**GDANSK UNIVERSITY OF TECHNOLOGY**
**Faculty of Electronics,**
**Telecommunications and Informatics**

**Michał Joachimczak**

# Evolution of gene regulatory networks and artificial embryogenesis in a simulated 3D environment

PhD Dissertation

Supervisors:
**dr hab. Borys Wróbel**
Systems Modelling Laboratory,
Institute of Oceanology, Polish Academy of Sciences, Sopot
Evolutionary Systems Laboratory,
Adam Mickiewicz University, Poznań
Institute of Neuroinformatics, University of Zürich and ETHZ, Zürich

**dr hab. inż. Wojciech Jędruch**
Faculty of Navigation and Naval Weapons,
Polish Naval Academy, Gdynia
Faculty of Electronics,
Telecommunications and Informatics,
Gdańsk University of Technology

**Gdańsk, 2012**

# Abstract

Development of a multicellular organism starts from a single, undifferentiated cell and progresses through subsequent cell divisions. The behaviour of each cell is a result of interactions between products of the genes encoded in the genome, signals from the environment (including signals sent by other cells) and the laws of physics. Each cell contains a copy of the same genome and thus its behaviour is determined by the same controller, which can be described as a network of interactions, known as a gene regulatory network. Even though they share the controller, cells can differentiate and take different roles during the development. Most importantly, nowhere in the genome specific information about the role of each cell is stored, and genomes with information content of hundreds millions bits encode the plan for building organisms with trillions of cells.

The work presented in this thesis belongs to the field of artificial embryogenesis, a subfield of Artificial Life, concerned with the investigation of the properties of biological development and capturing its properties *in silico*, both with the goal of its better understanding as well as for its potential engineering applications. The limitations of direct genotype-phenotype mappings in evolutionary computation mean that only relatively simple designs can be created. Simulated developmental process is looked upon as one of the solutions to the problem of efficiently encoding designs in a manner that would be highly evolvable and would scale to large structures, while at the same time display properties such as failure tolerance or even ability of self repair.

The objective of this work is twofold. One is to investigate properties and evolvability of biologically inspired multicellular development and artificial gene regulatory network in the context of their potential applications such as automatically designed controllers and self assembling 3D structures. The other is to create a model that can be used to investigate how genomes and morphologies evolve and that allows to perform simulation experiments relevant to evolutionary developmental biology.

The thesis introduces a new model of biologically inspired multicellular development that is controlled by a gene regulatory network and takes place in a 3D environment, with cells interacting through simulated physical forces. The ability to evolve desired shapes and cell differentiation is investigated. Properties of the evolution of development itself and properties of the evolved virtual embryos are examined. Features unselected

for, such as robustness to damage, are observed to emerge and are further analysed.

An open ended system for the evolution of morphologies without an objective fitness function is further introduced. In such a system, there is a continuous evolutionary pressure to generate novel morphologies. The evolution of morphologies and gene regulatory networks over time is investigated and discussed.

The model of artificial gene regulatory network proposed in this thesis is also analysed for its evolvability and applicability to control problems using a range of signal processing tasks and on a problem of controlling the behaviour of simulated animats.

## Acknowledgements

I would like express my gratitude to my thesis advisers prof. Borys Wróbel and prof. Wojciech Jędruch for their interest, support and numerous discussions which made this thesis possible. I would also like to thank prof. Ksenia Pazdro, prof. Tymon Zieliński and prof. Ewa Kulczykowska for the support I received from them during the PhD study. I wish to thank my colleagues from the Laboratory of Genetics and Marine Biotechnology for showing interest and being open minded about this interdisciplinary work and for the encouraging atmosphere in the lab. Many thanks to my friends at the institute for the good time spent together and for your support. Finally, I would like to thank my family for always encouraging my education and inspiring my interest in science.

This dissertation was supported by the Polish Ministry of Science and Education (project N519 384236). The computational resources used in this work were obtained thanks also to the support of the project N303 291234, the Tri-city Academic Computer Centre (TASK) and the Interdisciplinary Centre for Molecular and Mathematical Modelling (ICM, University of Warsaw; project G33-8).

# Contents

# List of Figures

# List of Tables

# The main thesis of this dissertation

1. The proposed model of artificial gene regulatory network coupled with a biologically inspired model of the genome constitutes a highly evolvable system that can be used as an alternative to evolving dynamic neural networks.
2. Biologically inspired artificial gene regulatory networks can be evolved to control multicellular development in three dimensions with simulated physics, allowing to obtain structures that self organize and display biological properties such as robustness to damage.
3. The proposed model of gene regulatory network and development can be used to investigate the evolution of morphology of living organisms by performing biologically relevant simulation experiments that would be difficult or impossible to perform using other approaches.

# Abbreviations

**alife** artificial life

**DNA** deoxyribonucleic acid

**evo-devo** evolutionary developmental biology

**GA** genetic algorithm

**GRN** gene regulatory network

**L-System** Lindenmayer system

**mRNA** messenger RNA

**RBN** random boolean network

**RNA** ribonucleic acid

**RNAP** RNA polymerase

**SMP** symmetric multiprocessing

**TBP** TATA binding protein

**TF** transcription factor

**tRNA** transfer RNA

# Publications

The main contributions of the presented thesis were the subject of the following peer reviewed papers.

1. Joachimczak, M. and Wróbel, B. (2012a). Evolution of robustness to damage in artificial 3-dimensional development. *Biosystems*. (in press)

2. Joachimczak, M. and Wróbel, B. (2009). Complexity of the search space in a model of artificial evolution of gene regulatory networks controlling 3D multicellular morphogenesis. *Advances in Complex Systems*, 12(3):347–369

3. Joachimczak, M. and Wróbel, B. (2012b). Open ended evolution of 3D multicellular development controlled by gene regulatory networks. In *Artificial Life XIII: Proceedings of the 13th International Conference on the Simulation and Synthesis of Living Systems*, Cambridge, MA. MIT Press. (in press)

4. Joachimczak, M. and Wróbel, B. (2011a). Evolution of the morphology and patterning of artificial embryos: Scaling the tricolour problem to the third dimension. In Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F., Mitchell, J. C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M. Y., Weikum, G., Kampis, G., Karsai, I., and Szathmáry, E., editors, *Advances in Artificial Life. Darwin Meets von Neumann: Proceedings of the 10th European Conference on Artificial Life (ECAL 2009)*, volume 5777 of *Lecture Notes in Computer Science*, pages 35–43, Berlin - Heidelberg. Springer

5. Joachimczak, M. and Wróbel, B. (2010a). Evolving gene regulatory networks for real time control of foraging behaviours. In Fellermann, H., Dörr, M., Hanczyc, M. M., Laursen, L. L., Maurer, S., Merkle, D., Monnard, P.-A., Stoy, K., and Rasmussen, S., editors, *Artificial Life XII: Proceedings of the 12th International Conference on the Simulation and Synthesis of Living Systems*, pages 348–355, Cambridge, MA. MIT Press

6. Joachimczak, M. and Wróbel, B. (2010b). Processing signals with evolving artificial gene regulatory networks. In Fellermann, H., Dörr, M., Hanczyc, M. M., Laursen, L. L., Maurer, S., Merkle, D., Monnard, P.-A., Stoy, K., and Rasmussen, S., editors, *Artificial Life XII: Proceedings of the 12th International Conference on the Simulation and Synthesis of Living Systems*, pages 203–210, Cambridge, MA. MIT Press

7. Joachimczak, M. and Wróbel, B. (2008a). Evo-devo *in silico*: a model of a gene network regulating multicellular development in 3D space with artificial physics. In Bullock, S., Noble, J., Watson, R., and Bedau, M. A., editors, *Artificial Life XI: Proceedings of the 11th International Conference on the Simulation and Synthesis of Living Systems*, pages 297–304, Cambridge, MA. MIT Press

8. Joachimczak, M. and Wróbel, B. (2008b). Evolution of 3D development controlled by a gene regulatory network: The complexity of the search space and evolvability. In Klemm, K., Merkle, D., and Olbrich, E., editors, *8th German Workshop on Artificial Life: Proceedings of the GWAL-8, Leipzig, Germany*, pages 11–22, US. IOS Press

9. Joachimczak, M. and Wróbel, B. (2011b). Ewolucja sieci genowych kontrolujących wirtualne organizmy jedno- oraz wielokomórkowe. In Obolewicz, P., Kujawa, K., and Sacharuk, P., editors, *ICT Young 1*, Zeszyty Naukowe Wydziału ETI Politechniki Gdańskiej, pages 179–184. (in Polish)

The models of the gene regulatory network and of the multicellular developmental introduced in this thesis were also used as the basis for the following peer reviewed papers.

1. Joachimczak, M. and Wróbel, B. (2012). Co-evolution of morphology and control of soft-bodied multicellular animats. In *GECCO '12: Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*. ACM. (in press)

2. Joachimczak, M., Kowaliw, T., Doursat, R., and Wróbel, B. (2012). Brainless bodies: Controlling the development and behavior of multicellular animats by gene regulation and diffusive signals. In *Artificial Life XIII: Proceedings of the 13th International Conference on the Simulation and Synthesis of Living Systems*, Cambridge, MA. MIT Press. (in press)

3. Erdei, J., Joachimczak, M., and Wróbel, B. (2011). Ewolucja chemotaksji organizmów jednokomórkowych w dwuwymiarowym środowisku. In Obolewicz, P., Kujawa, K., and Sacharuk, P., editors, *ICT Young 1*, Zeszyty Naukowe Wydziału ETI Politechniki Gdańskiej, pages 173–178. (in Polish)

4. Wróbel, B., Joachimczak, M., Montebelli, A., and Lowe, R. (2012b). The search for beauty: Evolution of minimal cognition in an animat controlled by a gene regulatory network and powered by a metabolic system. In *Proceedings of the 12th International Conference on Simulation of Adaptive Behaviour, From Animals to Animats 12 (SAB'12)*, Lecture Notes in Artificial Intelligence. Springer-Verlag. (in press)

# Extended abstract in Polish / Rozszerzone streszczenie w języku polskim

Poniżej zamieszczone zostało rozszerzone streszczenie rozprawy doktorskiej "Ewolucja sieci genowych oraz embriogenezy w symulowanym, trójwymiarowym środowisku" z podziałem na poszczególne rozdziały.

## 1. Wprowadzenie

Znane są tylko dwa procesy prowadzące do powstawania struktur zaprojektowanych tak, by pełniły konkretne funkcje. Pierwszym jest ludzka kreatywność. Stojąc przed problemem inżynieryjnym, jesteśmy w stanie zaproponować potencjalne rozwiązania, symulować je używając naszej wyobraźni, po czym ostatecznie je urzeczywistnić. Bardzo złożone problemy możemy rozwiązywać rozbijając je na mniejsze i skupiając uwagę na odpowiednim w danym momencie poziomie abstrakcji. Co ważne, dzięki modularnemu podejściu do projektowania możemy wielokrotnie wykorzystywać wcześniejsze rozwiązania, bez potrzeby pełnego rozumienia każdego szczegółu. Jako że olbrzymia część naszej działalności jako gatunku związana jest z tworzeniem i projektowaniem, być może nie powinno dziwić, że przez niemal całą historię istnienia cywilizacji, zawsze gdy byliśmy zafascynowani złożonością świata naturalnego, zakładaliśmy, że musi stać za nią projektant i do tego taki, który jest od nas o wiele bardziej uzdolniony. Minęło raptem 150 lat od momentu w którym, dzięki pracom Darwina oraz Wallace'a, zrozumieliśmy, że istnieje drugi, całkowicie naturalny proces, który prowadzi do powstawania struktur, które wyglądają na zaprojektowane. Złożoność projektów powstałych na drodze tego procesu nie tylko przewyższa złożonością struktury, które do tej pory zaprojektował ludzki umysł - on sam jest produktem tego właśnie procesu. Tym procesem jest ewolucja.

Chociaż sama idea mechanizmu ewolucji jest niezwykle piękna w swojej prostocie, nasze rozumienie jej mechanizmów oraz implikacji przebyło długą drogę od 1858 roku, kiedy została zaproponowana. Z jednej strony bardzo dobrze rozumiemy teraz jak dziedziczna informacja zakodowana jest w kwasach nukleinowych oraz poznaliśmy z dużą dokładnością historię życia na Ziemi. Z drugiej strony, postrzegamy

teraz ewolucję jako proces niezależny od substratu: uniwersalny mechanizm, który będzie zachodził zawsze, gdy spełnione zostaną określone warunki, czy to na innej planecie, czy też w obrębie symulacji komputerowej. Te warunki to obecność replikatorów, obiektów które mogą reprodukować się z okazjonalnymi modyfikacjami, oraz środowisko, które ogranicza dostępne im zasoby. Zawsze, gdy te warunki zostaną spełnione, replikatory, które niosą w sobie korzystne zmiany (mutacje), zaczynają w kolejnych generacjach zdobywać przewagę nad ich niezmodyfikowanymi przodkami. Jednak to, w jakim zakresie ewolucja będzie prowadziła do wzrostu złożoności, zależy od wielu czynników takich jak sposób kodowania dziedziczonych informacji. Wpływa to na tzw. ewoluowalność, którą rozumie się jako zdolność do generowania dziedzicznej różnorodności oraz pozyskiwania nowych funkcji, które poprawiają dostosowanie (Wagner, 2005). Kwestia tego jak można tworzyć sztuczne, ewoluowalne systemy jest jednym z centralnych zagadnień tej pracy.

Sztuczne Życie (ang. *Artificial Life*) jest interdyscyplinarnym kierunkiem badań, który rozwinął się połowie lat osiemdziesiątych XX wieku wraz z rosnącym zainteresowaniem komputerowym modelowaniem ewolucji i algorytmami ewolucyjnymi. Głównym przedmiotem badań Sztucznego Życia jest proces ewolucji oraz zjawiska emergentne, takie jak samoorganizacja w systemach złożonych z wielu jednostek w oparciu o lokalne reguły, niezależnie czy jednostkami są cząsteczki, komórki czy organizmy. Wprowadzenie pracy zawiera przykłady najbardziej znanych środowisk i modeli reprezentatywnych dla dziedziny Sztucznego Życia, takich jak Tierra, Avida, Framsticks czy pływające organizmy Karla Simsa.

Badania przedstawione w obrębie rozprawy należą do tzw. Sztucznej Embriogenezy (ang. *artificial embryogenesis*), poddziedziny sztucznego życia zajmującej się badaniem i modelowaniem procesu rozwojowego organizmów wielokomórkowych. Dzięki procesowi rozwojowemu organizm złożony z bilionów komórek (taki jak np. nasze ciało) powstaje na drodze kolejnych podziałów z pojedynczej komórki, wykorzystując informację genetyczną, której zawartość informacyjna wyraża się "jedynie" w setkach megabajtów. Proces ten koduje więc finalną strukturę organizmu w sposób niebezpośredni i wykazuje się przy tym niezwykłą ewoluowalnością, skalowalnością oraz odpornością na zaburzenia zewnętrzne. Nie bez powodu więc proces rozwojowy uważany za jeden kluczowych mechanizmów, który pozwala na pokonanie ograniczeń skalowalności wynikających z metod kodowania bezpośredniego stosowanych w algorytmach ewolucyjnych (Tufte, 2008).

Przedstawiona praca ma dwa główne cele. Pierwszym jest opracowanie ewoluowalnego modelu procesu rozwojowego, kontrolowanego w sposób biologicznie realistyczny, za pomocą sieci genowej zakodowanej niebezpośrednio w genomie, który pozwala na ewolucję zadanych, trójwymiarowych morfologii. W obrębie pracy badana jest również ewoluowalność samej sieci genowej i możliwość zastosowania jej do problemów innych niż kontrola procesu rozwojowego. Drugim celem pracy było stworzenie biologicznie realistycznego modelu ewolucji procesu wielokomórkowego, który pozwala na badanie tego, jak genomy, sieci genowe oraz morfologie wielokomórkowych organizmów ewoluują w czasie. Ten aspekt pracy wpisuje się w nurt badań należących do ewolucyjnej biologii rozwoju (nazywanej skrótowo *evo-devo*, od

ang. *evolutionary developmental biology*). Modele komputerowe, chociaż posiadają liczne ograniczenia, pozwalają na przeprowadzanie eksperymentów symulacyjnych, które w przypadku eksperymentów biologicznych byłyby trudne lub wręcz niemożliwe do wykonania.

#### Tezy pracy

1. Proponowany model sztucznej sieci genowej w połączeniu z biologicznie inspirowanym modelem genomu stanowi wysoce ewoluowalny system, który może być używany jako alternatywa dla ewolucji dynamicznych sieci neuronowych.

2. Biologicznie inspirowany model sztucznej sieci genowej może być zastosowany do ewolucji wielokomórkowej embriogenezy zachodzącej w 3 wymiarach oraz wykorzystującej symulowaną fizykę, pozwalając na otrzymywanie struktur, które są zdolne do samoorganizacji oraz wykazują biologiczne własności, takie jak odporność na uszkodzenia.

3. Proponowany model sieci genowej oraz procesu rozwojowego może być wykorzystywany do badania ewolucji morfologii organizmów żywych, pozwalając na przeprowadzanie eksperymentów symulacyjnych, które byłyby trudne lub niemożliwe do wykonania przy zastosowaniu innych metod.

### Wprowadzenie biologiczne

W obrębie pierwszego rozdziału przybliżone zostały podstawowe pojęcia oraz mechanizmy związane ze sposobem, w jakim organizmy żywe przechowują informację genetyczną oraz jak ta informacja wpływa na zachowanie się komórek. Przedstawiona została krótka historia odkryć, które doprowadziły do poznania natury kwasów nukleinowych. Opisany zostały proces przepisywania informacji zawartej w cząsteczce DNA na cząsteczkę mRNA (transkrypcja) oraz proces syntezy białka w oparciu o cząsteczkę mRNA (translacja). Wprowadzona została koncepcja sieci genowej jako opisu interakcji pomiędzy wzajemnie wpływającymi na swoją regulację genami oraz opisane zostały mechanizmy interakcji czynników transkrypcyjnych z DNA oraz ich wpływ na regulację ekspresji genów. Ostatnia część rozdziału 1 wprowadza podstawowe zagadnienia związane z procesem rozwojowym organizmów wielokomórkowych (embriogenezą). Proces rozwojowy został przedyskutowany w kontekście obserwowanych podobieństw (homologii) w strukturze budowy ciała organizmów, pomimo pozornych dużych różnic zewnętrznych. Organizmy wizualnie tak odmienne jak np. nietoperz czy wieloryb w istocie posiadają ten sam plan budowy ciała. Jest to szczególnie widoczne na etapie rozwoju embrionalnego i było już dostrzegane w czasach Darwina, stanowiąc silny argument za koncepcją pochodzenia organizmów od wspólnego przodka. Co więcej, morfologie organizmów wydają się cechować wysoką modularnością: struktury takie jak np. palce są wielokrotnie powtórzone i jednocześnie same składają się z serii mniejszych, powtórzonych kości. Organizmy mogą posiadać kończyny tak odmienne jak skrzydła i płetwy, ale

w istocie kończyny te mają identyczną strukturę, różniąc się liczbami powtórzeń elementów i ich wymiarami. Sugeruje to, że ewolucja w ogromnym stopniu operuje poprzez wykorzystanie pewnych ustalonych schematów budowy, generując różnorodność na drodze modyfikacji ich dosyć wysokopoziomowych parametrów, takich jak liczba powtórzeń czy wymiary poszczególnych elementów morfologii. Mutacje prowadzące do zmiany liczby palców kończyny mogą zaś wiązać się z bardzo niewielkimi zmianami w genomie.

Postęp w naszym rozumieniu relacji między mutacjami w genomie a zmianami w morfologii dokonał się dopiero stosunkowo niedawno, dzięki badaniom nad procesem rozwojowym muszki owocowej (*Drosophila melanogaster*), które doprowadziły do odkrycia genów homeotycznych (Hox). Geny Hox kodują pewien rodzaj czynnika transkrypcyjnego (białka mającego powinowactwo do DNA). Zauważono, że stanowią one rodzaj genetycznego zbioru narzędzi, które odgrywają kluczową rolę w kontroli procesu rozwojowego. Mutacje w genach homeotycznych mają zwykle duże konsekwencje dla wynikowej morfologii rozwijającego się organizmu. Przykładowo, zmiana poziomu ekspresji jednego z genów Hox (*distal-less*) jest odpowiedzialna za zaniknięcie kończyn u węży, a wprowadzając mutację w genie *Antennapedia* można doprowadzić do wyrośnięcia w pełni uformowanej kończyny na głowie muszki owocowej. Te i wiele innych odkryć, które dokonały się w przeciągu ostatnich 25 lat, stały się fundamentem dla rozwoju nowej dziedziny: ewolucyjnej biologii rozwoju (*evo-devo*). Dziedzina ta zajmuje się badaniem procesu rozwojowego w kontekście ewolucyjnym oraz relacjami między zmianami w morfologii a zmianami w genach.

