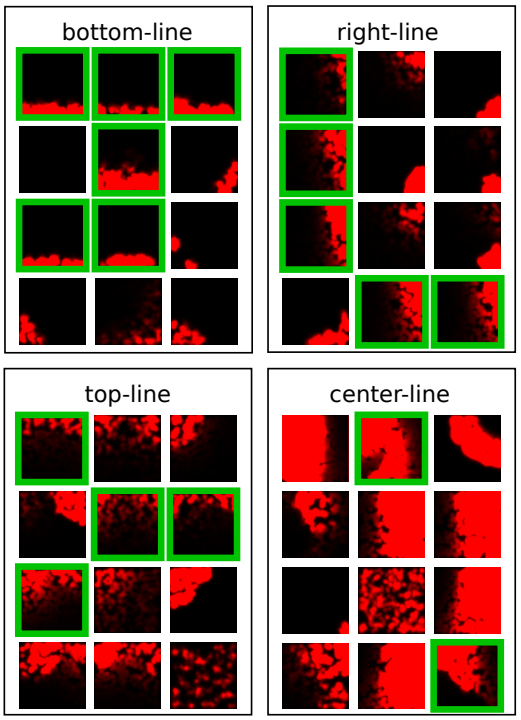

Figure 7: Best solutions of the last generation of each run for each target. Red indicates production of the self-assembly signal. Green borders highlight successful solutions (see Section 3.2).

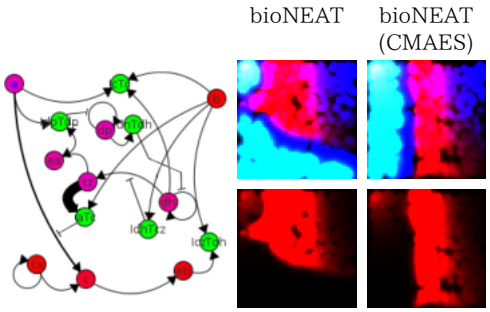

Figure 8: Best solution obtained by evolution, with additional parameter tuning done by CMA-ES. Left: chemical reaction network; top-right: chemical concentrations; bottom-right: production of anchoring signals.

they involve complex interactions between aggregated and non-aggregated beads, which are observed from the multiple signals produced throughout the environment.

4.2 Automation vs. expert knowledge

Prior to running our evolutionary algorithm, a control experiment was conducted. A human expert[4] in PEN toolbox had to design from scratch a chemical reaction network for the center-line target. While this was originally motivated by addressing the question of whether the center-line could be achieved or not, this ended up providing a reference value to evaluate evolved candidate solutions.

Attempts were done over a full week, with hundreds of trial and errors, each trial requiring simulation (from a few seconds to several minutes). In addition, we used the state-of-the-art CMA-ES algorithm [8] to automatically tune parameters over 4000 evaluations starting from the chemical reaction network obtained from the expert. Similarly, we used CMA-ES to tune the parameters of the solution previously obtained by bioNEAT for the center-line target. As a result, we get four different methods each defined by a specific design method, and respectively termed bioNEAT (see previous sub-Section), expert, bioNEAT+CMAES and expert+CMAES. Results are shown in Fig. 9 and 8.

The bioNEAT- and expert-designed solutions yield similar results, but with a statistically significant advantage for the latter (0.41 ± 0.09 vs. 0.47 ± 0.10 over 100 evaluations, and $p \ll 0.05$ with Mann-Whitney U test). Then, both benefits from additional parameter-tuning using CMAES, with bioNEAT+CMAES jumping to a fitness value of 0.81 ± 0.05 and expert+CMAES topping at 0.94 ± 0.02, this time leading with a statistically significant advantage ($p \ll 0.05$).

The design from the expert can be interpreted rather straightforwardly: the system performs a comparison of gradient signal concentrations; if the two concentration aren't close enough, production of the anchoring signal is inhibited. As this design partly fails to capture the influence of the distance to the gradient sources, the production of self-assembly signals widen as microbeads are farther away from the sources.

By comparison, the evolved system gains in compactness as unnecessary species can be easily spotted and trimmed, with only

[4]NHK, the first author of this paper, who acquired both extensive *in silico* and *in vitro* experience with the PEN toolbox during the past 6 years.

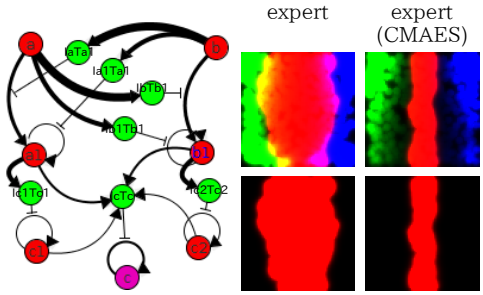

Figure 9: Best solution designed by the expert (both structure and parameter values). Left: chemical reaction network; top-right: chemical concentrations; bottom-right: production of anchoring signals.

10 species remaining without changing the dynamics (not shown here), versus 14 for the expert design, which we failed to compress further. It is indeed challenging for any human expert to deal with (1) non-linear behaviors due to chemical saturations, (2) structures with (many) feedback loops and (3) hidden interactions, such as the influence of the distance between micro-beads on local template concentrations. As for the evolved solutions, trading understandability for compactness is actually a sound option as smaller networks are less likely to give way to undesired interactions when tested *in vitro*.

4.3 Self-repairing from induced damages

A desired properties of self-assembled robotic system is the ability to recover from damage or, in other words, to *self-repair*. Given that control is distributed over the micro-beads, we may expect a similar feature in our system.

The protocol is as follows: the system runs during 1000 (resp. 4000) timesteps for the bottom-line, right-line and top-line target setups (resp. center-line setup), which corresponds to the standard development time used previously. Then, damage is induced by removing all micro-beads within an area corresponding to 1/5th of the surface at the center of the optimal target. This is followed by a recovery time of 30000 timesteps (i.e. approx. 3000 minutes) after which we stop the simulation. Results are shown in Table 3, compiled from 10 trials per candidate solution. In addition, Fig. 10 shows typical examples of the self-repairing process for the center-line target.

From target to target, the efficiency of the self-repair processes vary greatly, from efficient to detrimental when compared to the damaged starting point, and never succeed to match the original scores before damage. Zooming on the center-line target, the expert-designed solution yield catastrophic self-repairing ($p \ll 0.05$, Wilcoxon signed-rank test), while the bioNEAT-designed solution is able to partly recover from damage ($p \ll 0.05$). This is quite different when additional parameter tuning is involved: both the expert+CMAES and the bioNEAT+CMAES versions show recovery from damage ($p \ll 0.05$ in both cases).

Looking closer at the candidate solutions, it is possible to classify solutions as featuring stable or non-stable anchoring strands, which have an impact on the ability to recover from damage. On the one hand, stable strands display very little self-repair capability but also do not deform much over time. On the other hand, unstable strands continuously switch from self-assembled to solitary states, and this may help to perform self-repair (by spreading over the damage area), but are subject to variation over time even without introducing damage (ie. deformation). Unstable strands are due to low thermodynamical stability of their duplexed configuration (i.e. the binding between strands). This is illustrated by the example shown for the center-line target obtained with bioNEAT only: the final structure displays a similar fitness after self-repair, but its global shape is not similar as self-assembling continue in the upper part. A similar balance between stability and self-repair has been

target	before	damaged	recovery
bottom-line	0.92±0.02	0.71±0.02	0.78±0.05
right-line	0.85±0.02	0.64±0.02	0.70±0.04
top-line	0.53±0.06	0.43±0.05	0.40±0.06
center-line:			
bioNEAT	0.41±0.09	0.14±0.09	0.18±0.06
expert	0.47±0.10	0.28±0.10	0.10±0.10
bioNEAT+CMAES	0.81±0.05	0.53±0.05	0.63±0.09
expert+CMAES	0.94±0.02	0.75±0.01	0.89±0.03

Table 3: Results for self-repair. For each targets, the best candidate solution is evaluation $N = 100$ times, by evaluating its fitness value before damage, after damage, and after recovery.

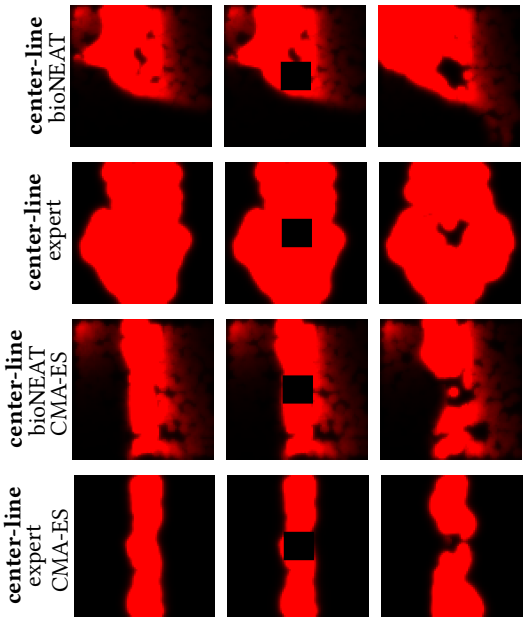

Figure 10: Typical examples of self-repairing systems for the center-line target (each line correspond to a particular method). Left: self-assembled swarm of micro-beads before damage; Center: just after damage; Right: after recovery (30000 time steps later).