## 2. Przegląd modeli sieci regulacji genowych i embriogenezy

Rozdział 2 pracy zawiera krótki przegląd komputerowych modeli sieci genowych oraz modeli procesu rozwojowego kontrolowanego przez sieci genowe. W ramach omówienia modeli sieci genowych opisano podejście polegające na opisie interakcji pomiędzy poziomami ekspresji genów za pomocą równań różniczkowych. Takie podejście zwykle stosuje się w celu zamodelowania konkretnej sieci interakcji genowych w oparciu o dane eksperymentalne. Następnie omówione zostały podejścia służące badaniu bardziej ogólnych własności sieci genowych, nie zakładające z góry określonej topologii sieci. Opisane zostały losowe sieci boolowskie (ang. random boolean networks, RBNs), czyli rekurencyjne sieci o generowanych w sposób losowy topologiach oraz losowych binarnych funkcjach przejścia wierzchołków. Przedyskutowano własności ich dynamiki (chaotyczność, atraktory). Szerzej omówione zostały modele bezpośrednio związane ze stosowanym w tej pracy, polegające na niebezpośrednim kodowaniu sieci genowej w wirtualnym, linearnym genomie. W takich genomach zwykle wyróżnić można fragmenty, które kodują wirtualne czynniki transkrypcyjne oraz obszary regulatorowe. Czynniki transkrypcyjne, wchodząc w interakcje z obszarami regulatorowymi, wpływają na poziomy ekspresji genów (zwykle kodowanych w bezpośrednim sąsiedztwie tych obszarów regulatorowych). Z większością takich modeli powiązany jest niebezpośredni sposób wyznaczania powinowactwa pomię-

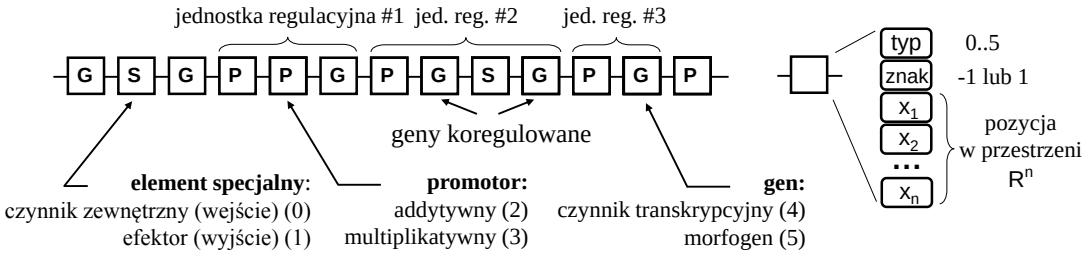
dzy czynnikami transkrypcyjnymi a obszarami regulatorowymi, będący abstrakcją strukturalnych podobieństw pomiędzy cząsteczką czynnika transkrypcyjnego a fragmentem cząsteczki DNA do którego się dowiązuje. Omówione zostały symulowane eksperymenty polegające na analizie własności sieci powstałych z losowych genomów oraz próby ewolucji genomów, które kodują sieci o zadanych własnościach (czy to topologicznych, czy też o zadanej dynamice ekspresji genów lub funkcjonalności).

W następnej kolejności przedyskutowane zostały główne podejścia związane z wykorzystaniem symulowanego procesu rozwojowego. Bardziej szczegółowo zostały opisane modele, w których pożądane morfologie uzyskiwane są za pomocą algorytmów ewolucyjnych, a sam proces rozwojowy kontrolowany jest przez sztuczne sieci genowe. Model przedstawiony w niniejszej pracy należy właśnie do tego rodzaju klasy modeli. Przedstawione zostały również niektóre próby zastosowania tych modeli do rozwiązywania problemów inżynieryjnych, jak np. projektowanie morfologii dla robota oraz użycie procesu rozwojowego jako metody kompresji.

## 3. Model sieci genowej oraz ewolucji

Rozdział 3 zawiera kompletny opis zaproponowanego w dysertacji, inspirowanego biologicznie modelu genomu oraz sieci genowej. Model ten został zaprojektowany tak, by stanowić abstrakcję najistotniejszych własności ewoluujących sieci genowych. Przede wszystkim, topologia sieci zakodowana jest w sposób niebezpośredni w genomie o nielimitowanej długości i liniowej strukturze. Geny kodują własności czynników transkrypcyjnych, które w czasie symulacji komórki mogą w sposób specyficzny dowiązywać się do obszarów regulatorowych na genomie i kontrolować poziom ekspresji genów. Sam proces dowiązywania się nie jest jednak symulowany bezpośrednio. Zamiast tego, powinowactwo wirtualnych białek (czynników transkrypcyjnych) do obszarów regulatorowych danego genu reprezentowane jest jako siła oddziaływania regulacyjnego pomiędzy dwoma genami, tj. jako waga krawędzi w grafie sieci genowej. Czynniki transkrypcyjne w komórce reprezentowane są zaś jako stężenia, wyrażone liczbą rzeczywistą od 0 do 1 i uaktualniane są w sposób ciągły. Ich stężenia zmieniają się w oparciu o obliczoną w każdym kroku symulacji wydajność syntezy danego produktu oraz jego degradacji. Topologia sieci genowej wyznaczana jest na podstawie analizy zawartości genomu i nie zmienia się w czasie życia symulowanej komórki.

Genom składa się z listy elementów o jednorodnej strukturze (Rys. 1), z których każdy należy do jednej z trzech kategorii – produktów (G), promotorów oznaczających elementy regulacyjne (P) oraz elementów specjalnych (S), używanych do zakodowania wejść i wyjść sieci. W celu utworzenia grafu połączeń sieci genowej, genom jest parsowany sekwencyjnie, przy czym za każdym razem, gdy napotkana zostanie seria elementów regulatorowych (P) po której następuje seria genów (G), tworzona jest jednostka regulacyjna stanowiąca wierzchołek w grafie sieci genowej. W trakcie symulacji, jeśli ekspresji podlega jednostka regulacyjna w której zakodowanych jest wiele produktów, wszystkie syntetyzowane są z taką samą wydajnością.




**Rysunek 1:** Struktura wirtualnego genomu oraz elementów genetycznych. Ciąg promotorowych (P) elementów genetycznych po którym następuje ciąg elementów będących produktami (G) interpretowany jest jako jednostka regulacyjna, czyli wierzchołek w grafie sieci genowej. Elementy specjalne (S) kodują wierzchołki odpowiadające ze przekazywanie sygnałów z zewnątrz i na zewnątrz wirtualnej komórki.

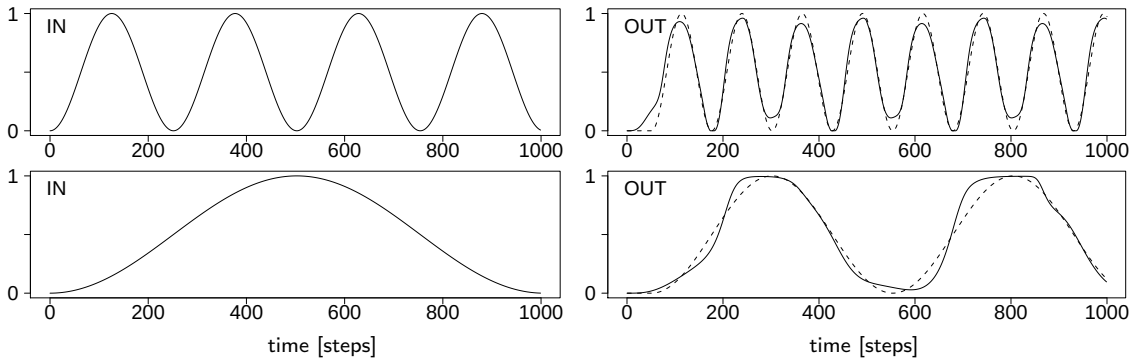
Każdy element genetyczny przechowuje $n$ liczb rzeczywistych (gdzie $n$ jest parametrem modelu), interpretowanych jako współrzędne skojarzonego z nim punktu w przestrzeni $\mathbb{R}^n$. Odległość pomiędzy punktami określonymi przez produkt i promotor pozwala wyznaczyć ich wzajemne powinowactwo (malejące wykładniczo z odległością), które stanowi wagę krawędzi w sieci genowej. W celu uniknięcia grafu pełnego, używana jest odległość odcięcia, powyżej której siła wiązania spada do zera. Taki mechanizm stanowi abstrakcję procesu, w którym trójwymiarowa struktura kodowanego przez gen czynnika transkrypcyjnego (białka) ma strukturalne dopasowanie do konkretnego obszaru na DNA, i do którego dowiązana może promować lub hamować ekspresję zakodowanych w jego sąsiedztwie genów.

Przedstawiona struktura genomu poddawana jest symulowanej ewolucji za pomocą algorytmu genetycznego. Ewolucja zaczyna się od wygenerowania populacji losowych genomów. Stosowane są operatory genetyczne modelujące mutacje zachodzące w biologicznych genomach, takie jak duplikacje całych fragmentów genomu i delecje. Zmiany zachodzą również na poziomie pojedynczych elementów, gdzie dochodzi do zmiany pozycji przypisanego do danego elementu punktu w $\mathbb{R}^n$, co z kolei prowadzi do zmiany powinowactwa, a więc wag krawędzi w sieci genowej. Odpowiada to mutacjom punktowym zachodzącym w cząsteczce DNA. Opracowany został również algorytm pozwalający na krzyżowane genomów, także o różnej długości.

## 4. Ewolucja sieci genowych przetwarzających sygnały

W rozdziale 4 pracy ewoluowalność przedstawionego modelu sieci genowych poddana została analizie. Polegała ona na postawieniu ewolucji za cel uzyskania sieci genowych zdolnych generować zadany wzorzec zmian stężenia w czasie wybranego czynnika transkrypcyjnego. Funkcją dostosowania algorytmu genetycznego był błąd pomiędzy poziomem ekspresji czynnika uzyskanym na wyjściu sieci, a pożądaną odpowiedzią w ustalonym okresie symulacji (rzędu 1000 kroków). Bazowa funkcja obliczająca błąd wspierana była przez dodatkowe człony mające poprawić ewoluowalność.

W pierwszej kolejności uzyskane zostały sieci, w których stężenie czynnika oscy-

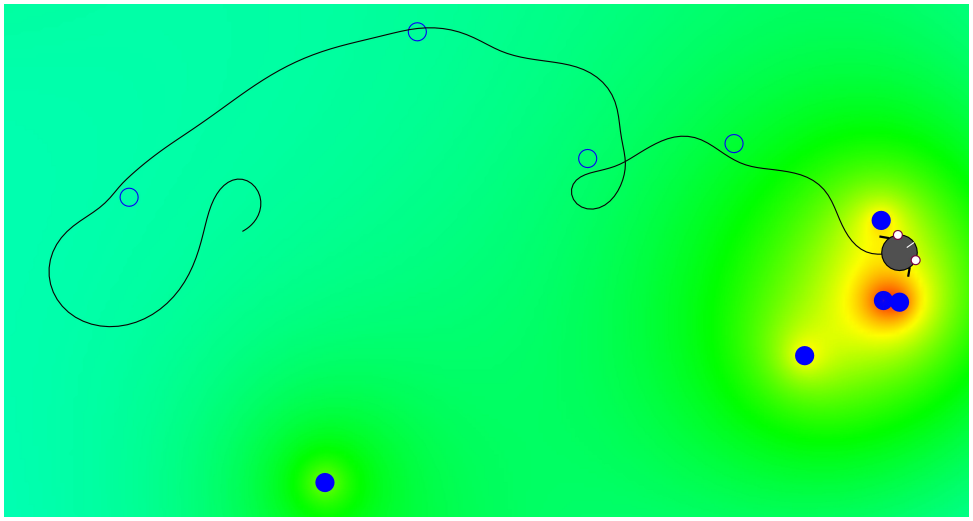



**Rysunek 2:** Zestaw uczący par wejście (IN)/wyjście (OUT) zmian stężenia produktu w czasie, użytych do uzyskania sieci podwajających częstotliwość sygnału wejściowego wraz z odpowiedzią najlepszej sieci. Przerywana linia: oczekiwana poprawna odpowiedź.

luje według zadanego wzorca, bez żadnego zmiennego sygnału zewnętrznego. Następnie zbadano możliwość uzyskania algorytmem genetycznym sieci, które byłyby w stanie w sposób ciągły reagować i przetwarzać zmieniający się w czasie sygnał zewnętrzny, przekazywany do sieci jako sterowane zewnętrznie stężenie wirtualnego czynnika transkrypcyjnego. Aby uzyskać sieci o pożądanych własnościach, zastosowane zostały zbiory uczące w postaci par przykładowego sygnału wejściowego i oczekiwanej odpowiedzi (Rys. 2). Pozwoliło to na uzyskanie sieci, które funkcjonują jako filtr częstotliwości, reagują na czas trwania impulsu wejściowego czy pomnażają częstotliwość oscylacji zewnętrznego sygnału. Co ważne, zaobserwowano bardzo dobrą zdolność do generalizacji: uzyskane na drodze ewolucji sieci nie odpowiadały poprawnie wyłącznie na przypadki uczące, ale bardzo dobrze radziły sobie również z przypadkami pośrednimi, spoza zbioru uczącego. Właściwość ta jest fundamentalna dla większości potencjalnych zastosowań praktycznych sztucznych sieci genowych.

W rozdziale porównana została również ewoluowalność w zależności od rodzaju użytej funkcji dostosowania oceniającej jakość uzyskanej sieci. Wykazano, że dodatkowe człony w funkcji dostosowania, mające pomóc w przeszukiwaniu przestrzeni rozwiązań, w sposób statystycznie istotny poprawiają ewoluowalność. Porównane zostały także dwie wersje modelu sieci genowej: model wykorzystujący zmieniające się w sposób ciągły poziomy ekspresji genów oraz jego wersja, w której aktywność jednostek regulacyjnych zmienia się w sposób natychmiastowy, wyłącznie w oparciu o sygnały na wejściu węzła sieci, działając w ten sposób analogicznie do rekurencyjnych sieci perceptronów (w których poszczególne węzły nie mają stanu). Na badanym problemie, wersja wykorzystująca zmieniające się w sposób płynny poziomy ekspresji genów okazała się generować sieci dające lepsze dopasowanie do pożądanej funkcji docelowej. Sugeruje to potencjalny obszar zastosowań dla sztucznych sieci genowych związany z przetwarzaniem i generowaniem ciągłych sygnałów.

W końcowej części rozdziału zbadany został wpływ szumu w poziomie ekspresji genów na działanie sieci. Porównane zostały sieci ewoluowane bez szumu z sieciami, które ewoluowały z szumem ekspresji genów. Chociaż każdy z rodzajów sieci wykazywał się pewną tolerancją na zaburzenia działania swoich elementów, sieci ewoluowane z szumem osiągały zbliżone poziomy dostosowania do tych ewoluowa-



**Rysunek 3:** Przykładowa trajektoria kontrolowanego przez wyewoluowaną sieć genową wirtualnego organizmu jednokomórkowego odnajdującego źródła energii (niebieskie kółka). Puste kółka oznaczają znalezione źródła energii. Mapa jest pokolorowana zgodnie z lokalną intensywnością sygnału chemicznego (niebieski - niska, od zielony do żółty - średnia, czerwony - wysoka).

nych bez, były jednak kilkukrotnie bardziej odporne na szum. Dotyczyło to również poziomu szumu ekspresji znacznie większego niż ten oryginalnie obecny podczas ich ewolucji.

## 5. Ewolucja zachowania wirtualnych organizmów jednokomórkowych kontrolowanych przez sieci genowe

W rozdziale 5 podjęto próbę zastosowania sieci genowych do sterowania zachowaniem wirtualnego organizmu, a więc wykorzystania sieci genowej jako kontrolera. Organizmy umieszczane były w wirtualnym dwuwymiarowym środowisku, w którym znajdują się punktowe źródła energii. Ich obecność mogła być wykrywana na podstawie dyfundującego w otoczeniu „śladu zapachowego", słabnącego z odległością od źródła. Funkcja dostosowania nagradzała osobniki za ilość pozyskanej w zadanym czasie energii. W toku eksperymentów zaobserwowano, że na przestrzeni kilkuset generacji możliwe jest wyewoluowanie komórek zdolnych do chemotaksji: poruszających się w stronę źródła sygnału dzięki detekcji rosnącego gradientu (Rys. 3). Przeanalizowano również historię ewolucji sterujących najlepszymi animatami sieci i zaobserwowano proces stopniowej optymalizacji ich struktury w czasie i usuwanie nadmiarowych połączeń. Podjęta została również próba ewolucji kontrolerów dla bardziej skomplikowanego problemu, który wymagał poszukiwania źródła energii i unikania "toksycznych" źródeł oraz zamianę ról tych źródeł w czasie życia organizmu. Wymuszało to więc ewolucję sieci, które zależnie od fazy życia symulowanego organizmu prezentują całkowicie odmienne zachowanie dla tych samych sygnałów chemicznych.

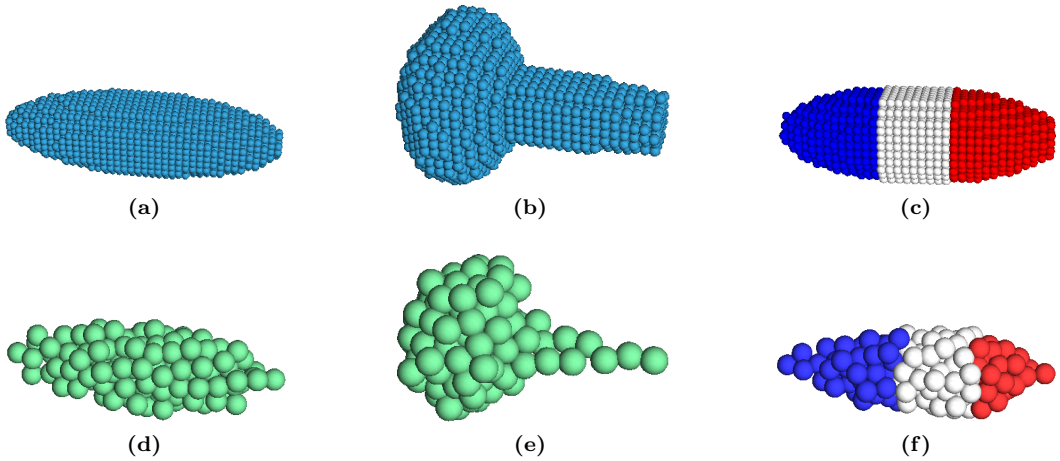
## 6. Ewolucja wielokomórkowego procesu rozwojowego

Rozdział 6 wprowadza opracowany w ramach tej dysertacji model procesu rozwojowego zachodzącego w trójwymiarowym środowisku z symulowaną fizyką, w którym komórki dzielą się i przemieszczają się pod wpływem wzajemnego nacisku oraz utrzymują strukturę dzięki siłom adhezyjnym. Każda z nich sterowana jest opisaną w rozdziale 3 siecią genową, przy czym dodatkowo wprowadzono możliwość emitowania przez komórki dyfundujących w środowisku morfogenów oraz reagowania na nie. Każda z komórek może w czasie swojego życia podejmować decyzję o podziale, zmianie swojej orientacji w przestrzeni (decydującej o kierunku podziału) czy też zaprogramowanej śmierci (apoptozie). Jest to jeden z pierwszych (a według mojej wiedzy pierwszy) z modeli, w którym symulowany proces rozwojowy zachodzi w 3 wymiarach i komórki nie są umieszczone na sześciennej siatce, tylko poruszają się dowolnie w przestrzeni, kontrolowane przez symulowaną fizykę oraz symulowaną sieć genową.

Pierwsza seria przeprowadzonych eksperymentów polegała na uzyskaniu za pomocą algorytmu genetycznego sieci genowych tak sterujących podziałami komórek, by powstała pożądana morfologia (Rys. 4ad,be). Przeanalizowana została historia ewolucji morfologii w czasie, począwszy od sferycznych struktur otrzymywanych w pierwszych generacjach, do uzyskania zadanego kształtu. W celu poznania jak geny wpływają na zmiany morfologii, przeprowadzona została seria symulowanych eksperymentów polegających na inaktywacji genów. Zidentyfikowano geny, których wyłączenie powoduje utratę całej struktury morfologicznej. Następnie przedyskutowane zostało zagadnienie ewolucji embrionów, które potrafią same zakończyć swój proces rozwojowy, bez ograniczeń w postaci sztywnego limitu liczby komórek lub zasobów energetycznych.

W części 6.3 rozdziału zbadana została odporność wyewoluowanych embrionów na uszkodzenia w czasie procesu rozwojowego. Na wybranych etapach rozwoju, określonych jako osiągnięcie zadanej liczby komórek, losowo usuwane były komórki. Następnie proces rozwojowy był kontynuowany, a dostosowanie tak otrzymanego embrionu porównywane było z dostosowaniem embrionu, który przeszedł niezaburzony proces rozwojowy. Zaobserwowano bardzo wysoką tolerancję na uszkodzenia - utrata 25% a nawet 50% komórek była w większości regenerowana. Jako że embriony nie były w czasie swojej ewolucji nagradzane za to w funkcji dostosowania, uzyskaną wysoką odporność na uszkodzenia należy uznać za emergentną własność prezentowanego modelu procesu rozwojowego. Nie zaobserwowano natomiast zdolności do odrastania usuniętych fragmentów embrionu w sytuacji, gdy proces rozwojowy już się zakończył. Zmodyfikowano więc funkcję dostosowania tak, by nagradzała *explicite* zdolność embrionów do regeneracji po zakończeniu procesu rozwojowego. Pozwoliło to uzyskać embriony zdolne do odrastania utraconych fragmentów.

W części 6.4 zostało zademonstrowane, jak model może zostać użyty do jednoczesnej ewolucji morfologii oraz różnicowania się komórek. W tym celu klasyczny w dwuwymiarowych modelach embriogenezy problem różnicowania się komórek na 3 obszary, znany jako problem flagi francuskiej, został przeniesiony do 3 wymiaru.



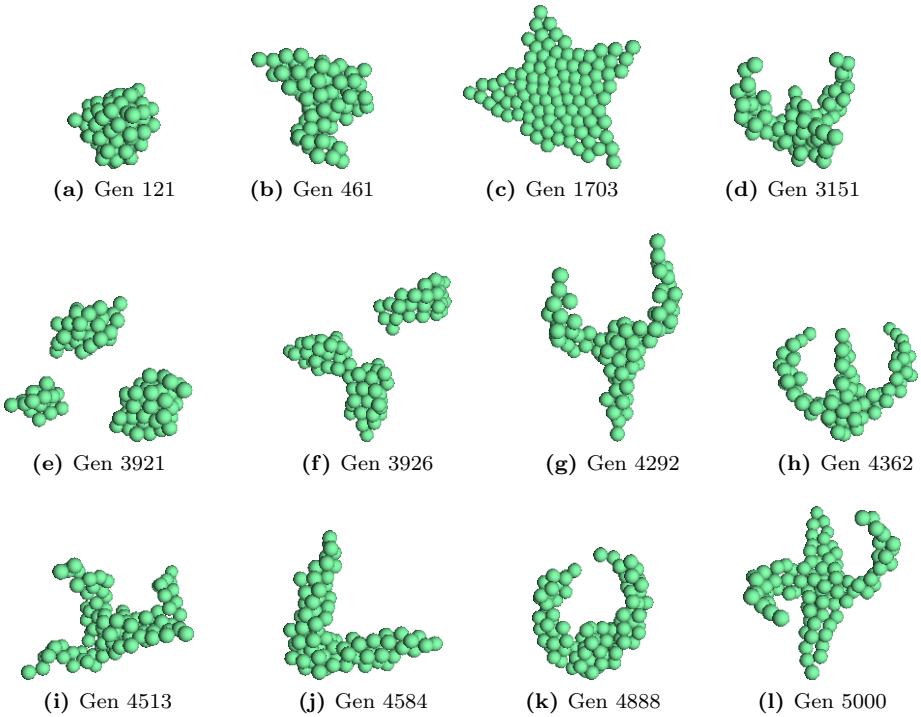

**Rysunek 4:** Ewolucja trójwymiarowych morfologii oraz różnicowania się komórek. (a,b,c) - definicja docelowego kształtu (małe sfery oznaczają voxele na poziomie których dokonywane jest porównywanie uzyskanego kształtu z docelowym), (d,e,f) - wyewoluowane embriony (sfery reprezentują komórki). W przypadku (c,f) jednocześnie ewoluowano zadany kształt oraz pożądane różnicowanie komórek (trójwymiarowa wersja problemu flagi francuskiej).

Z powodzeniem uzyskano trójwymiarowe elipsoidalne embriony, w których komórki różnicują się na trzy lub więcej obszarów (Rys. 4cf). Udawało się to nawet w sytuacji w której komórki nie miały do dyspozycji gradientu zewnętrznego czynnika na podstawie którego mogłyby wyznaczać swoją pozycję. Zaobserwowano natomiast, że ewoluowane w ten sposób embriony same wytwarzały lokalnie podwyższone stężenia morfogenów, umiejscowione na przeciwnych krańcach embrionu, które mogły być wykorzystywane jako sygnały do zróżnicowania się dla innych komórek.

## 7. Otwarta (ang. open ended) ewolucja wielokomórkowych morfologii

W rozdziale 7 zaproponowana została nowatorska metoda pozwalająca na stworzenie systemu, w którym wielokomórkowe morfologie ewoluują w sposób otwarty, tj. bez narzuconej z góry obiektywnej funkcji dostosowania, która premiowałaby konkretne morfologie. W tym celu wykorzystany został algorytm poszukiwania nowości (ang. *novelty search*), zaproponowany przez Lehmana i Stanleya (2008). Nagradza on fenotypy (a więc w tym wypadku morfologie), które różnią się od tych, które istniały dotychczas. Połączenie novelty search z opisanym w rozdziale 6 modelem embriogenezy oraz tylko w niewielkim stopniu zmodyfikowanym algorytmem genetycznym doprowadziło do powstania systemu, który nieustannie generuje nowe morfologie i, podobnie jak biologiczna ewolucja, nie zmierza ku zadanemu celowi. Co najciekawsze, uzyskane morfologie wykazywały się wizualnie o wiele większą złożonością (Rys. 5) niż morfologie, które można było uzyskiwać za pomocą algorytmu genetycznego (rozdział 6).

W rozdziale poddano analizie historię ewolucji wybranej morfologii, odnajdując każdego z jej przodków. Chociaż różnice pomiędzy kolejnymi pokoleniami były



**(a)** Gen 121 **(b)** Gen 461 **(c)** Gen 1703 **(d)** Gen 3151

**(e)** Gen 3921 **(f)** Gen 3926 **(g)** Gen 4292 **(h)** Gen 4362

**(i)** Gen 4513 **(j)** Gen 4584 **(k)** Gen 4888 **(l)** Gen 5000

**Rysunek 5:** Przykładowe morfologie uzyskane w eksperymencie z otwartą ewolucją. Próbka morfologii z zapisu historii symulowanej ewolucji trwającej 5000 generacji.

zwykle bardzo niewielkie, dalsi przodkowie potrafili być bardzo odmienni od analizowanej końcowej morfologii. Jest to sytuacja bardzo odmienna od eksperymentów, w których istniała obiektywna funkcja celu: w takich eksperymentach kolejne pokolenia musiały stawać się coraz bardziej podobne do oczekiwanego kształtu. W rzeczywistości, droga prowadząca do uzyskania pożądanej morfologii może wymagać "przejścia" przez formy pośrednie, które są dalekie od pożądanego kształtu. Klasyczne algorytmy genetyczne pozwalają na to jedynie w bardzo ograniczonym stopniu, nakładając silne ograniczenia na obszary w przestrzeni dostosowania, które zostaną odwiedzone. Zaobserwowana różnica w złożoności morfologii, które można uzyskać za pomocą otwartej ewolucji i klasycznego algorytmu genetycznego, jest przykładem negatywnej presji jaką algorytm genetyczny z obiektywną funkcją celu nakłada na eksplorację interesujących i potencjalnie niezbędnych do odkrycia pożądanego rozwiązania obszarów przestrzeni dostosowania.

W toku dalszych badań podjęta została próba wizualizacji relacji podobieństwa pomiędzy morfologiami istniejącymi na różnych etapach ewolucji za pomocą skalowania wielowymiarowego (MDS). Analiza pozwoliła wyróżnić 3 główne klastry typów morfologii oraz zaobserwować sposób, w jaki stopniowo eksplorowana jest przestrzeń możliwych do uzyskania kształtów. Przeprowadzona została również analiza zmian rozmiaru genomu w czasie i zestawiona ze zmianami w samej sieci genowej.

Zaproponowana w rozdziale 7 metoda konstrukcji systemu otwartej ewolucji jest uniwersalna i można ją zastosować również do innych modeli embriogenezy. Co więcej, konieczne do wprowadzania zmiany w algorytmie genetycznym są stosunkowo niewielkie i sprowadzają się do zamiany funkcji dostosowania na funkcję oceny

nowości. Proponowane podejście może więc być wykorzystane np. jako metoda pozwalająca na ocenę spektrum morfologii, jakie można uzyskać w innych modelach symulowanej embriogenezy. Pozwala też na stworzenie bardziej biologicznie realistycznego modelu ewolucji morfologii oraz sieci genowych niż środowisko, w którym ewolucja zmierza do z góry określonego celu.

## 8. Podsumowanie

Rozdział 8 zawiera podsumowanie uzyskanych wyników oraz przedstawia proponowane kierunki dalszych badań. Przedstawiony w dysertacji inspirowany biologicznie model sieci genowej okazał się cechować wysoką ewoluowalnością na szerokim spektrum problemów, do których został zastosowany. Zbadana została możliwość automatycznego projektowania sieci genowych pełniących funkcje takie jak przetwarzanie w sposób ciągły zewnętrznych sygnałów oraz sterowanie w czasie rzeczywistym wirtualnym animatem/robotem. Zaobserwowana została bardzo dobra zdolność do generalizacji przedstawionych problemów, jak również odporność działania sieci na zaburzenia w działaniu jej elementów, symulowana za pomocą szumu nakładanego na stężenia produktów.

Wprowadzony w rozdziale 6 model procesu rozwojowego zachodzącego w trzech wymiarach i kontrolowanego przez sieci genowe pozwala na ewolucję zadanych trójwymiarowych morfologii oraz wykazuje się emergentnymi własnościami, takimi jak odporność procesu wzrostu wielokomórkowej struktury na zaburzenia oraz uszkodzenia w postaci utraty nawet znacznej części komórek. Zademonstrowano również jak można uzyskać embriony, które są w stanie częściowo regenerować usuwane z nich fragmenty nawet po zakończeniu rozwoju. Model pozwolił również na ewolucję embrionów, których komórki różnicują się przestrzennie. Przedstawiono także system, w którym morfologie nieustannie ewoluują, premiowane wyłącznie za posiadanie formy odmiennej od tych, które występowały wcześniej. System ten umożliwia stworzenie biologicznie realistycznego scenariusza, pozwalającego na badanie własności ewolucji morfologii oraz kontrolujących ich sieci genowych *in silico*, pozwala także na ocenę zakresu morfologii jakie można uzyskiwać w tym i innych modelach procesu rozwojowego.

Oprócz znaczenia przedstawionych badań dla biologii systemów oraz ewolucyjnej biologii rozwoju, stanowią one eksplorację własności nowych metod niebezpośredniego kodowania w algorytmach ewolucyjnych. Modele obliczeń oparte o sztuczne sieci genowe mogą być, podobnie jak sztuczne sieci neuronowe, wykorzystywane jako ewoluowalne układy sterujące. W połączeniu z procesem rozwojowym mogą być zaś wykorzystane do ewolucyjnego wytwarzania niepodlegających ograniczeniom ludzkiej intuicji, zdolnych do samonaprawy i samoorganizacji, modularnych konstrukcji, takich jak np. złożone z wielu jednorodnych komponentów roboty. Zaprezentowany w pracy model procesu rozwojowego zademonstrował już swoją użyteczność do tego celu w toku dalszych, trwających obecnie badań (Joachimczak et al., 2012; Joachimczak and Wróbel, 2012).

# Chapter 1

# Introduction

Only two processes are known to generate complex and sophisticated designs. The first one is human creativity. Faced with a problem to solve, our brains allow us to propose potential solutions, visualize them and to ultimately implement them. By decomposing the problem into smaller modules, we can focus our attention on the appropriate level of abstraction. Importantly, using modular approach we can reuse earlier designs, without a need to understand the detail of every component. This way, we can build on top of existing solutions and create designs of increasing complexity, far beyond full comprehension of a single designer. In that sense, almost every non trivial human design is a product of the mind of its designer as well as of the minds of all the people who contributed to the currently available tool set. Given this powerful capability of our species, it is therefore perhaps not surprising that for the most of the history of the civilization, whenever we were fascinated by the complexity discovered in a natural world, we assumed that it could have only been created by a designer and a one far superior to us. It was not until 150 years ago when we realized that a second, completely natural process can lead to the emergence of such complexity. Its sophistication not only outcompetes that of designs created by our minds but also authored the minds themselves. This process is known as evolution.

Although the concept of evolution is beautifully simple and powerful at its core, the understanding of its implications and mechanisms has gone a long way since it was first proposed by Darwin and Wallace in 1858. We now fairly well understand the molecular mechanisms that encode hereditary information in living organisms as well as the history of life on Earth. At the same time, we have enriched our understanding of the process of evolution and now see it as independent from the substrate: an universal mechanism that is bound to occur whenever certain conditions are met, be it on another planet or inside a computer simulation. These conditions are: the presence of entities that replicate with occasional imperfections and an environment with limited resources. Whenever these two conditions are met, the replicators carrying changes (mutations) that allow them to better exploit available resources will start to outcompete their ancestral type. However, the extent to which this process can lead to an increasing complexity depends on many factors, such as how

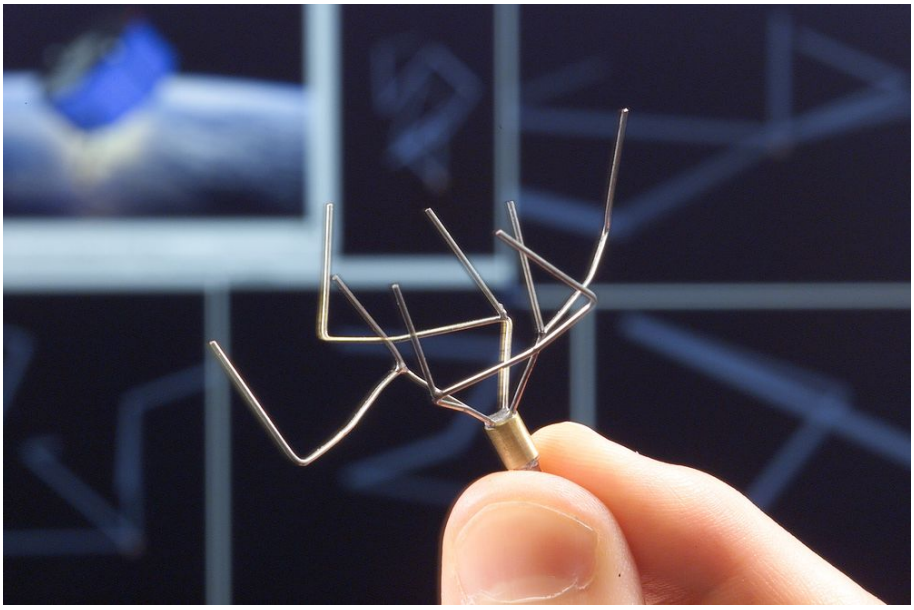


**Figure 1.1:** Evolved novel antenna design used on a satellite in NASA ST5 mission (Hornby et al., 2010). Image source: `http://www.nasa.gov`.

the hereditary information is encoded. They influence what is known as evolvability of a system, which is understood as its capability to generate heritable variation and to acquire novel functions that increase adaptation (Wagner, 2005). The question of how evolvable systems can be created is one of the main interests of this work.

#### Evolution *in silico*

The first computer simulations of evolution date to the early 50s, but it was primarily John Holland's work in the late 60s and 70s that popularized the concept of evolution-inspired optimization method: a genetic algorithm (GA). The growing availability of desktop computers and their ever growing speed increased the feasibility of evolutionary algorithms and resulted in development of various evolution inspired methodologies such as genetic programming (Koza, 1992), evolution strategies (see, e.g., Beyer and Schwefel, 2002) or neuroevolution (Stanley and Miikkulainen, 2002), to name a few. Although GAs are sometimes (rightly) criticized for their computational cost and the need to adjust the algorithm and fitness function to suit the problem for which, more often than not, no hard guidelines exist, they have found their use in many domains. They have been demonstrated to produce working designs that outcompete human designers in performance, but more importantly, also in creativity. Free from the limits imposed by human imagination and paradigms, simulated evolution can tinker with designs that would be very unlikely to be conceived by a human, as illustrated by a novel antenna design in Fig. 1.1. As biologist Leslie Orgel jokingly stated in what is now known as Orgel's Second Rule: "evolution is cleverer than you are".

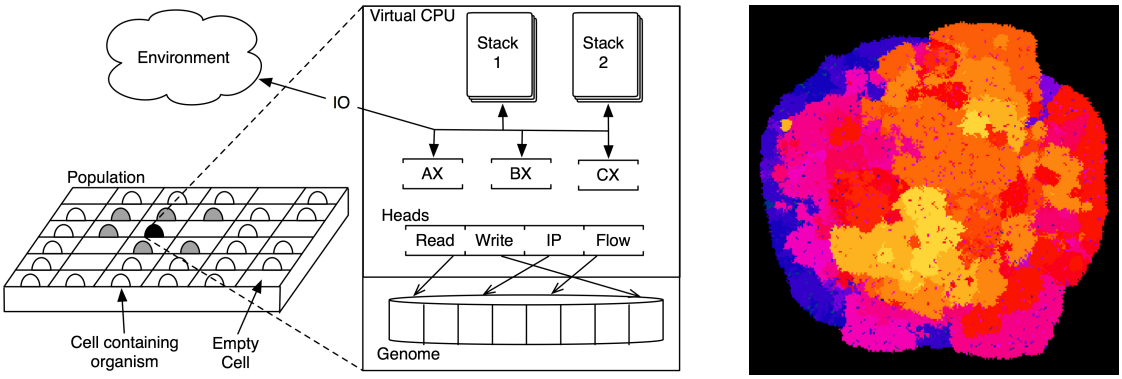



**Figure 1.2:** Avida, an open-ended alife system, (a): the model of an organism and environment, (b): example environment with each species coloured differently. Figures from Beckmann et al. (2007) and `http://kurzweilai.net` (credit: Kaben Nanlohy, Michigan State University), respectively.

### Artificial Life

Artificial Life (or alife, in short) is an interdisciplinary field that emerged from the early computer models of evolution in the mid 80s. It now covers a broad range of research endeavours unified by a common goal of understanding life and its evolution through the use of models. This typically involves computer models, like the work presented here, but also hardware, as is the case in robotics. It can also involve biochemical systems, as is in the case of "wet" branches of alife. Of the principal interest to alife is the process of evolution and emergent phenomena such as self-organization among multiple agents, be it molecules, cells or organisms.

Some of the now classic and most recognized examples of alife projects include, e.g., Tierra (Ray, 1992) and Avida (Adami et al., 1994). Both are artificial ecosystems in which computer programs compete for limited resources (program memory in Tierra, space on a 2D lattice in Avida) and evolve over time. They belong to the open-ended artificial life systems, that is, there is no explicit fitness function and the success of a given individual is only determined by the ability to successfully reproduce in its environment (Fig. 1.2). Such systems can be used to investigate otherwise difficult to test evolutionary hypotheses. For example, Avida was just recently employed to model the origin of altruism (Clune et al., 2008) and the evolution of mutation rates (Clune et al., 2010). Phenomena like parasitism were also observed to quickly emerge in such simulated environments. Yet another well recognized class of alife systems, the artificial chemistries, focuses on a lower level phenomena on the borderline between physics and biology: the emergence of self-organization (Chou and Reggia, 1997; Hutton, 2007; Jędruch and Barski, 1990; Sienkiewicz and Jędruch, 2011). In these systems, each simulated agent represents a molecule, and virtual molecules interact according to the rules of physics and chemistry present in the environment. However, organization of virtual molecules into higher level structures is not defined a priori in the system and emerges through these interactions. Unfortunately, such low level simulations have enormous computational costs and thus they focus on modelling the emergence of life rather than the process of

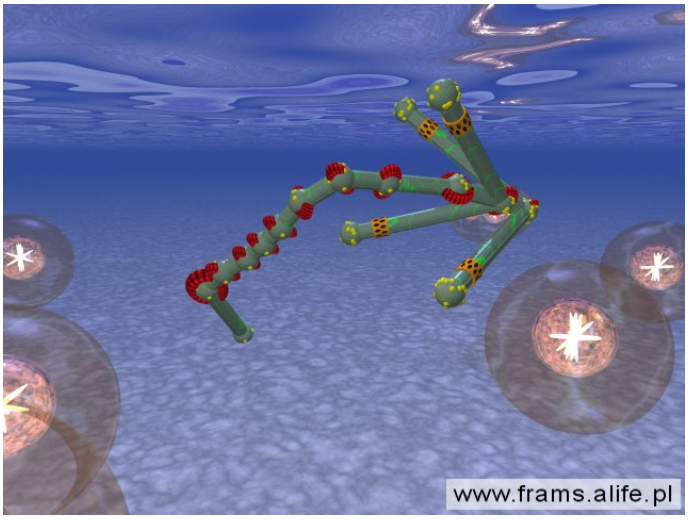



**Figure 1.3:** A swimming creature in Framsticks, a body-brain coevolution system by Komosinski and Ulatowski (1999).

evolution itself. On the other side of the spectrum of levels of abstraction used in alife modelling, whole agents consisting of multiple components (such as blocks and joints) can be evolved together with their controllers. Some of these projects have become icons of alife modelling, e.g., the co-evolution of body and brain in Karl Sim's (1994) 3D creatures, Framsticks (Fig. 1.3; Komosinski and Ulatowski, 1999) or creatures with complex neural vision systems living in an open ended 3D environment PolyWorld (Yaeger, 1993).