analyzed in self-healing polymeric materials (e.g. see the review in [23]).

It remains an open question as to why some candidate solutions feature unstable anchoring strands, or stable ones. Also, it is possible that enforcing self-assembly to come to a halt may help with both stability and self-repair, as it was previously noted for other developmental systems [24].

5 DISCUSSION AND CONCLUSION

In this paper, we started with describing the first *in vitro* demonstration of self-assembly using a swarm of micro-robots composed of one million DNA-functionalized micro-beads. Then, we described and validated (using simulation) a method for the automated design of self-assembling behaviors for a microscopic robot swarm that combines a realistic simulator and either the bioNEAT algorithm (to optimize the topology of chemical reaction networks) or CMA-ES (to optimize parameters), or both at once. We showed that bioNEAT is able to find reaction networks close in performance to expert-designed solutions, but in shorter time and with better compacity. We also showed that CMA-ES very significantly improves both bioNEAT- and expert-designed reaction networks. Finally, we showed promising results w.r.t self-repairing capabilities.

We are left with the question of how realistic the chemistry simulated in this work is. We simulate micro-beads of $5\mu m$ in diameter, which is in-between large micro-beads capable of self-regulation but which are not subject to Brownian motion due to their size [17], and smaller micro-beads which may move but not self-regulate. One possible solution would be to use large micro-beads ($> 10\mu m$) and emulate Brownian motion with vibration motors attached to the chamber. While this approach remains technically challenging, it is far from unrealistic. However, the question remains open as to the interaction between induced-vibration and localization gradients. Another solution is to use small micro-beads ($1\mu m$) and to make do with the lack of self-regulation, which in our case may not be a problem as long as self-regulation is not used.

Another open question is whether self-assembling kills the production of signal. Clusters of beads may be too tight for enzymes to properly interact with templates, meaning that aggregates would be functionally limited. One possible solution is to use porous spheres to maintain diffusion throughout aggregates of beads, but this remains to be tried *in vitro*. That said, it is also not clear that such a limitation would be an issue as the production of anchoring signal is necessary at the *frontiers* of aggregates, not within.

ACKNOWLEDGMENTS

This project was supported by the ELSI Origins Network (EON), which is supported by a grant from the John Templeton Foundation. The opinions expressed in this publication are those of the authors and do not necessarily reflect the views of the John Templeton Foundation. This work was also financially supported by the "PHC Sakura" program (project number: 34171WG), implemented by the French Ministry of Foreign Affairs, the French Ministry of higher education and research and the JSPS. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of a Titan X GPU used in a part of this research.

ANNEX: IN VITRO EXPERIMENT (METHODS)

1μm beads prepared with 100 nM aggregation strands were incubated at 43 °C in glass capillaries in a buffer of 20 mM Tris-HCl, 10 mM $(NH_4)_2SO_4$, 10 mM KCl, 50 mM NaCl, 8 mM $MgSO_4$, 10 μM each dNTP, 1 μM dUTP, 1% DTT, 1x EvaGreen intercalating dye, 2.5% BSA, 4% Nb BSMI nicking enzyme, 0.03% Bst Warmstart polymerase, 0.0085% ttRecJ exonuclease, 100 nM streptavidin, 0.4% SSB and 30 nM template strand producing the self-assembly strand. After the glass capillaries were filled, the system was triggered either with water (control) or signal strand activating the template. Additional details on preparing PEN toolbox systems can be found in [25]. The system was observed under a confocal microscope, looking both at visible light and green fluorescence (from Evagreen dye).