#### Thesis context and objectives

The work presented in this thesis belongs to the field of artificial embryogeny, a subfield of alife that focuses on modelling the process of multicellular development, i.e., self organization of cells into an organism. To achieve biological plausibility, it employs a model of the genome that indirectly encodes a gene regulatory network (GRN), a network of interactions between genes that determines the behaviour of each cell. The development itself is simulated in 3D and cells of the embryo interact through a simulated physics.

The objective of this thesis is twofold. On one hand, it attempts to create an evolvable model of GRN controlled multicellular development. Biological development allows to encode information necessary to build a structure consisting of trillions of cells (such as a human body) in a genome which has the information content of only hundreds of megabytes. Developmental process can be seen as an indirect method of encoding a phenotype which, in this case, is a self assembling 3D structure. The scalability and evolvability of this encoding is well demonstrated by the enormous diversity and adaptability of multicellular life on Earth. At the same time, the poor scalability of direct genotype to phenotype mappings have long been recognized in evolutionary computation, and simulated developmental processes are seen as one of the ways to overcome limitations of direct encodings

(Tufte, 2008). This thesis explores evolvability and properties of a highly biologically inspired model of development, together with investigating the evolvability of GRNs (also indirectly encoded in the genome) and their applicability to control problems other than multicellular development.

The second objective was to create a plausible model of evolution of multicellular development that would allow to investigate how genomes, regulatory networks and morphologies evolve and increase in complexity over time. Computer simulations can allow to perform simulation experiments relevant to evolutionary developmental biology (*evo-devo*) that would be otherwise difficult or impossible to conduct.

## 1.1 Thesis layout

The thesis consists of two introductory chapters. Chapter 1 (this) provides an introduction and a brief overview of the relevant biological concepts that this work models and abstracts, i.e., the genomes of biological organisms, regulatory networks and evolution of development. Chapter 2 overviews the prior work on the simulation of regulatory networks (GRNs) and models of artificial embryogenesis employing GRNs. Chapter 3 introduces the model of the GRN and simulated evolution that is the basis of this work. In Chapter 4, the evolvability and properties of the selected approach is evaluated on a range of signal processing tasks. Chapter 5 attempts to create a more biologically plausible setting for simulated GRN evolution and applies GRNs to control real time behaviours of simulated unicellular organisms. Chapter 6 introduces the model of multicellular development and investigates the evolvability of 3D embryos as well as their properties using the GRN model introduced in earlier chapters. Chapter 7 demonstrates an open ended environment for evolution of 3D morphologies. The thesis concludes with a summary and discussion of future directions.

The following sections of the introduction outline fundamental biological concepts that are relevant for the presented work. By necessity, I will ignore many of the intricacies of the discussed mechanisms which were the subject of important discoveries in molecular biology over recent years. Instead, I will focus on the core mechanisms presented from the perspective of computer science.

## 1.2 DNA: life's digital encoding

The diversity of living forms on Earth, their range of scales and sheer complexity is breathtaking. Life evolved to thrive in even the harshest conditions such as freezing temperatures at the poles or enormous pressures and high temperatures surrounding undersea hydrothermal vents, up to 5 kilometres below the surface, where the main source of energy are chemicals and heat. But what is perhaps even more amazing and a testament to the common ancestry is that all living organisms, no matter how diverse, all depend on DNA to encode their genetic information.

The concept of discrete units of heredity (i.e., genes) reaches to the 1860s works of Gregor Mendel. Although his works were published right at the time when Theory of Evolution was bringing a sudden paradigm shift to biology and was in desperate need for the explanation of how traits can be passed from parents to offspring, Mendel's works remained mostly unnoticed by a larger scientific community. It remained so until their rediscovery more than three decades later. At about the time Mendel was performing his experiments on peas, Ernst Haeckel, one of key figures of the evolutionary revolution, basing on microscopic observations, postulated that cell nucleus carries hereditary information. However, it was not until 1884 when one of his students, Oscar Hertwig, would confirm so experimentally. The next 50 years brought incremental discoveries such as the fact that genetic information is stored in chromosomes as well as the discovery of nucleic acids, RNA and DNA. Still, however, without understanding of their structure. Independently of the increasing understanding of biochemical mechanisms of the cell, mathematical models of evolution were being developed, leading to the formulation of what is now known as a the modern synthesis. A proof to the generality of evolutionary principles, even though it was developed without knowledge of molecular mechanisms, the modern synthesis remains the currently accepted account of evolution 80 years later.

The understanding that DNA molecules carry the genetic information came in 1944, thanks to what is now known as the Avery-MacLeod-McCarty experiment. The hunt was on to discover what actually constitutes this information. It culminated with the discovery of the doubly helical structure of the DNA molecule by Watson and Crick in 1953 (awarded with the Nobel Prize). It was followed a few years later by the discovery of how genetic information is encoded. It employs an alphabet of 4 nucleotides and the so called "genetic code" to encode protein sequences in DNA.

However, understanding how the information is stored is very different from knowing how this information leads to the creation of an organism. Although we now understand the basic molecular mechanisms of DNA replication and protein assembly, the larger scale view of how genes and their interactions lead to the formation of a multicellular organism is still a subject of ongoing and exciting research. Hence, even though we sequenced genomes of multiple organisms, including humans, understanding the meaning of this information will remain a challenge for many decades to come.

On its highest level of organization, taxonomy divides life by the presence of nucleus in their cells into Eukaryota (having a nucleus) and Prokaryota (lacking a nucleus). Eukaryota are organisms that include, among others, plants and animals. There are two groups of Prokaryota: Archaea and Bacteria, both taxons of equal standing to eukaryotes. The smallest cells of eukaryotic organisms are an order of magnitude larger than the smallest prokaryotic cells and differ in many aspects. For example, DNA of prokaryotic organisms is typically organized into a single circular molecule, whereas eukaryotic DNA is organized in many linear, tightly packed chromosomes residing inside a nucleus. Typically, most of DNA in prokaryotes encodes proteins ($\sim 90\%$), whereas often less than 5% of eukaryotic genomes does so.

The number of nucleotides in a genome for prokaryotic genomes is in $10^{5-6}$ range and in $10^{7-9}$ range for eukaryotic genomes. To put it in terms of data storage, this translates to $\sim$ 120kB (kilobytes) for the smallest known bacterial genomes and a few megabytes for the largest known bacterial genomes, which is also the size of the smallest known genomes of eukaryotes. Humans (and mammals in general) store around 800MB in their genomes, but as tempting it would be to associate genome size with the apparent complexity of organisms, the salamander's genome is 40 times larger than ours. Furthermore, some plants are known to have very large genomes ($\sim$ 50GB for the currently largest known plant genome, Pellicer et al., 2010), though such enormous genomes often consist of several copies of an ancestral genome that became repeatedly duplicated. The lack of correlation of apparent complexity of an organism with its genome size is recognized as the C-value paradox (see, e.g., Gregory, 2001).

## 1.3 Biological genomes and gene regulation

DNA is a macromolecule in a form of a long chain of nucleotides. Each nucleotide consists of a sugar and phosphate backbone and one of the 4 possible bases: adenine (A), guanine (G), thymine (T) or cytosine (C). In its most common form, DNA consists of two complementary strands, structured into a double helix and held by weak hydrogen bonds between complementary pairs (A-T, G-C). The actual physical length of the molecule can be considerable and is in the range of centimetres for human chromosomes. This gives around 2-3 metres of DNA that fits in a $\varnothing 1\mu m$ nucleus of every cell. It is only possible thanks to the very dense packing that literally wraps DNA around proteins specialized for that purpose (histones), much like "beads on a string", which are further folded into a higher order helix.

The meaning of the term "gene" evolved over time with our increasing knowledge of mechanisms of heredity and can easily become a source of confusion. In a most general and accepted sense, a gene is a functional fragment of a genetic sequence. However, many genes encode proteins and so, in the context of protein synthesis, the term "gene" is synonymous with a fragment of a genome that encodes a sequence of a protein, i.e., a "protein coding gene" (for a summary of different uses of the term see, e.g., Gerstein et al., 2007). But in its most general sense, the term encompasses not only protein coding genes, but any functional genetic sequences, for example, regions of DNA that are important for gene regulation or fragments that undergo transcription to RNA but do not result in proteins.

### 1.3.1 From DNA to protein

The read out of a genetic information leading to the synthesis of a protein is a two step process. It consists of transcription and translation. First, the sequence of a single gene determines the sequence of a messenger RNA molecule (transcription), which is then used as a template for creation of a protein (translation). This means that the information encoded in a genome is ultimately converted into protein

molecules. The observation that information can only flow from DNA to proteins but not from proteins to DNA is referred to as the central dogma of molecular biology. The essential features of transcription and translation are shared between both prokaryotic and eukaryotic organisms, the details can however be quite different and reflect the fact that the eukaryotic lineage added its own mechanisms on top of what its bacterial ancestor would already have.

#### Transcription

Transcription results in the creation of a messenger RNA molecule (mRNA). Just like DNA, mRNA is a chain of nucleotides but it is single stranded. The sequence of mRNA is complementary to one strand of the DNA. In both eukaryotes and prokaryotes the genomic DNA is double stranded. The strands are complementary and have opposite directions. When nucleotides are bound together to form nucleic acids, it is always the so called 3' end that is being extended (the number corresponds to a carbon atom in the sugar moiety, ribose or deoxirobose). It is common to refer to regions as "downstream" (or in 3' direction) and "upstream" (5' direction) of a site using the direction of a relevant strand. The process of transcription is performed by an enzyme (a protein) called RNA polymerase (RNAP). RNAP binds non specifically to DNA (in fact, many of RNAPs will do so in parallel every second). Bound RNAP initiates a linear search for a region that will signal the beginning of transcription, typically a particular sequence of nucleotides located at very specific distances from each other. Upon finding it, RNAP will then start synthesizing a complementary mRNA chain until another region signalling an end of transcription is detected. To allow for physical access to nucleotides, the hydrogen bonds between two DNA strands are temporarily broken during the read out, forming a moving "transcription bubble". Even before the whole mRNA molecule is synthesized, transcription starts in prokaryotes. In eukaryotes, it is necessary first to transport mRNA out of the nucleus. Before it happens, most eukaryotic mRNA undergoes a process called splicing. Splicing starts already before mRNA synthesis is finished and involves removal of pieces of mRNA (so called introns). The final mRNA molecule is spliced (hence the name) from the remaining pieces (exons) one-by-one.

#### Translation

Translation results in the creation of a protein using mRNA transcript as a template. Proteins are macromolecules consisting of a chain of amino acids (sometimes multiple chains) and serve essential roles for the functioning of an organism. They function as enzymes (i.e., they catalyse chemical reactions), regulate cellular processes and transcription of genes. Proteins also have structural roles, forming the scaffolding of cells. Organisms on Earth rely on only 20 different amino acids, limit to a few exceptions: two other amino acids are used by some prokaryotic organisms. The average length of a protein amino acid chain is about 500 amino acids, the longest known is above 20 000. However, proteins do not maintain a linear structure,

but already during the synthesis of the chain they undergo a process known as folding, during which they obtain a complex 3D structure (conformation). It is the conformation that determines the chemical properties. The folding occurs due to the chemical properties of their interacting amino acids and can undergo spontaneously through a process of energy minimization or with the help of additional proteins (chaperones). Prediction of a native conformation of proteins based on their amino acid sequences is currently one of the most important challenges and active research areas of bioinformatics. It is of great relevance to medicine and drug design.

The order of amino acids is encoded directly by the sequence of nucleotides in the mRNA molecule. This encoding is known as the genetic code: each triplet of nucleotides (codon) represents a single amino acid. Since this gives $4^3 = 64$ possible codons that need to encode 20 amino acids, the genetic code is redundant. Furthermore, some of the codons are used to mark the beginning or the end of amino acid chain assembly. Since an enormous number of such translation tables is possible, the fact that essentially the same genetic code is shared between all living organisms is a powerful evidence for the common ancestry of all life on Earth. There is also recent evidence suggesting that the code is far from being random and underwent evolution that optimized its robustness to mutations and fault tolerance, prior to the existence of the common ancestor of all currently living organisms (Freeland et al., 2003; Tlusty, 2007).

The process of translation is performed by a molecular complex called a ribosome. Just as multiple RNAPs exist in each cell and work in parallel, so do multiple ribosome molecules. They bind to mRNA molecules and assemble proteins using mRNA as a template. To make it possible, adequate amino acids have to be delivered to extend the amino acid chain after each new codon had been read out. This is done with the use of transfer RNA (tRNA) molecules. tRNAs are RNA molecules folded in a very particular 3D shape. On one end they expose an anticodon: a triplet of bases that will match a particular codon in an mRNA molecule. Its opposite side binds an amino acid that matches this particular codon. Hence, at least 20 different types of tRNA have to exist (and are encoded in the genome), and considerable numbers of such molecules have to be continuously present in a cell to allow for protein synthesis. tRNA molecules attach to a ribosome bound to an mRNA transcript and release their amino acid which joins the so far build chain. The freed tRNA molecule is then ready to be reused and to bind another free flowing amino acid.

### 1.3.2 Gene regulation

The process of critical importance for the workings of a cell and central for this work is that of gene regulation (here, the word "gene" refers to genes encoding proteins). Cells need to control what kinds of proteins are synthesized and in what amounts. Also, for prokaryotic cells, synthesizing proteins that are currently not needed is a non-negligible energetic cost. Gene regulation can occur on various levels, and for all levels multiple mechanisms are already known, with the list likely to be extended.

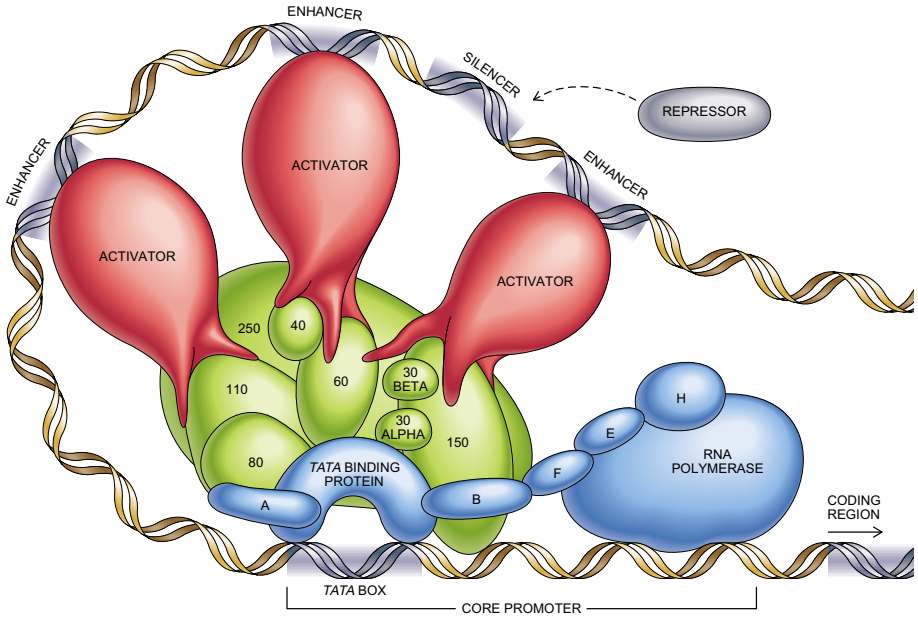



**Figure 1.4:** Regulation of eukaryotic transcription. Blue: basal factors, green: co-activators (drawing by Jared Schneidman, reproduced from Tjian, 1995).

The lowest level is transcription: the rate of synthesis of a given protein can be controlled by influencing the rate at which transcripts are created which can be, e.g., done by interfering with RNAP binding to DNA. The rate of synthesis of a protein itself can also be affected by active degradation of mRNA before they get to be translated into a protein (one of the many mechanisms of post-transcriptional regulation). The same applies to regulation at the level of translation. Proteins can be actively degraded, become deactivated by inhibitor molecules, or require activating molecules (which, again, can have different concentrations) to become functional.

Since transcription is a first stage necessary for the synthesis of a protein, regulation at this level can be expected to be of a key importance. The principal mechanism through which it occurs involves transcription factors (TFs). TFs are proteins that have an affinity to DNA. By binding to sites located in the vicinity of the protein coding regions, they can enhance or silence the transcription of this region, ultimately influencing the synthesis rate of a protein. One of the classes of eukaryotic TFs are TATA binding proteins (TBP). They identify a very particular sequence on DNA which signifies the start of a transcribed region: T-A-T-A. Together with other TFs that bind to them, TBPs belong to so called basal factors, factors that help recruit and position RNA polymerase before the coding region (Fig. 1.4). Other TFs, known as activators, bind to enhancer sequences and through coactivators increase the probability of TBPs binding in a given region. TFs from yet another class, repressors, bind specifically to sequences known as silencers and interfere with formation of the RNAP complex. Hence, the interplay between various TFs and the presence of specific sequences around the protein coding region (often upstream) influences the rate of mRNA synthesis and is the basis of gene regulation logic.

The first complex regulatory circuit identified and now a classic textbook example

of gene regulation is the *lac-operon* in *Escherichia coli* (a prokaryote). One of the controlled genes encodes a protein (enzyme) that allows the cell to digest the sugar lactose. However, the enzyme is produced if and only if lactose is detected in the environment, and if glucose, a preferred energy source, is not present. In all other cases, transcription is inhibited (see, e.g., Berg et al., 2002, for a detailed explanation).

### 1.3.3 Gene regulatory networks

The network of interactions between environmental signals, proteins and gene expression can be depicted as a graph in which vertices represent products of genes and edges represent regulatory interactions. A transfer function can be specified for each node representing the logic behind regulation of the synthesis of a given gene product. In practice, this logic is often limited to specifying whether an interaction represented by an edge should be treated as inhibiting or enhancing. Even though gene regulatory networks often hide complexities of many levels of possible regulatory interactions behind a single edge in a graph, they are a useful level of abstraction for understanding the logic behind processes that occur in cells. The hidden complexity also reflects the fact that often we do not know and do not have an easy way to determine how regulation occurs. Furthermore, from the point of view of understanding how the system behaves, we are usually interested only in finding out how concentrations of one protein will influence concentration of another. It should therefore be not surprising that the past two decades have brought a rapidly increasing interest in decoding the structure of regulatory networks.

Reconstructed regulatory networks of biological organisms can be very complex (see Fig. 1.5 for an example), and it can be very laborious to decode them. Establishing the existence and the nature of each edge requires a series of independent experiments and, sometimes, dozens of scientific papers. However, the recent progress of high throughput methods that allow to observe patterns of expression of multiple genes over time enables less accurate but higher scale decoding based on correlations in changes in expression levels. Our knowledge of regulatory networks is still very incomplete, but it appears that regulatory networks exhibit a high degree of modularity, and the global network of interactions consists of many modules that evolved to control different processes. Hence, each such module can, to an extent, be analysed and understood separately. Still, the actual extent to which our current view of modularity of gene regulatory networks represents actual modularity in their global structure, or is an artefact of our limited capabilities to decipher their structure, is part of an ongoing debate.

It is expected that statistical properties of regulatory networks are not random and have been shaped by selection pressures such as the need to be robust to damage and interference as well as by the nature of mutations through which genomes evolve (especially, the effects of gene duplications). Hence, there is a growing interest in determining those global properties. This is being investigated both on biological data as well as through modelling paradigms, of which this thesis is a part of.

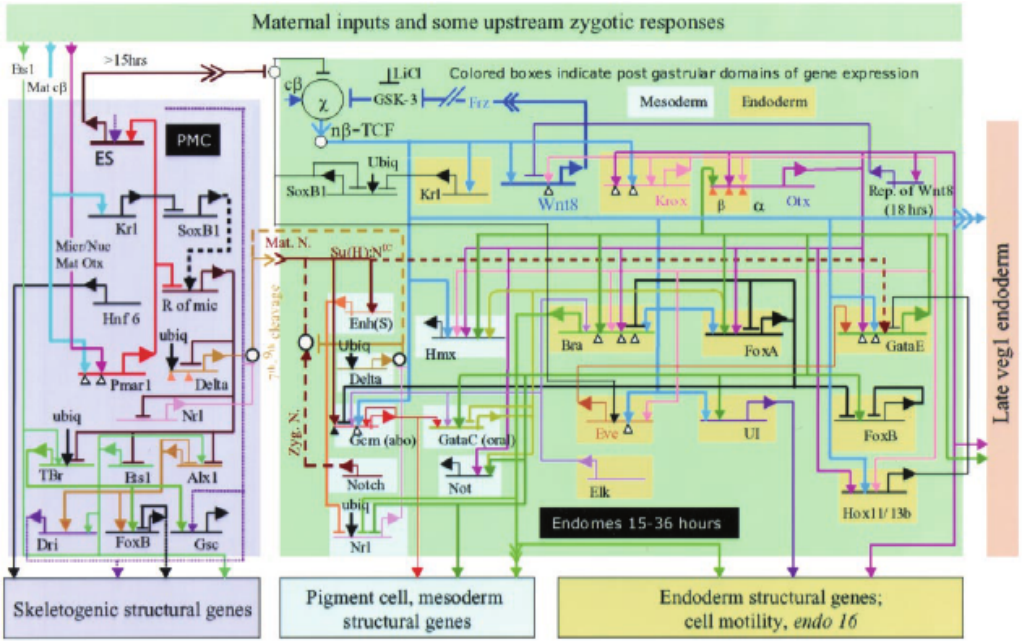



**Figure 1.5:** Fragment of the regulatory network controlling the initial stages of development of a purple sea urchin (reproduced from Davidson et al., 2003).

For example, apart from being modular, it has been postulated that regulatory networks are scale-free and have the small-world property (see, e.g., Barabási, 2009; van Noort et al., 2004; Wuchty, 2001). Furthermore, certain motifs (subgraphs) were postulated to be overrepresented depending on the functionality of a network (Milo et al., 2002).

## 1.4 Multicellularity and embryogenesis

Multicellular life evolved from unicellular organisms more than once during the history of life on Earth. Currently, more than twenty such evolutionary transitions were identified in different lineages. The oldest is postulated to have occurred among cyanobacteria, 3.5 billions years ago (Bonner, 1998), i.e., relatively quickly after the appearance of life and almost 3 billions years before complex multicellular eukaryotic organisms that we typically associate multicellular life with. Just recently, it has been demonstrated that primitive multicellularity can evolve in the lab in just a few dozens of generations, given an appropriate selection pressure (Ratcliff et al., 2012). What are the benefits of being multicellular? For one, in many environments there is an advantage of simply being bigger. Furthermore, being multicellular allows for a higher level of functionality through cell specialization. This however requires additional, much more sophisticated cell regulation that will allow cells to differentiate and take on different roles, despite having the same genome and thus the same controller (i.e., a regulatory network). Two cells with the same regulatory network can only behave differently because the state of each cell is determined by its current concentration of the proteins in the cytoplasm (especially, the transcription factors). Hence, if due to some external factor (or even noise) a particular TF reaches

different concentrations in two cells, it can set each of them on a different regulatory trajectory, further activating or deactivating different regions of regulatory network through a cascade of interactions

Of principal interest, both because of its complexity and relevance to humans, is the embryogenesis (i.e., multicellular development) of animals. All animals start their life from a single cell and develop through successive cell divisions. Related organisms that are visually quite different such as, e.g., a whale or a bat, display remarkable similarities at their earlier stages of development, a phenomenon that was recognized already at the times of Darwin and popularized by iconic (albeit exaggerated to emphasize certain features) drawings of Ernst Haeckel. Also, at adult forms, organisms display homologies (similarities stemming from common descent). Features such as limbs share a common skeletal plan between all vertebrates and, even more interestingly, are clearly modular in their structure. For example, each digit in a limb consists of a number of smaller bones that usually differ only in size. On the other hand, the number of digits can be different even between closely related and otherwise visually similar species. In fact, a mutation that can change the number of digits is not uncommon even among humans (the result is known as polydactylia). This suggests that a relatively simple genetic change must be behind it. Although such additional digit is usually degenerate, it is still not that rare for it to be fully functional, i.e., connected by nerves, under the control of the brain and providing sensation. All this shows that the process of development has a remarkable level of modularity: some regions of DNA can influence how many digits will be created and there is no need to specify how to connect each of them with nerves and veins: these will be laid out accordingly to the number of digits being built. It is thanks to this modularity that evolution can continuously discover and test slight variations to existing forms through apparently relatively simple changes in the genome: mutations can "tweak" the parameters of certain morphological features by controlling the amount of growth of certain bones or overgrowth of tissue, even repeating whole structures (see, e.g., West-Eberhard, 2003, for a comprehensive introduction). This plasticity of evolution has been realized already at the times of Darwin, but it was not until early 1980s when light had been shed on actual genetic mechanisms governing the development, thanks to the research on a fruit fly *Drosophila melanogaster* and the discovery of the homeotic (Hox) genes.

Hox genes encode a particular type of protein. These proteins all have characteristic 60 amino acid domain (homedomain), and genes that encode them form clusters on the genome. Interestingly, the relative order of Hox genes on the DNA corresponds to their expression patterns along the axis of the developing organism (Fig. 1.6). The mechanism that allows for this pattern of expression is not yet fully understood. Hox proteins function as TFs, i.e., they bind to enhancers of other protein coding genes to promote or inhibit their expression. Some of those genes were identified as triggers that initiate pathways that lead to a formation of whole morphological structures such as a limb. Hence, mutations to Hox genes can results in spectacular and large scale mutations, such as mutations to *Antennapedia* gene, which results in a fully formed legs growing in place of antennae on a fruit fly's

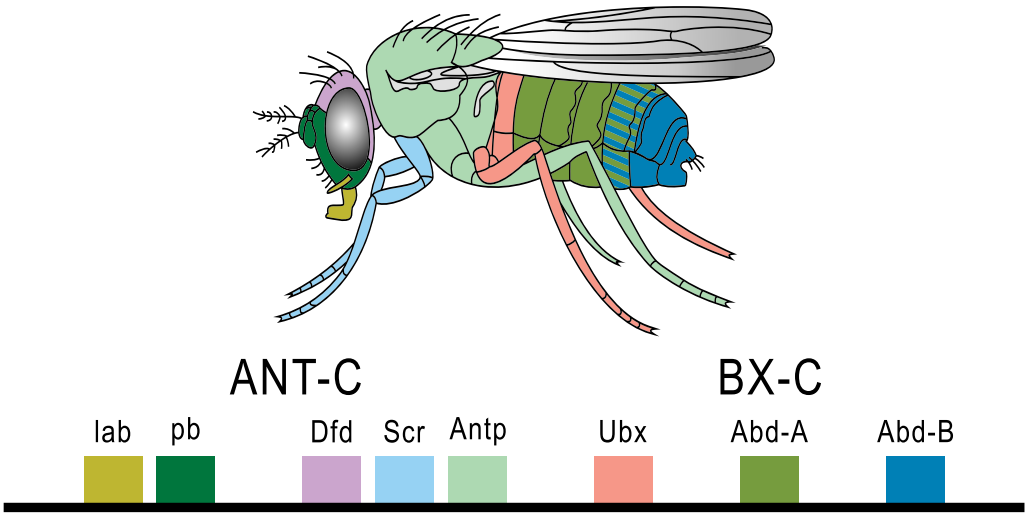



**Figure 1.6:** Two clusters of Hox genes on the fruit fly DNA and corresponding body segments in which they are expressed (source: `http://commons.wikimedia.org/wiki/File:Hoxgenesoffruitfly.svg`).

head. A testament to their importance, Hox genes were found to be remarkably conserved across the animal kingdom. They are present both among invertebrate and vertebrates (the latter having 4 Hox clusters). The level of similarity in lineages that are more than 500 million years apart is so high that a chicken Hox protein can replace the corresponding Hox protein in a fruit fly and will lead to a normal development (Lutz et al., 1996).

Hox genes are currently recognized as a part of a larger developmental-genetic toolkit: a repertoire of highly conserved TFs and morphogens that take part in laying out the body plan of an organism. By generating different overlapping spatial patterns of their expression and through the regulatory logic, an embryo can be divided into subregions of increasingly finer detail in which appropriate pathways become activated. For example, TFs encoded by the so called pair-rule genes divide the early fruit fly embryo into bands, closely related to future body segments (Fig. 1.7). Whether some of the segments will turn into frontal or tail segments is then determined by other transcription factors setting up the anterior-posterior axis of the embryo. In the case of the fruit fly, the anterior-posterior axis is determined by maternal factors that are predeposited in the egg, called *Bicoid* and *Nanos*.

The discovery of the developmental toolkit became a foundation for a new discipline called evolutionary developmental biology (evo-devo) and brought the 1995 Physiology and Medicine Nobel Prize to Eric F. Wieschaus and Edward B. Lewis for their work on Hox genes. The high level control that the toolkit genes have over development has great implications for how evolution of animal forms have progressed and how novel features can appear. Mutations that result in changes of levels of expressions of those genes or regions in which they are expressed can have impact on large scale structures of an organism such as body segments or limbs. Whole features can be reshaped or even added and removed. Mutations with results at the segment level can be seen in particular among arthropods. The number of body

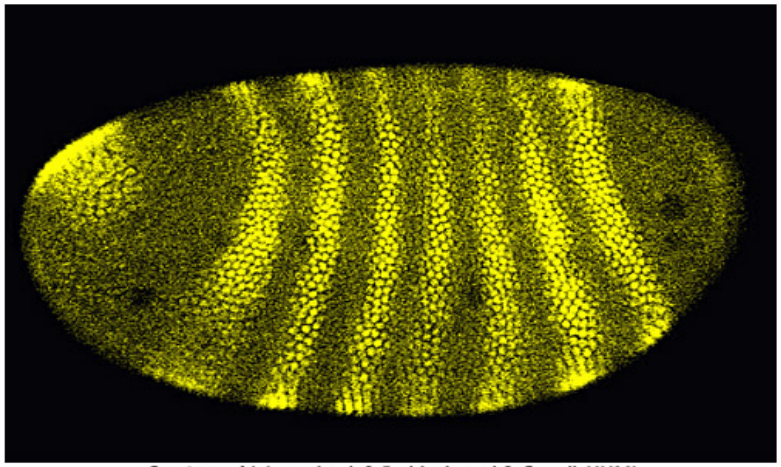



**Figure 1.7:** The pattern of expression of pair rule genes in the fruit fly embryo: the regions are closely related to the body segments.

segments and the nature of protrusions underwent a variety of changes. Although now different segments host different protrusions (e.g., wings, antennae or legs), all of these differentiated from the same protrusion type in the common ancestor. Despite large visual differences between lineages, arthropods share the same overall segmented body plan and the same genetic toolkit. The effects of mutations in the genetic toolkit were important as well for vertebrates and, for example, the loss of limbs in snakes can be attributed to the reduction in expression of a particular Hox gene (*distal-less*).

The rapidly growing science of evo-devo is currently reshaping our views on the role of mutations to protein coding genes during evolution. What it suggests is that evolution acts mostly on regulatory regions of DNA and provides variation within a largely conserved set of protein coding genes and within highly constrained body plans. It shows how duplications played a major role in the complexification of the body plans and also provides a possible explanation why organism can differ considerably despite overwhelming similarity of their DNA sequences (e.g., 99% of similarity between human and chimpanzee sequences) as well as to the fact that only 1.5% of the human DNA seems to encode proteins.

Interestingly, the discovery of the genetic toolkit provides also a new potential explanation for the Big Bang of animal evolution known as the Cambrian Explosion, a short (in the geological time scale) period of a few dozen millions years. It appears that during this time, relatively simple soft tissue organic forms with hard structures, which were fossilized, were replaced by a great diversity of life that included most major phyla and existing animal body plans. Although probably the key trigger were the changing environmental conditions that created new ecological niches, the evidence suggests that the genetic toolkit emerged before the Cambrian explosion and favourable conditions may have allowed evolution to quickly explore a variety of body plans and variations of body segments and appendages. Nonetheless, the issue of relative importance of mutations in regulatory regions and these to protein

coding sequences is a matter of lively debate in the field of evo-devo (see, e.g., Haygood et al., 2007; Hoekstra and Coyne, 2007) and, for a popular introduction to the concepts of evo-devo, the best selling "Endless forms most beautiful" (Carroll et al., 2004).

# Chapter 2

# Existing models of GRNs and embryogenesis

This chapter provides a short overview of a related work and focuses on existing approaches to gene regulatory network modelling and to creating artificial developmental systems. In general (and that applies to all biologically inspired fields of computation), there are two main reasons for the interest in creation of biologically inspired systems. One is purely scientific and is fed by the desire to understand the nature of biological processes and to reveal the rules that govern them. The other is an engineering one: the hope is that biologically inspired techniques and methodologies can be applied to practical problems, either now or in the foreseeable future. Although humans can create machines of incredible complexity and power, even the simplest biological organisms far exceed capabilities of human designed systems when it comes to failure tolerance, ability to self repair and to function in a wide spectrum of environmental conditions. By inspiring ourselves with solutions and approaches discovered over 4 billions of years of evolution, we will hopefully be able to construct artificial systems that share these properties.

## 2.1 Models of gene regulatory networks

The most general approach to model biological gene regulatory networks (and probably the most used in biological sciences) is to describe the relationships between concentrations of different gene products as a set of ordinary differential equations (ODEs) that represent the kinetics of product synthesis and degradation. The synthesis rate is influenced by the binding of transcription factors to DNA, typically simulated with the use of models of enzymatic reactions, such as the Michaelis-Menten kinetics. Since regulation of gene expression is known to occur on many levels and to be influenced by a variety of mechanisms (mRNA transcription/degradation, RNA interference, regulated protein degradation or protein modification, just to name a few), ODEs can be constructed to describe interactions at the desired and potentially very high level of realism. Naturally, because more complex models have more

parameters and because many of those processes are still not known well enough to be characterized kinetically, simpler models that hide many complexities in a single relation are usually preferred. A regulatory network described in this way can then be simulated using iterative ODE solver (e.g., using the Runge-Kutta method) or be analysed as a complex dynamical system for which steady states, attractors and natural cycles of oscillations can be determined. However, typically this type of modelling approach is suitable for simulation of the dynamics of a known (or hypothesized) and usually relatively small gene networks, for which the reaction constants can be fitted to the existing experimental data (see, e.g., Alon, 2006; Bolouri, 2008, for an introduction to GRN modelling in biology).

The approach discussed above, although allows for a high degree of realism, assumes continuous levels of concentrations and is fully deterministic. In some situations, especially when the numbers of involved molecular species are small, the stochasticity associated with each reaction can considerably influence the behaviour of a system. In such a case, a stochastic simulation can be used. An ODE based model can be converted into a stochastic model by assuming initial numbers of each molecular species and simulating the system, e.g., using the Gillespie algorithm (1977). This allows to directly compare the results of stochastic simulations with deterministic solutions to find out whether the observed behaviour of a system is robust in the presence of noise or with low molecule counts.

Quite often, a much simpler model can explain observed interactions: a Boolean network. In this case, the regulatory network is modelled as a directed graph in which vertices represent genes and a link between vertices represents regulatory interaction between genes. The defining feature is that each vertex in every time step can be in a binary state of either activity or non activity. A Boolean transfer function is associated with every vertex and is used to calculate its state in the next time step, based on the states of its predecessors (nodes that link to it). This means that all types of regulation known from molecular biology are reduced to a simple binary presence of a factor or a lack of thereof. Although greatly simplified in comparison to dynamics described by ODEs, Boolean networks have been frequently demonstrated to be a useful parsimonious model describing interactions in simple genetic circuits.

Probably the most studied and influential type of Boolean networks are random boolean networks (RBNs), proposed more than 40 years ago by Kauffman (1969, 1993). Their connectivity as well as transfer functions of each node are initialized randomly. Then, during simulation, the state of all nodes is updated synchronously. Since the number of states in a network of size $N$ is finite ($2^N$), such networks are guaranteed to revisit an earlier state and repeat their behaviour. However, the length of such repeating cycle is typically orders of magnitude lower than $2^N$. In such cases, networks are said to have entered an attractor which can either be a single steady state (a point attractor) or repeating series of states (a cycle attractor). Kauffman investigated how the networks respond to perturbation by randomly changing the state of a single node. He observed that depending on the average number $K$ of inputs to a node, the perturbations would tend to quickly die out in the networks (an

ordered behaviour, $K < 2$), affect a limited number of nodes (a critical behaviour, $K \sim 2$) or propagate through the whole networks (a chaotic behaviour, $K > 2$). He then suggested that regulatory networks of living organisms evolved to function on the borderline of chaos and order ($K \sim 2$) as this allows for both stability and potential for evolutionary improvements. Whether this is a case for biological networks should be still considered an open question, but some evidence to support this hypothesis exists (Shmulevich et al., 2005).

Due to their simplicity and elegance, random boolean networks were a subject of extensive research over the years and a number of variations on the original concept was proposed. For example, random networks with asynchronous state updates were proposed in which genes do not all update at the same time and in which next node to be updated is chosen randomly (Harvey and Bossomaier, 1997). Such networks become non deterministic and are no longer guaranteed to have a cycle attractor. A deterministic variant of asynchronously updating networks has also been developed (Gershenson, 2003). Other investigated variants include more complex state representations for the nodes (multi-state), addition of stochasticity and various approaches to generate topologies in order to obtain certain statistical properties (e.g., scale-free networks, Serra et al., 2004).

The approaches discussed so far either assume a predefined network topology (based on experimental data) or rely on generating networks with topological properties that are specified a priori (e.g., with a certain degree distribution), and then investigate their average properties. As such, they allow to infer only in a limited manner how regulatory networks evolved over time and how evolution shapes their global properties. This is especially important if one considers that we only have the ability to reconstruct small fragments of biological networks and it is possible that the global properties of the networks are different from those of their sub networks known from experimental data (Stumpf et al., 2005). Thus, to investigate how gene networks evolve, grow in complexity and obtain their properties, a different type of modelling approach is used, one in which regulatory networks are allowed to change over time under a simulated evolutionary process. The topologies of such networks can be encoded directly, as a connectivity matrix or a list (see, e.g., Azevedo et al., 2006 and Leclerc, 2008 for recent examples of this approach used to test biological hypotheses). However, a more realistic approach, the one that is also the focus of this work, relies on an indirect encoding of a network in a model of a biological genome. In such a case, the topology of a regulatory network depends on the relations between regulatory and protein-coding elements of a genome, mimicking the way topology of a biological network emerges from interactions between TFs and DNA.

An example of a work that lies in between the approach based on directly encoding the topology of a network and evolving artificial genomes is a system proposed by Reil (1999). In his model, genomes encoding GRNs are generated randomly. Each genome is represented by a sequence of randomly generated digits with a certain length. Every sequence of 0101 (arbitrarily chosen) that occurs in a genome indicates the promoter of a downstream gene, similarly to a TATA box found in eukaryotic genomes (see Section 1.3.2). The sequence of $N$ numbers that follows the promoter is

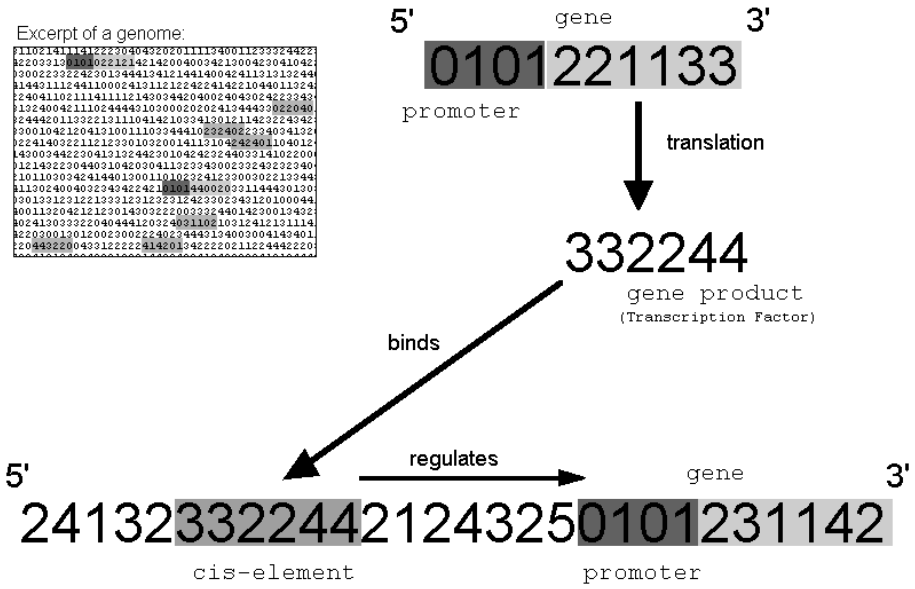



**Figure 2.1:** Reil's GRN encoding of a regulatory network in a string of digits (reproduced from Reil, 1999).

interpreted as a gene and can undergo expression ($N = 6$ in the original work). Each gene that undergoes expression produces a product that has a sequence associated with it, computed from the sequence of its coding gene, an abstraction of translation of an mRNA into protein. Translation was modelled as a simple incrementation of every digit. The topology of the network was determined by localizing all the regions in the genome that have a sequence matching to a given product and then forming a connection in the regulatory graph between this product and the product encoded downstream from the matching sequence (Fig. 2.1). This simple mechanism attempts to abstract the chemical affinity between proteins and specific DNA regions to which proteins can bind to promote or interfere with expression of downstream genes (sometimes dubbed as a "lock and key" mechanism). Whether a given product would increase or inhibit expression of a given gene would depend on a last digit in its sequence. Reil created random genome sequences and simulated boolean networks that were encoded in them. He observed network behaviour on the edge of chaos. He would also observe high robustness of such networks to perturbation of the state of single nodes. Since those networks were not a product of evolution, their robustness could be claimed to be a property of the chosen network encoding scheme.