REFERENCES

[1] M Rubenstein, A Cornejo, and R Nagpal. Programmable self-assembly in a thousand-robot swarm. *Science*, 345(6198):795–799, 2014.
[2] R O'Grady, AL Christensen, and M Dorigo. Swarmorph: multirobot morphogenesis using directional self-assembly. *IEEE Transactions on Robotics*, 25(3):738–743, 2009.
[3] E Tuci, R Groß, V Trianni, F Mondada, M Bonani, and M Dorigo. Cooperation through self-assembly in multi-robot systems. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 1(2):115–150, 2006.
[4] R Groß, M Bonani, F Mondada, and M Dorigo. Autonomous self-assembly in swarm-bots. *IEEE transactions on robotics*, 22(6):1115–1130, 2006.
[5] R Groß and M Dorigo. Self-assembly at the macroscopic scale. *Proceedings of the IEEE*, 96(9):1490–1508, 2008.
[6] N Aubert, C Mosca, T Fujii, M Hagiya, and Y Rondelez. Computer-assisted design for scaling up systems based on dna reaction networks. *Journal of The Royal Society Interface*, 11(93):20131167, 2014.
[7] K O Stanley and R Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127, 2002.
[8] N Hansen. The CMA evolution strategy: a comparing review. In *Towards a new evolutionary computation*, pages 75–102. Springer, 2006.
[9] L M Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266(5187):1021–1024, 1994.
[10] L Qian and E Winfree. Scaling up digital circuit computation with DNA strand displacement cascades. *Science*, 332(6034):1196–1201, 2011.
[11] L Qian, E Winfree, and J Bruck. Neural network computation with DNA strand displacement cascades. *Nature*, 475(7356):368–372, 2011.
[12] N C Seeman. Dna in a material world. *Nature*, 421(6921):427–431, 2003.
[13] A Padirac, T Fujii, and Y Rondelez. Bottom-up construction of in vitro switchable memories. *Proceedings of the National Academy of Sciences*, 109(47):E3212–E3220, 2012.
[14] A Padirac, T Fujii, A Estevez-Torres, and Y Rondelez. Spatial waves in synthetic biochemical networks. *Journal of the American Chemical Society*, 135(39):14586–14592, 2013.
[15] K Montagne, R Plasson, Y Sakai, T Fujii, and Y Rondelez. Programming an in vitro dna oscillator using a molecular networking strategy. *Molecular systems biology*, 7(1):466, 2011.
[16] M Hagiya, N Aubert-Kato, S Wang, and S Kobayashi. Molecular computers for molecular robots as hybrid systems. *Theoretical Computer Science*, 632:4–20, 2016.
[17] G Gines, A Zadorin, JC Galas, Teruo Fujii, A Estevez-Torres, and Y Rondelez. Microscopic agents programmed by dna circuits. *Nature Nanotechnology*, advance online publication, 2017.
[18] W Benjamin Rogers and Vinothan N Manoharan. Programming colloidal phase transitions with dna strand displacement. *Science*, 347(6222):639–642, 2015.
[19] M Hadorn, E Boenzli, K T Sørensen, H Fellermann, P E Hotz, and M M Hanczyc. Specific and reversible dna-directed self-assembly of oil-in-water emulsion droplets. *Proceedings of the National Academy of Sciences*, 109(50):20320–20325, 2012.
[20] A J Genot, A Baccouche, R Sieskind, N Aubert-Kato, N Bredeche, JF Bartolo, V Taly, T Fujii, and Y Rondelez. High-resolution mapping of bifurcations in nonlinear biochemical circuits. *Nature Chemistry*, pages 760–767, 2016.
[21] N Aubert, H Q Dinh, M Hagiya, H Iba, T Fujii, N Bredeche, and Y Rondelez. Evolving cheating dna networks: a case study with the rock-paper-scissors game. In *Advances in Artificial Life, ECAL*, volume 12, pages 1143–1150, 2013.
[22] H Q Dinh, N Aubert, N Noman, T Fujii, Y Rondelez, and H Iba. An effective method for evolving reaction networks in synthetic biochemical systems. *Evolutionary Computation, IEEE Transactions on*, 19(3):374–386, 2014.
[23] R P Wool. Self-healing materials: a review. *Soft Matter*, 4(3):400–418, 2008.
[24] A Devert, N Bredeche, and M Schoenauer. Robustness and the halting problem for multicellular artificial ontogeny. *IEEE Transactions on Evolutionary Computation*, 15(3):387–404, 2011.
[25] A Baccouche, K Montagne, A Padirac, T Fujii, and Y Rondelez. Dynamic dna-toolbox reaction circuits: a walkthrough. *Methods*, 67(2):234–249, 2014.