A very similar model was more recently used by Quayle and Bullock (2006) to compare the topologies of networks encoded in the random genomes with those that were subjected to evolution. In the evolved networks, the authors observed degree distributions that were different from those that would occur in random networks, yet also different from scale-free networks (a property postulated for biological networks).

In his pioneering work, Jakobi (1995) proposed a more sophisticated mechanism to determine the topology of the network based on a match between gene products and regulatory regions on the genome. Virtual proteins were represented as circular

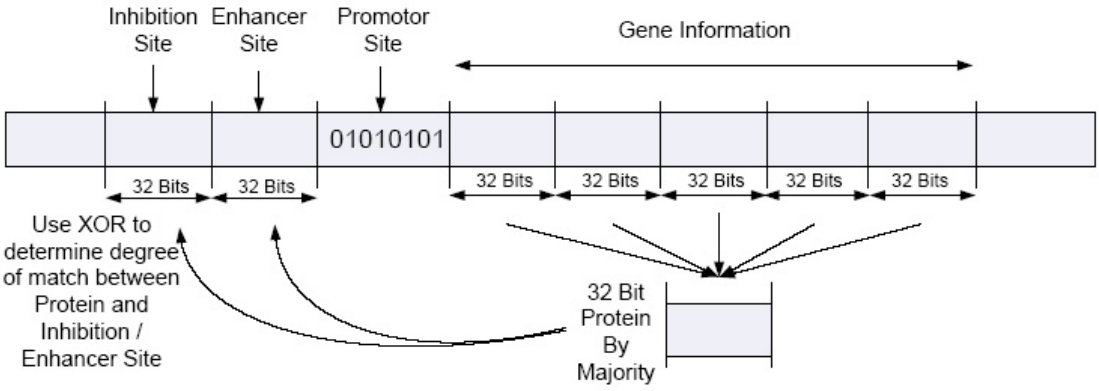



**Figure 2.2:** Banzhaf's GRN encoding of a regulatory network (reproduced from Kuo et al., 2004).

strings using a 4 letter alphabet. The affinity between proteins and regulatory regions was determined by a match between triplets of letters in proteins to the sequences of the genome, thus introducing a layer of indirectness mimicking the triplet based genetic code. In what was probably one of the earliest attempts both to apply artificial regulatory networks to solve problems and to control developmental process, Jakobi evolved networks that guided the divisions of cells that later formed a simple neural network that would later control a simulated robot.

Banzhaf (2003), in a manner analogous to Reil, investigated the properties of networks encoded in random genomes. However, instead of generating Boolean networks, he allowed for continuous changes in gene expression levels and encoded networks in a binary sequence. In his model, predefined 8-bit sequence would mark a promoter in the genome, whereas products would be defined by three 32-bit words that would follow, preceded by two words marking the binding sites to which other products could bind (one with an enhancing effect, the other with inhibiting). The affinity between products and binding sites could change gradually and was calculated by comparing sequence similarity of a product (32-bit number obtained from 160-bit gene) and a biding site (Fig. 2.2). Banzhaf analysed dynamics of continuously changing expression levels in randomly generated genomes (Fig. 2.3). He also investigated changes in network behaviour in response to random mutations in the genome.

The same model was further employed by Kuo et al. (2004) to evolve regulatory networks in which concentrations of certain proteins become stable or oscillate. Authors have also analysed the frequencies of occurrence of three and four node motifs in the evolved networks.

In yet another work based on the same model, Nicolau and Schoenauer (2009) have shown that it is possible to evolve networks that have specific topological properties (such as scale free degree distribution) by explicitly rewarding these properties in the fitness function. In a later work (Nicolau et al., 2010) the same model was applied to the classical reinforcement problem: balancing of the pole.

Flamm et al. (2007) created a very "low level" model of gene regulatory network and probably one of the most biologically realistic. The genes that were expressed

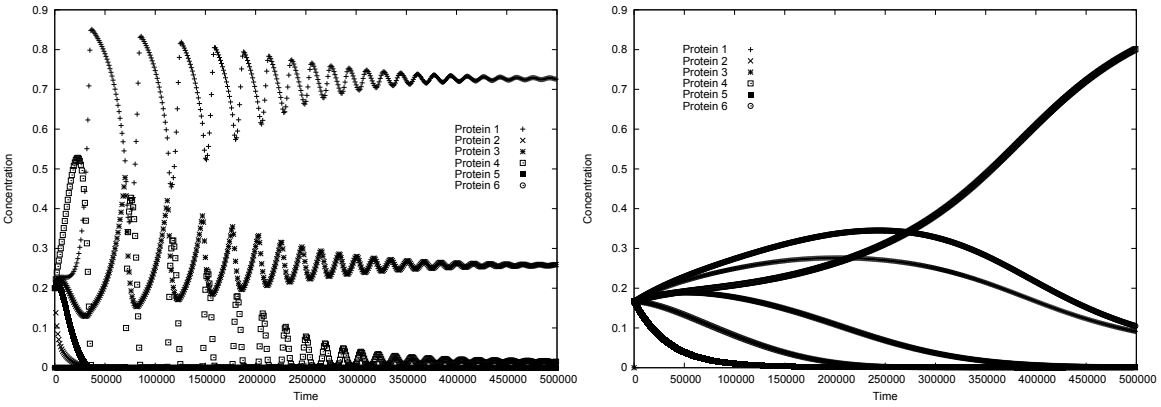



**Figure 2.3:** Example dynamics observed by Banzhaf in networks encoded by random genomes (reproduced from Banzhaf, 2003).

were converted to virtual mRNA molecules that would have a secondary structure (generated using an RNA folding algorithm used to predict the folding of real RNA). Such molecules would then bind to DNA with their exposed parts. Furthermore, the system would simulate cellular metabolism by allowing for chemical interactions between molecules to occur.

One of the most original and indirect methods to model affinity between proteins and DNA molecule in a GRN was proposed by Bentley (2003). The affinity was determined by calculating the similarity between square subsets of the Mandelbrot set, encoded in the interacting elements. Each promoter or genetic product would encode 3 real values determining the position in the Mandelbrot set and the size of its square subset (thus, the scale). Affinity was calculated by comparing the subsets after they were discretised. Bentley (2004a,b) employed the idea of "fractal proteins" to evolve gene regulatory networks that stabilize on certain levels of gene expression and to evolve simple robotic controllers.

Quick et al. (2003) proposed another genome and GRN model named BioSys and used virtual cells as simple control devices. The networks were evolved to control a virtual thermostat (a metaphor of cellular homoeostasis) and phototaxis of a simple robot. Concentrations of select transcription factors were interpreted as the output of a network and input signals were provided as externally driven concentrations of select factors. The model employed a genome represented as a string of 1s and 0s. Fixed size of the genome was assumed. Each transcription factor was encoded by 3 bits, thus allowing for existence of 8 types of virtual proteins. Each gene could be preceded by a fixed number of regulatory regions. Furthermore, a special part of the genome would contain constants such as decay rates for each protein type (via a look-up table).

BioSys model was later extended into xBioSys by Knabe et al. (2006). Genes were allowed to come in two types: one that was expressed only when it was promoted by the binding factors or one that was expressed by default (and could be further regulated). The type was determined by an additional bit field in every sequence encoding a gene. Furthermore, the genome encoded some of the properties global for the whole cell: degradation coefficients for each protein type, global affinity

constant (common for all proteins) and global saturation level for the proteins. This type of genetic encoding was subjected to evolution in order to obtain networks that act as biological clocks, i.e., genetic circuits in which concentrations of certain products oscillate, stimulated by an external, periodic signal. The ability to sustain oscillations without periodic signal and robustness to noisy signal was investigated. Knabe et al. (2006) observed a very high degree of tolerance. In a later work (Knabe et al., 2008a), the model was employed to determine whether networks of different functionalities have measurably different prevalence of structural motifs. Networks were evolved to perform either a single function or two functions (as a model of cell's ability to differentiate and to obtain networks that are modular). The results, however, did not show a measurable difference in motif distributions among the two types of networks.

Taylor (2004) employed a model of gene regulatory network very similar to the BioSys model to evolve real-time controller for a team of underwater robots. The GRN controller was successfully evolved to solve the task of robots grouping together.

## 2.2 Artificial embryogeny

The field of artificial embryogeny draws its inspirations from the process of multicellular development and focuses on modelling or abstracting it. Often, but not necessarily, in the scope of evolutionary computation. The interest in such approaches comes from the immense scalability of biological development which can drive growth of an organism that consists of trillions of cells. Development displays remarkable failure tolerance and robustness to perturbations. The ability to encode such tremendously complex structures in genomes that consist from only thousands of protein coding genes is an ultimate example of the power of indirect encoding. Nowhere in the genome locations of cells are specified and the organism unfolds through a process of mutual feedback between genetic control and the laws of physics. In result, small modifications to the timing of genetic events (heterochrony) or concentrations of growth factors can lead to certain parts of an organism changing in size, with all related structures (veins, nerves) adjusting accordingly. Such properties are one of the reasons of inherent evolvability and scalability of the developmental process, demonstrated by the history and the current diversity of life on Earth. Interestingly, forms morphologically such diverse as a whale, a bat and a mouse have a similar body plan and very similar genomes (see also p. 46). Even though superficially very different, their fins, wings and hands are homologous. The past two decades brought an increasing understanding of the relationships between genes, development and evolution, so it should come as no surprise that artificial developmental systems have been experiencing a surge of interest. From the engineering standpoint they are seen as one of the most promising ways to overcome the poor scalability of direct genotype-phenotype mappings in evolutionary computation.

In their comprehensive review of the state of the field, Stanley and Miikkulainen (2003) divide existing approaches into two main branches: grammatical and cell chemistry models. The first branch relies on a higher level abstraction of development and employs some algorithm that is used to iteratively grow a structure by incrementally adding new components. A classical example of such system are L-Systems, proposed more that 40 years ago by Lindenmayer (1968). In an L-system, rewriting rules allow to iteratively construct a shape, by subsequently replacing elements with their more detailed versions, yielding plant-like morphologies. In overall, evolving developmental systems based on grammatical approaches turned out to be a very fruitful avenue of research that allowed to create some of the most recognizable Artificial Life systems such as Sim's (1994) 3D swimming creatures or Framsticks by Komosinski and Ulatowski (1999), to name a few. One of the reasons of their high success is that their capture some of the essential elements of the development: modularity and reuse of genetic material, while still having very small computational cost.

However, it is the other branch of artificial embryogenesis systems that is the focus of the interest of this thesis: the cell chemistry systems. The inspiration for this approach dates to the seminal paper by Alan Turing (1952) in which he introduced a reaction-diffusion model. The model showed how local interactions between uniformly distributed and reacting substances can, by amplifying random fluctuations, lead to the emergence of patterns that are similar to those found on shells and in animal coatings. Although proposed for chemical reactions, it is general in a sense that it applies to emergence of a pattern via local rules with higher level entities, such as cells. Cell chemistry based models of artificial embryogenesis try to tap into this emergent complexity coming from local interactions: cells in a developing embryo communicate using chemical signals (morphogens) that diffuse from cell to cell or on a grid and this, together with internal rules governing the behaviour of each cell, can lead to self organization. Cell chemistry models themselves cover a broad range of approaches that differ by their choice of interacting elements (cells of various shapes, cells on a grid, blocks) and the method used to control the behaviour of the elements. Since this thesis focuses on the gene regulatory network based approaches, a following short review discusses developmental systems in which basic entities (typically, cells) are driven by GRN inspired controllers. In such systems, each cell is a separate entity and can take actions such as division or differentiation. Each cell is controlled by a copy of the same gene regulatory network, and what allows cells to take different actions are differences in the state of each cell, determined by the concentrations of TFs.

Eggenberger Hotz (1997), in his pioneering work, which was one of the main inspirations for this thesis, proposed a 2D/3D developmental system with cells driven by GRN. In his model, each product and promoter has an associated real number. The affinity between product and promoter is defined by the difference between the associated numbers, whereas the genome itself consists of groups of product coding genes preceded by regulatory areas. Artificial cells divide on a grid and are able to emit morphogens: signalling molecules that influence actions of other cells. This

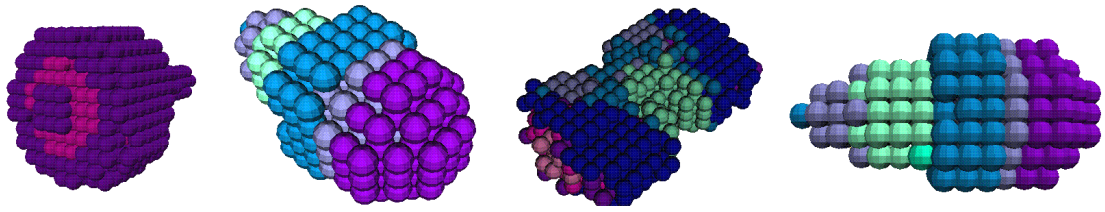


**Figure 2.4:** 3D embryos evolved for bilateral symmetry (reproduced from Eggenberger Hotz, 1997).

allowed him to evolve genomes generating planar patterns that resemble those found on butterfly wings. In the 3D version of his model (Eggenberger Hotz, 2004), he evolved simple elongated shapes using symmetry as a fitness function (Fig. 2.4). He also emphasized the importance of the interplay between the genome and the physics and allowed the cells to flex the grid using their adhesion (Eggenberger Hotz, 2003b). One of the more interesting features of this model was the emergence of regenerative capabilities allowing the embryos to regrow their removed parts (Eggenberger Hotz, 2003a).

Bongard (2002) employed a GRN model similar to that of Reil (1999) (see p. 53 of this thesis) to grow bodies of virtual organisms. Instead of GRN controlling the cells, Bongard used higher level modules: in his model, the bodies consist of cylindrical segments connected by actuating joints. Apart from controlling the growth of the segments, regulatory network encodes the structure of simple neural network. After the growth phase is finished, the neural network takes control over the organism. Initially, Bongard was able to obtain only very simple forms of locomotion. In a later work using similar approach (Bongard and Pfeifer, 2003), the authors evolved morphologies that consisted of spheres of various size. They were able to evolve animats that were capable of efficient locomotion and performing simple tasks such as block pushing.

Kumar and Bentley (2003) devised another early model of GRN-driven developmental process, named EDS (evolutionary developmental system). Their model of the genome consists of two parts. One encodes the properties of 8 types of proteins that exist in the system (synthesis and degradation rate, diffusibility). The other is a list of genes, each consisting of cis-region (regulatory) and protein coding region. Each gene encodes a single protein and can contain multiple cis elements, so that the expression of a given gene can be controlled by multiple proteins. Depending on the concentrations of the proteins, cells divide or differentiate (change colour). Each cell contains multiple receptors on its surface through which it can detect morphogens diffusing from other cells. The direction of cell division is determined by the axis of division which is allowed to take a couple of discrete orientations in relation to the grid. Using a genetic algorithm, the authors evolved simple spherical or cubical forms.

Knabe et al. (2008b) used their earlier devised GRN model, xBioSys (Knabe et al., 2006, see also p. 56 of this thesis), to evolve developmental process on a two dimensional grid. To represent a multicellular embryo, the authors employed

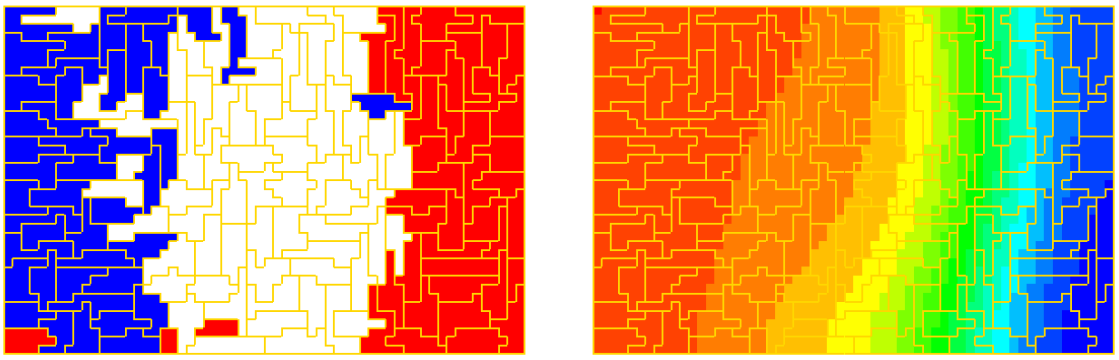


**Figure 2.5:** Self organized French flag pattern (left) and concentration of an underlying morphogen gradient (right) (reproduced from Knabe et al., 2008b).

the Cellular Pots Model (Glazier and Graner, 1993), a cellular automaton in which each embryo cell consists of multiple cells on the grid, bounded by a virtual cell membrane. Cells can secrete morphogens that diffuse on the grid and can be sensed by other cells as an average concentration from all of the grid cells that belong to this cell. Authors employed their system to evolve French flag embryos, i.e., the embryos in which cells form three differentiated regions on a rectangular area, each cell type represented with a different colour (Fig. 2.5). The structure of the embryos would self organize without any maternal gradients, thanks to the intercellular communication, generating asymmetric morphogen gradient. Authors have also investigated how an already evolved network can be evolutionarily adapted to a slightly modified target pattern with different stripe widths in just a few generations.

Schramm et al. created a developmental model (Schramm et al., 2011; Schramm and Sendhoff, 2011) with a GRN inspired by the work of Eggenberger Hotz (1997) to simulate development of a 2D multicellular organisms that are later evaluated for their ability to swim in a fluid-like environment. All cells in a developed embryo are connected by virtual springs and the cells on the outline of the body are able to modify the resting length of the two springs that connect them with neighbouring external cells. In one of the works (Schramm et al., 2011), the resting length oscillated according to the globally set frequency and with phase shift that was subjected to evolution. However, the controller that defined how cells contract was encoded in a separate chromosome, different from the one controlling the developmental process. The fitness function was a combination of a reward received for the elongation of the shape and the distance the virtual organism was able to travel in the simulated fluid. The authors obtained elongated organisms which would move using realistically looking undulating movements. In a later work (Schramm and Sendhoff, 2011), cells were allowed to differentiate into pattern generating neurons. In a related publications, the authors have also investigated evolved regulatory networks for the presence of motifs that are overrepresented (Schramm et al., 2010), but without conclusive results.

A very interesting result from the point of view of the complexity of generated morphologies was obtained by Hogeweg (1999, 2000). Her model of an organism is based on the Cellular Pots Model (as is the later work by Knabe et al., 2008b).

**Figure 2.6:** Example development of a 2D embryo in the Cellular Pots Model (reproduced from Hogeweg, 1999).

The genome encodes a Boolean network and, interestingly, the fitness function is not based on obtaining certain morphology but on maximising the distance between attractors reached by GRN in every cell, in a spirit of Kauffman's interpretation that cellular differentiation is an effect of cells reaching different attractors. This, by itself, led to the emergence of many processes important in biological development such as cellular re-differentiation, migration, bulging or cell death.

Recently, Andersen et al. (2009) investigated the robustness of virtual embryos to the removal of large fractions of cells. They used their own model of gene regulatory network with morphogens diffusing to neighbouring cells. The developmental process took place on 3D grid. The fitness function was based on a similarity of the obtained embryo to the target shapes: hollow sphere (50 cells), cuboid (128 cells) and a hollow cube (74 cells). Obtained embryos were observed to have a high degree of capability of self-repair after cell ablation, even though they were not selected for it during evolution.

Beurier et al. (2006) proposed a controller that was encoded in a manner more loosely inspired by gene regulatory. The genome would consist of regulatory elements, homeotic, behavioural and segmenting. It was demonstrated to evolve development of French and Japanese flags on a uniform grid.

In a more recent application of Banzhaf's model (2003), Chavoya et al. (2010) used it to evolve three dimensional, cubic embryos that differentiated into three layers on a 3D grid, hence creating his extension of the French flag problem into the 3rd dimension (Fig. 2.7). He would then analyse how patterns of gene expression changed over time during development.

In a novel and very interesting application of an artificial developmental system

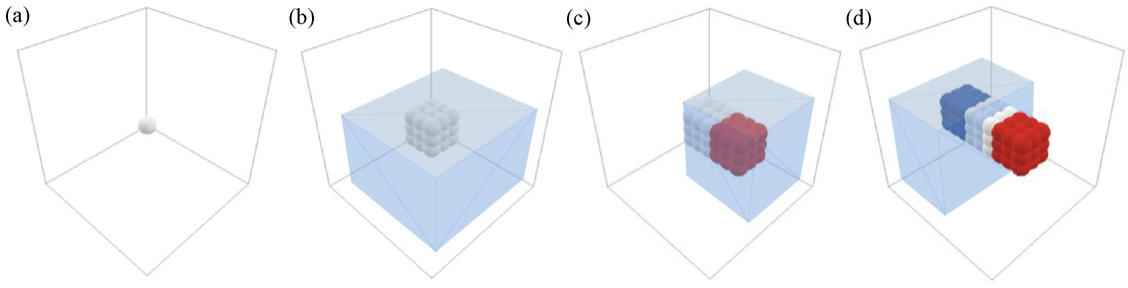



**Figure 2.7:** Subsequent stages of development of 3D "French flag" (reproduced from Chavoya et al., 2010).

controlled by a GRN, Trefzer et al. (2010) evolved genomes that controlled the developmental process that results in formation of a desired image, effectively treating the GRN and development as an image compression method. They were able to achieve compression ratios comparable and often higher than that of the JPEG algorithm.

# Chapter 3

# The model of GRN and evolution

This chapter introduces the model of the artificial genome, the artificial gene regulatory network (GRN) and the genetic algorithm that is used in this thesis to simulate the evolution of genetic control and multicellular development. The genome indirectly encodes the topology of the GRN. The decoded GRN represents the graph of interactions between virtual TFs and is used to simulate the behaviour of an artificial cell. It is however the artificial genome that is subjected to evolution.

## 3.1 Genome

The model of the genome and its evolution employed in this thesis was designed to capture the most essential features of evolving, biological genomes. The artificial genome is defined as a sequence of genetic "elements", which can function either as regulatory regions or encode products that will be produced by a virtual cell. Each genetic element consists of $N$ real numbers, hence the genome can be represented as a vector of size $cN$, where $c$ is the number of genetic elements.

The genome encodes an artificial gene regulatory network: a network of interactions between genes. Each vertex of gene regulatory network (GRN) represents a product (or a set of) that can exist with some concentration in a cell. The edge between vertices $A$ and $B$ means that gene $A$ regulates gene $B$, that is, the concentration of $A$ influences (positively or negatively) the synthesis rate of $B$. The actual connectivity of the network is encoded indirectly and is decoded by determining the affinities between elements encoded in the genome. The topology of the network remains static during the lifetime of a cell and is decoded from the genome before the GRN simulation is started. An important feature of the encoding employed in this work is that neither the size of the genome, nor the size and the topology of the network are fixed. Both the number of vertices and the number of distinct types of products (i.e., types of TFs having different binding properties) that cells can produce is limited only by the size of the genome itself and subjected to evolution.

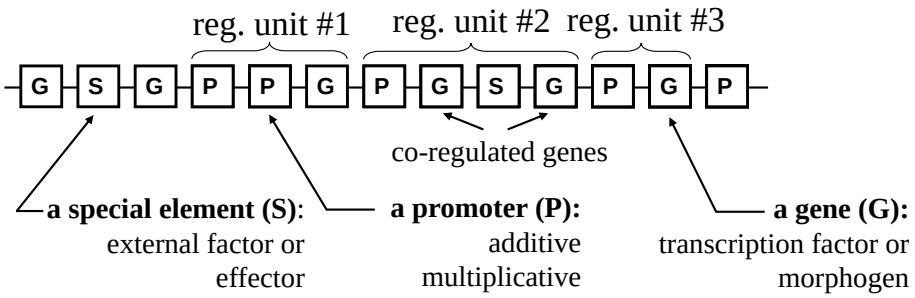



**Figure 3.1:** Overview of genome structure. A series of genetic elements marked as promoters (P) followed by products (G) forms regulatory units. Special elements (S) encode input and output nodes of the network, but do not belong to regulatory units.

### 3.1.1 Overall structure

Genetic elements that form the genome fall into three classes: "products" (encoding virtual proteins), regulatory elements (or "promoters", regions to which virtual proteins bind) and "special" elements, which are used to encode inputs and outputs of the network. The core assumption is that products can have certain affinity to promoters and will bind to them with an effect dependent on their concentration. However, the binding of TFs to regulatory elements is not simulated explicitly and is abstracted as weights in the regulatory graph.

Products represent the abstraction of DNA sequences coding for proteins that can regulate expression of other genes (TFs). Although proteins in a biological organism play a role in processes other than gene regulation, in the model presented here, with the exception of elements belonging to the class "special", all products of the genome serve a regulatory purpose.

In a version of the model that allows for multicellular development (Chapter 6, p. 115) a second type of product is also used: a morphogen. Morphogens in the system behave in a manner analogous to ordinary TFs, but they diffuse outside the cell in which they are produced and can bind to regulatory elements of other cells, hence allowing cells to react to the presence of other cells or even to communicate.

Fig. 3.1 provides an overview of the genome structure. To determine the topology of a regulatory graph, the genome is parsed sequentially and a "regulatory unit" (a vertex in the graph) is created whenever a contiguous series of $P$ elements (promoters) is followed by a contiguous series of $G$ elements (genes encoding products). "Special" elements ($S$) are ignored during the initial scan for regulatory units and can be freely interspersed among products and promoters: they are connected to the network at a later step. Since one regulatory unit of the GRN can be composed of multiple promoters and multiple products, any two nodes in the graph can be connected by several edges. Products that happen to exist at the very beginning of the genome and promoters at its very end become non-active. Furthermore, each regulatory unit occurs directly after the previous one so, apart from "special" elements, there can be no other elements between them. However, many non-functional elements can exist in the genome: not all regulatory units have to connect to the rest of the network. The presence of genome fragments that do not perform any function

**Table 3.1:** The classes and types of genetic elements defined in the model.

| Class | Type | Description |
|---|---|---|
| Promoters | Additive<br>Multiplicative | Inputs of regulatory units, their effects sum up or multiply, respectively (see Eq. 3.4) |
| Genes | Transcription factor<br>Morphogen | Influences promoters inside the same cell<br>Diffuses from the cell and influences promoters of other cells |
| Special | External factor<br>Effector | Input for the GRN (signals from the environment)<br>Output of the GRN |

is considered to be advantageous from the point of view of evolvability, since such regions are free from selection pressure and can accumulate mutations over time. Explicit "pseudogene" elements can also be easily added to the system, so that a product or promoter can, by a simple mutation of its type, become inactive. Such elements were used in a paper based on the model discussed here (Joachimczak and Wróbel, 2009).

### 3.1.2 Genetic elements and affinity

All genetic elements share the same structure, shown in Fig. 3.2. The "type" field defines to which of the three classes of elements it belongs. It is simply an integer value indicating the type of an element. More than a single type can belong to a given class (e.g., different types of promoters can exist). The actual set of types of elements depends on the experiment (for example, there is no use for morphogens in experiments with single cells and not all types of regulatory elements are necessary). Table 3.1 sums up all the types of elements employed in the system.

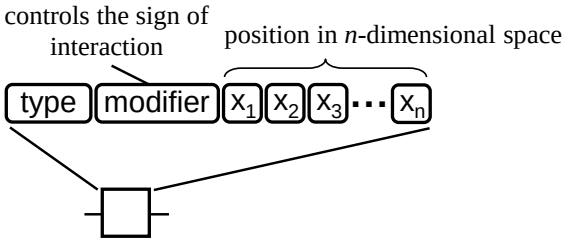



**Figure 3.2:** Internal structure of a genetic element. In this thesis, the associated position was always in 2D space.

The essential part of each element is a sequence of $n$ real numbers that represents coordinates in $\mathbb{R}^n$ space. The position of the point associated with an element can be understood as an abstraction of 3D structure of a protein or nucleic acid and is used to calculate the affinity between products and regulatory elements (i.e., the weights of connections in the GRN). In particular, the degree of affinity between a product G and a promoter site P is a function of the Euclidean distance between the points in $\mathbb{R}^n$ space they encode. A distance of zero between the two points creates a GRN edge with the maximum allowed weight (set to 10), while a greater distance results in a smaller weight according to a decreasing exponential function. To prevent full

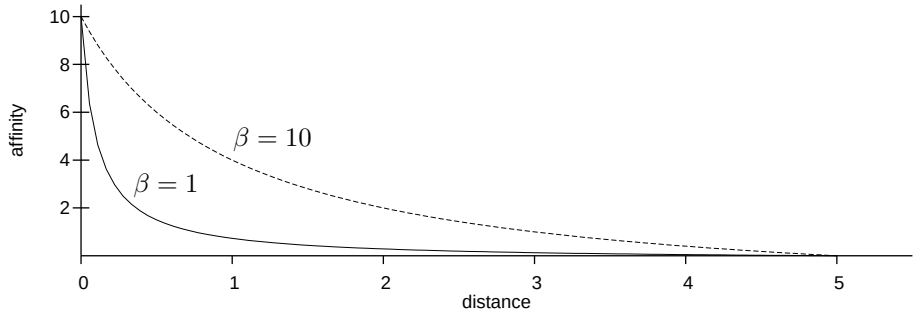



**Figure 3.3:** Two examples of different deflections of the affinity curve used to convert distance between points in $\mathbb{R}^2$ associated with genetic elements into the weights in regulatory graph. Depending on experiment, $\beta = 1$ or $\beta = 10$ was used.

connectivity in the regulatory graph, a cut off value of 5 is used. The maximum distance of interaction is an element of biological realism. Without it, elements far away from each other could, in principle, still interact. Also, its introduction allows to reduce the computation cost, as otherwise all possible connections in the GRN would exist. Furthermore, a maximum value for affinity prevents biologically unrealistic effects where very small concentrations of modelled substances could have huge effects. The absolute level of the affinity between the two elements G and P is calculated as follows

$$w_{P,G} = \begin{cases} d_{P,G} \leq 5 : \beta \cdot \frac{2(5-d_{P,G})}{10 d_{P,G}+\beta} \\ d_{P,G} > 5 : 0 \end{cases} \tag{3.1}$$

where $d_{P,G}$ is the Euclidean distance between the points associated with the elements P and G, $\beta$ is a parameter of the system and allows to control how quickly the weight diminishes with the increasing distance (see Fig. 3.3). The"modifier" fields in the two interacting genetic elements are used to determine whether the connection has positive or negative weight. The "modifier" can be either 1 or -1 and the sign of the weight is equal to the product of the modifiers (i.e., it is negative if the values differ).

The dimensionality $n$ of the space of coordinates is a property of the whole system and in all of the evolutionary experiments presented in this work $n = 2$ was used. This allows to visualize all genetic elements and the paths they follow during evolution on a plane. A higher number of dimensions allows for more neutral mutations to coordinates of elements and neutrality of the search space has frequently been postulated as important for evolvability (Galván-López and Poli, 2006; Shipman et al., 2000). However, more dimensions also result in an increased search space. The influence of the dimensionality of genetic elements on evolvability in the presented model was analysed in depth elsewhere (Joachimczak and Wróbel, 2008b, 2009). In brief, no advantage nor disadvantage of a higher number of dimensions was detectable under typical experimental conditions. However, if the evolutionary algorithm was severely handicapped by removing more complex types of mutations, the additional dimensions were detectably detrimental.

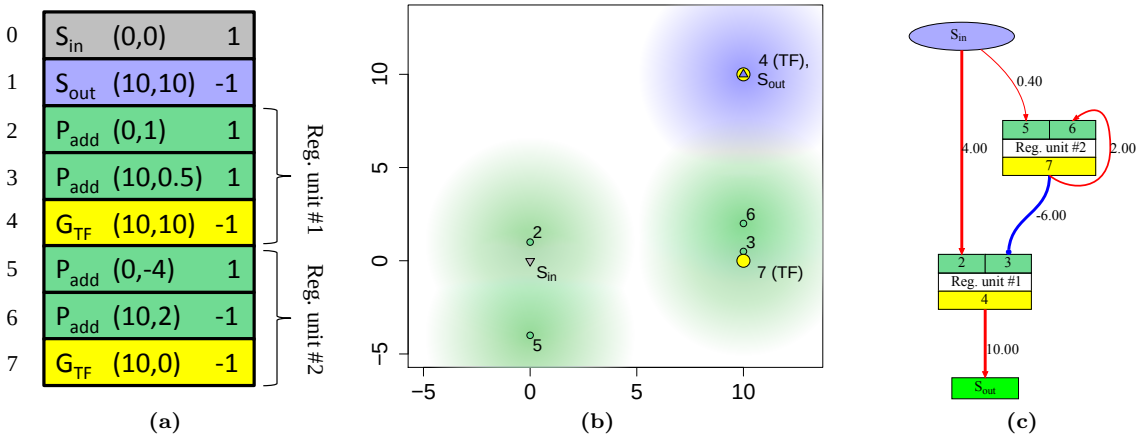



**Figure 3.4:** The encoding of the regulatory network in a linear genome. (a) example of simple genome consisting of 8 genetic elements, elements carry coordinates in $\mathbb{R}^2$. (b) representation of genetic elements on the surface with the areas of connectivity of promoters. (c) the resulting topology of the regulatory network.

## 3.2 Artificial Gene Regulatory Network

The artificial regulatory network is represented as a multidigraph, in which each node represents a regulatory unit and each edge has an associated weight and a promoter which it regulates (which can be either additive or multiplicative). After the connectivity has been determined, it remains static during the simulated life of a cell (and shared among cells in multicellular embryos), and only product concentrations change over time.

Figure 3.4 illustrates the process of creation of the regulatory graph from a sequence of genetic elements. In the first step (Fig. 3.4a) regulatory units are identified. Then, the affinities between products and promoters are computed based on locations of their associated points in $\mathbb{R}^n$ (Eq. 3.1). If a gene encoding a product belonging to a given regulatory unit is found inside the interaction distance of a promoter in another (or the same) regulatory unit (Fig. 3.4b), an edge is added with a weight equal to the computed affinity. Finally, special elements (encoding inputs and outputs, $S_{in}$ and $S_{out}$ in Fig. 3.4) are connected to the regulatory network after regulatory units have been identified. The complete algorithm is provided in the form of a pseudo code in the Appendix (Listing 1, p. 178).

External factors (inputs) act on the network just like any other TF does. That is, they form connections to regulatory units if they have some affinity to their promoters. The only difference is that during simulation of a cell, the concentration of an external factor is driven externally and the network cannot influence it. Effectors (outputs) act as special transcription factors whose concentration can be read externally from the network (e.g., to signal cellular division). The concentration of this special TF is controlled as it was a product of a special regulatory unit with a single additive promoter whose coordinates are taken from the genetic element that encodes this particular effector. By design decision, the concentration of a cellular effector cannot influence other regulatory units in the regulatory network, nor it can

be influenced directly by an external factor.

For experiments where multiple different external factors and effectors are defined, the exact function of each special genetic element (i.e., an element having the type field equal to "effector" or "external factor") depends on its order of occurrence in the genome. For example, in an experiment in which 2 effectors are defined: cell division and cell apoptosis (death), the first genetic element of type "effector" will be interpreted as a signal for division, the second as a signal for apoptosis, and any further such element will simply be ignored and remain non-functional.

During the simulation of a cell, the concentration associated with each node in the regulatory graph is synchronously updated in discrete time steps. Since a single regulatory unit can contain multiple products, all have the same concentration. Such products can be interpreted either as multiple proteins with identical concentrations or as a single transcription factor with multiple DNA-binding domains.

The change of the concentration $L$ during a single simulation time step is calculated as a sum of sigmoidal term and degradation term, as follows (the units are omitted):

$$\Delta L = (\frac{1 - e^{-A}}{1 + e^{-A}} - L)\Delta t \tag{3.2}$$

where $\Delta t$ is an integration time step in the Euler method, $L$ is the current concentration (restricted to the interval $[0, 1]$), and $A$ is the activation level of the regulatory unit (Eq. 3.4). The sigmoidal term above is equal to the hyperbolic tangent ($\tanh \frac{A}{2}$). For the experiments discussed in this work, $\Delta t$ equal to 0.05 or 0.1 was used.

To calculate the activation level $A$ of a regulatory unit, the activity of all its promoters is first computed:

$$p_i = \sum_{k=1}^{K} L_k w_{k,i} \,. \tag{3.3}$$

where $p_i$ is the activity of a given promoter, $K$ is the total number of binding factors in the genome, $L_k$ denotes the perceived level of the factor $k$, and $w_{k,i}$ is the chemical affinity (weight of the connection in the regulatory graph) between the factor $L_k$ and the promoter $p_i$. From that, the activation of the whole regulatory unit is calculated:

$$A = \prod_{i=0}^{I} p_{m,i} \sum_{j=1}^{J} p_{a,j} \tag{3.4}$$

where $I$ and $J$ denote the number of multiplicative and additive promoters (respectively), while $p_{m,1..i}$ and $p_{a,1..j}$ describe their activations (Eq. 3.3). The presence of a multiplicative promoter in a regulatory unit results in a strict requirement for the presence of a product with an affinity to it, otherwise the unit is not expressed. The "all-or-nothing" regulation is quite common in biological systems and is difficult to incorporate when only additive units are used (such as in classical perceptron neural networks). The actual indexing of promoters starts from 1, $p_{m,0}$ is reserved for the identity element of multiplication ($p_{m,0} = 1$). The pseudo code of the algorithm

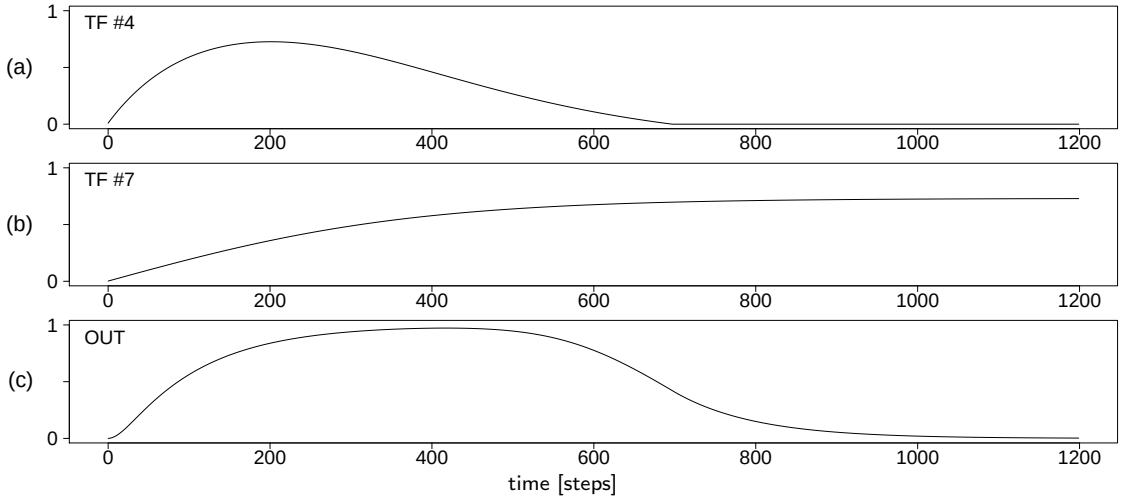



**Figure 3.5:** Simulated concentrations of transcription factors over time (ab) and the concentration of output cellular effector $S_{out}$ (c) in the example network (Fig. 3.4) during 1200 simulation steps.

used to update the GRN is provided in the Appendix (Listing 2, p. 179).

The functionality of the example network seen in Fig. 3.4c can be easily inferred from its structure. The external input signal $S_{in}$ set to the constant maximum level of "1" is weakly connected to regulatory unit 2 and strongly connected to regulatory unit 1. When the simulation starts, $S_{in}$ will cause the transcription factor encoded by element #4 to be highly expressed and to start quickly accumulating over time (Fig. 3.5a). At the same time, its weak connection to regulatory unit 2 will cause transcription factor #7 to be expressed at a non zero level (Fig. 3.5b). Since this TF binds back to the second promoter and regulates its own expression, a positive feedback loop will ultimately cause it to reach high concentration. Since TF #7 downregulates expression of TF #4, the latter will at some point start to degrade, until it is no longer expressed, resulting in a "single pulse" pattern. A similar pattern will be repeated by the output node, which is directly regulated by the concentration of TF #4 (Fig. 3.5c).

## 3.3 Genetic algorithm

All experiments with simulated evolution presented in this thesis employ a version of a genetic algorithm. For real valued genome representations, often a more formally defined method is used, the evolution strategy (Beyer and Schwefel, 2002), with an important feature of self-adapting mutation pressure. Similarities exist in the way coordinates in genetic elements are mutated in the presented model and in the evolution strategy. However, since the goal of the presented model was to create a plausible model of evolving genomes, full control over what kind of mutations exist in the system was considered essential, hence the use of a more openly defined concept of a genetic algorithm.

The genetic algorithm employed in this thesis uses a fixed population size (either 100 or 300 individuals). If elitism is enabled, a new generation is created by copying

a few best individuals (typically 5) without mutation. Remaining individuals are created by mutating selected individuals in the current generation (asexual reproduction). If sexual reproduction is enabled, a fixed number of individuals will be created through multi-point crossover between two genomes of (usually) different sizes and with additional mutations added. Sexual crossover for diverse population is a highly deleterious mutational operator, and so, no more than 50% of the population would be created sexually.

Although elitism protects best genomes found so far, if neutral mutations occur in an elite genome, it will be replaced by its mutated version. In this way, even without fitness improvement, elite genomes can accumulate mutations over generations, wandering through the neutral space of solutions (hence, elitism protects the phenotypes, not genotypes). The random wander through neutral regions in search space is postulated to improve evolvability, as it allows to discover paths to new unexplored areas in fitness landscapes (Galván-López and Poli, 2006; Shipman et al., 2000).

### 3.3.1 Initialization

An initial generation is formed from a population of randomly created individuals. Each genome is created by inserting elements representing all possible effectors and external factors at the beginning of the genome and a certain number of regulatory units (from 1 to 15 in this thesis). Each unit consists of a single promoter and product or a small, randomized number of each. The coordinates associated with genetic elements are randomized by moving from the point $(0, 0)$ by a distance drawn from the uniform distribution and in a random direction. A number of special elements with a type set to "effector" equal to the number of defined cellular effectors in the experiment is placed at the beginning of the genome, so each possible output has one node associated with it (although it may still be disconnected from the network). The same is done for all possible external factors (input nodes).

### 3.3.2 Selection

To select candidate genomes for mutation and crossover, tournament selection is used (see, e.g., Mitchell, 1998). Tournament selection operates by choosing $k$ individuals from the population randomly and then choosing $n$-th best individual among them with a probability $p(1-p)^{n-1}$ (with the best one having a chance of $p$ to be selected). Larger tournament sizes and lower values of $p$ decrease selection strength. In the simplest scenario used in most of presented experiments, a binary selection was used ($k = 2$, $p = 1$), meaning that a pair of individuals was always chosen randomly and the best of the two would enter new generation after mutation or crossover.

One of the reasons to use tournament selection instead of fitness proportional methods, such as roulette-wheel selection, is that it does not require any additional fitness scaling to compensate for the non-linearity of the fitness function. To prevent rapid loss of initial genetic diversity, it is also possible to set selection probability

**Table 3.2:** Summary of genetic operators implemented in the system. Depending on the experiment, a subset of these would be chosen and probability of each would be set independently.

| Scope | Operator | Description |
|---|---|---|
| Element | Position change | moves the element-associated point in a random direction by a distance drawn from a normal distribution |
| | Type change | randomly changes the type of element, probabilities of occurrence of each type are set on the level of experiment |
| | Modifier sign change | switches all the interactions of this element from inhibition to activation or vice versa |
| | Insertion | creates and inserts a new random element at random position |
| Genome | Deletion | removes a group of genetic elements |
| | Duplication | copies and inserts a group of genetic elements at random position |
| | Recombination | multipoint crossover between two individuals |

to a lower initial value and increase it over time, a method used in some of the discussed experiments.

### 3.3.3 Genetic operators

Genetic operators are designed to work on the level of genetic elements rather than single bits or real numbers, because a genetic element is the basic unit of heredity in the presented model. Table 3.2 lists the genetic operators defined in the system. Whenever a genome was selected to be passed to a new generation genetic operators were applied with probabilities specific to a particular evolutionary run or a set of runs. Depending on the scope of an operator, the probability of a mutation would be defined at the level of a whole genome or at the level of a single genetic element (i.e., each genome or genetic element in every genome would mutate with a certain probability).

#### Duplications and deletions

Duplications and deletions were performed by selecting a group of elements from a random position in the genome and either copying it to a random location (duplication) or removing it (deletion). The probability for duplications and deletions was controlled separately (although usually the same probability was used). The length of a selected group of elements was drawn from the geometric distribution. Depending on the experiment, two approaches were used to decide whether to activate a genetic operator. In the first approach, the probability of each type of event occurring was defined for each genome. A random number was drawn from the range [0,1] and if it was lower than the defined mutation probability $p$, duplication (or deletion) was applied. The number was then drawn again, so that multiple events could occur. The process was however stopped as soon as the drawn number was higher than $p$.

In the second approach, the number of mutation events occurring per single genome was directly proportional to its size, i.e., on average, for a genome two times larger, the number of duplications or deletions would be two times higher. However,

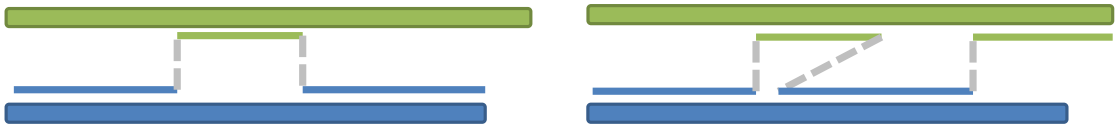


**Figure 3.6:** Sexual recombination between two genomes of different size: two examples of possible scenarios of multipoint crossover between parents using the algorithm provided in the Listing 3, p. 180.

in both approaches, the average number of duplications or deletions occurring per each mutated genome was kept below 1. The latter approach was used only in the experiments discussed in Chapter 5.

#### Recombination

To allow for sexual recombination between genomes of potentially different lengths, a multi point crossover was used to create a single individual from the two parents (see Fig. 3.6). For each of the two parents, a pointer to the current genetic element was initialized to point to the first element. Then, the new genome was created by repeatedly fetching genetic elements from the currently selected parent, with a small probability of switching to the other parent. The pointer to the active element would then be incremented for either both of the parents or just the one used to supply previous genetic element. The Listing 3 on p. 180 of the Appendix provides the pseudo code of the full algorithm.

### 3.3.4 Viability criteria

In many of the evolutionary runs, "viability" criteria were defined, with the goal of assisting the search algorithm by cutting off access to irrelevant areas of the search space. Non-viable phenotypes would not take part in the creation of a new generation and thus be removed from the gene pool immediately. For example, in some experiments in multicellular development (Chapter 6), a phenotype is defined as viable only if it has at least two cells at the end of development. This prevents genomes that lost the capability to divide from passing their damaged genotypes to a new generation.

If enabled, viability criteria would also be used during the creation of an initial population. In such a case, genomes were repeatedly generated randomly, until a viable individual was found. This however puts a limit on the acceptable strength of viability criteria, as it must be possible to create a viable individual randomly in a reasonable number of attempts. Multiple retries were allowed, so that for a more demanding criteria it could take up to a few thousands of randomly generated genomes before a viable individual was found. This process was however many times faster than typical evaluation due to various optimizations that were enabled. For example, if the cell division effector is unconnected in a regulatory graph, it is already known before simulating the cell behaviour, that the phenotype will never have more than one cell.

## 3.4 Summary

The simulations described in this thesis are based on a new model of biologically inspired artificial genome indirectly encoding the structure of gene regulatory network. The approach presented in this thesis allows linear genomes to encode networks of arbitrary size and topology. Regulatory units can have an unconstrained number of binding sites and can encode arbitrary number of products. The number of types of TFs (or morphogens) is also unconstrained. The affinity between product and promoter is calculated as a function of Euclidean distance between points in $\mathbb{R}^n$ associated with each of them.

The concentrations of TFs are stored as real values constrained to $[0, 1]$ and change over time continuously by increasing their synthesis rate or actively degrading them. TFs are also subjected to spontaneous exponential degradation. The process of TFs binding to promoters is not simulated explicitly. Instead, a genome is decoded into a regulatory graph (GRN) whose topology remains static during the lifetime of a cell. Regulatory influence (affinity) is represented as a weight of an edge in this graph. During the simulation, the concentrations of TFs (represented as nodes in the graph) are updated, but the weights in the regulatory network remain fixed.

Products binding to a regulatory region can work either additively to increase the rate of production or multiplicatively. In the latter mode, a presence of a product binding to a multiplicative promoter is necessary to activate the synthesis and amplifies it proportionally to the activation of such promoter. Genomes are evolved using biologically inspired mutational operators: duplications and deletions of their fragments, and by introducing small changes to the coordinates stored in each genetic element, thus allowing for smooth modifications of the affinities between products and promoters.

In the following chapters, evolvability and applicability of the introduced model of gene regulatory network to computational tasks and to control problems will be evaluated. Finally, the model is applied to control artificial embryogenesis in 3D, where starting from a single cell, a multicellular morphology emerges through subsequent cell divisions, with all cells sharing the same genome (and thus GRN).

# Chapter 4

# Processing signals with regulatory networks

In this chapter, the evolvability of the introduced artificial GRN model is investigated. The evolutionary algorithm is challenged with the task of finding genomes capable of performing computational tasks of increasing difficulty.

Biological GRNs can be thought of as life's computers, organizing all processes that occur inside every cell and performing computation long before the evolution of the nervous system. They are well known for their robustness to external interference and to damage caused by mutations. Hence, the evolved properties of such networks and their applicability for control of artificial and synthetic systems are of great interest for both the Artificial Life and the Systems/Synthetic Biology research community. On one hand, artificial GRNs can be considered a promising computational model and an alternative to, e.g., neural networks. This stimulates the interest in exploration of evolvability of various genome encodings and trade-offs between biological realism and computational efficiency (see section 2.1, p. 51) On the other hand, there is a growing interest in the design of synthetic (that is "wet") regulatory networks that are capable of computation. Synthetic GRNs in which gene expression oscillates with a desired period or which can count subsequent external signals (Elowitz and Leibler, 2000; Friedland et al., 2009) have recently been constructed and represent important milestones in the rapidly developing field of synthetic biology. The hope is that synthetic networks engineered to produce proteins or mRNAs in a desired and intelligent manner will soon find their use for therapeutic and industrial purposes.

The results discussed in this chapter demonstrate how a genetic algorithm (GA) can be used to evolve artificial gene regulatory networks in which expression of certain genes follows a desired target pattern. In most of the experiments, the target pattern depends on external stimuli to the network, requiring either a certain response or continuous computation performed on the external signal. From a biological point of view, such input can be understood as a concentration of a chemical substance in the environment.

## 4.1 Experimental setup

The training set was defined as a set of input concentration patterns that were presented to each individual (genome decoded into GRN) at each evaluation during the GA, paired with the target concentration pattern. The networks were simulated for a predefined period of time and the fitness function would compare the pattern generated by the networks with the desired response. If the training set consisted of multiple input/output pairs, the networks were reset to their initial state before they were stimulated with a next input pattern. Unless specified otherwise, artificial evolution was always repeated (using different random seed) in 10 independent runs and the behaviour of the best network obtained in all of the 10 runs is presented overlaid on the training set (i.e., input and desired output pairs). Furthermore, the quality of the obtained solutions and the level of generalization was investigated for obtained individuals.

### 4.1.1 Fitness function

The goal of simulated evolution was to obtain networks that generate desired expression dynamics in response to particular dynamics of the input signal and so, the fitness function had to capture the discrepancy between obtained and desired response. The most straightforward approach (used, e.g., by Knabe et al., 2006) would be to minimize the total error:

$$f_{err} = \sum_{t=0}^{L-1} |o_t - d_t| \tag{4.1}$$

where $o_t$ and $d_t$ are the desired and obtained expression levels at the time $t$ (alternatively, a squared error can be used) and $L$ is the number of steps the network is simulated. This approach was found to work during initial experiments. However, sometimes it generated solutions that would have relatively low error value (i.e., high fitness) despite being far from what the designer of the target pattern had in mind. Discrepancy between an objective fitness measure and perceived usefulness of an obtained solution is a problem shared among all fitness driven optimization methods and reflects the fact that a fitness function is often only a crude approximation of the actual concept of fitness that an engineer has in mind. To give an example, if the desired behaviour of the network would be to generate two consecutive spikes in concentration of an artificial protein, the actual timing of the onset of their occurrence may not be important for the designer, whereas the above fitness function rewards only precise timing. In such a case, evolvability could most likely be improved by embedding additional information into the fitness measure based on the knowledge about relevant features of the desired pattern. In the example discussed, a modified fitness function that selects minimal error value among possible delays would better reflect the intention of the designer. It is a common property of optimization methods that as long as the search space is not random, the search

algorithm can always be improved by incorporating additional knowledge about the problem into the search process.

Two such modifications were introduced to the above fitness function (Eq. 4.1), in a form of additional terms, to better suit the nature of the problems presented in this chapter. The first one was based on the observation that the proper response of a network would frequently involve responding with a desired number of oscillations, i.e., the important information was contained in the number of concentration pulses, not their exact shapes. For this reason, an additional term was added that explicitly rewards the correct number of oscillations in the response.

The second modification stems from the fact that for the problems presented in this thesis, the relevant part of the response is the part where non zero concentration of the cellular effector interpreted as the output of the system is expected. Since such a response lasts only for a short time compared with the whole length of the response window, a higher weight was assigned to the "active" part of the response (the part with expected expression above 0). The justification for adding such term comes from the fact that, by the nature of the system, producing some non zero concentration of TF requires more genetic components than keeping it at zero. This creates a bias towards the simplest solution of producing no response at all. Such a simple solution manifested itself in experiments where the final score is a trade-off between error values obtained from many test cases. Thus, a term was added to the fitness function that assigns higher weight to those time steps in the windows of desired response that require concentration greater than zero.

The adjusted fitness function used in the experiments described in this chapter was:

$$f = \mu \sum_{t=p}^{L-1} |o_t - d_t|(1 + kd_t)\frac{1}{2 - S} \quad \in [0, 1] \tag{4.2}$$

where $L$ is the lifetime of the GRN (between 600 and 1500 time steps, depending on the experiment) and the term $(1 + kd_t)$ provides higher weight to the time steps where non zero concentration is expected ($k = 1$ was used in experiments discussed). $p$ is the propagation time after which the activity of the output is evaluated (set to 50 time steps). This accounts for the fact that there will always be some latency in the network's response. $\mu$ is a normalizing term:

$$\mu = \frac{1}{\sum_{t=p}^{L-1} \max(1 - d_t, d_t)(1 + kd_t)}$$

The final term $(\frac{1}{2-S})$ in Eq. 4.2 promotes the correct number of oscillations. $S$ becomes equal to 0 when there is a match in the number of oscillations. It becomes 1 when no oscillations are found in the target pattern or there are more than two times then desired. For intermediate cases, $S$ was equal to $\frac{|N_{obtained} - N_{desired}|}{N_{desired}}$. The number of oscillations was defined as the number of events when the concentration crossed the level of 0.5 (i.e., $d_{t-1} < 0.5$ and $d_t \geq 0.5$ or $d_{t-1} \geq 0.5$ and $d_t < 0.5$), with

**Table 4.1:** Types of products and promoters enabled in the experiments on evolving GRNs for signal processing and the interpretations of subsequent input and output elements.

| Promoter types | Product types | External factors | Effectors |
|---|---|---|---|
| additive<br>multiplicative | transcription factor | "1" (fixed high concentration)<br>input 1 (depends on the experiment)<br>input 2 (depends on the experiment) | signal output |

**Table 4.2:** Essential GA parameters used in the experiments on evolving GRNs for signal processing. Additional parameters are provided in the Appendix (Table C.1, p. 184).

| Parameter | Value |
|---|---|
| Population size | 300 |
| Elite individuals | 5 |
| Asexually created individuals | 195 |
| Individuals through crossover | 100 |
| Initial population | randomized genomes, 5 regulatory units each |
| Termination condition | no improvement for 500 generations |
| Selection | tournament, $k = 10, p = 0.3$ |

a minimum distance between such events set to 10 time steps, to prevent counting trivial fluctuation around the level of 0.5.

For experiments where multiple training input-output pairs were used, the final fitness was an average of error for every tested pair of stimulus and response.

Section 4.4.1 (p. 90) investigates whether additional terms added to the fitness function (Eq. 4.2) indeed provide a benefit over the simpler version (Eq. 4.1).

### 4.1.2 Genetic algorithm and model settings

The overview of the GA employed in the simulations described in this thesis was introduced in section 3.3 (p. 69). In the experiments discussed in this chapter, a subset of available types of genetic elements was used (Table 4.1). An external signal was provided through an externally driven concentration of the external factor, called *Input 1*, with additional *Input 2* used in some experiments. The concentration of cellular effector *Signal output* was interpreted as the output of the network.

Genetic algorithm (Table 4.2) was configured to stop when no improvement in fitness of the best individual was observed for 500 generations. This typically resulted in runs lasting for a few thousands generations.

## 4.2 Internally induced oscillations

Genetic oscillators are known to perform vital functions (organizing, e.g., the cell cycle and serving as various biological clocks). The concept of oscillations in simulated GRNs was central already to their very first RBN-based models (see section 2.1, p. 52). However, so far a limited number of works focused on evolving oscillating regulatory networks with continuous product concentrations (see, e.g., Knabe et al., 2006; Kuo et al., 2004, for some recent examples). Such genetic oscillators can be

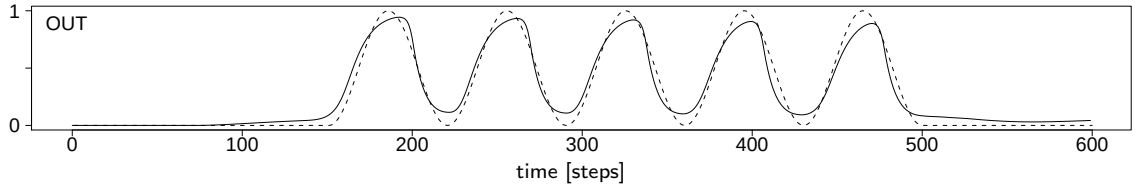



**Figure 4.1:** Behaviour (concentration of the "signal output" effector over time) of the network that generates a sine wave expression pattern lasting for five periods (the best network in 10 independent runs); dashed line: the desired response.

driven by external periodic stimuli and can also be sustained internally when such stimuli are removed (although usually with the accuracy degrading over time).

In the example discussed in this section, the external periodic input was not enabled to test if the desired dynamics can be initialized and sustained entirely by the gene regulatory network itself. The only input to the system was the constant high concentration of external factor "1", necessary to bootstrap any subsequent activity (gene expression) in the network. It was found that networks generating sine waves of various frequencies are very easy to obtain and correct frequency and phase was observed in all 10 independent runs. Although the worst runs resulted in a reduced amplitude of oscillations, the phase and frequency were always correct. Furthermore, the oscillations were observed to be stable and persisted even beyond network's evaluation time.

In an attempt to create a more challenging task, the target expression pattern was set to a sine wave that was expected to start at $t = 150$ and to end after 5 periods. A synthetic version of such a network could, e.g., work as an element of an intelligent drug delivery system, releasing a certain dose over 5 consequent days and then disabling itself. Figure 4.1 demonstrates the best individual obtained in 10 runs (see Fig. 4.16, p. 90 for a summary of fitness function values obtained for all problems). This problem turned out to be more challenging for the GA and only one in ten runs resulted in the desired behaviour. All other runs ended with individuals in which oscillations would start at the desired time, but would never terminate. However, since the part of the lifetime of the individual when oscillations are switched off was only a small component of the calculated error, it is reasonable to expect higher yield of desired solutions if the fitness function was modified to put higher emphasis on the final period of inactivity.

Another conceptually simple pattern tested, an oscillation with frequency decreasing over time (Fig. 4.2), did not result in a valid solution in any of the 10 repeated evolutionary runs. Despite further attempt to repeat the experiment with lifetime of an individual extended to cover more oscillations, networks would just remain locked to some single average frequency. Thus, in this case, the apparent ease with which GRNs generate oscillating patterns leads to a difficulty in obtaining a pattern where frequency has to change over time.

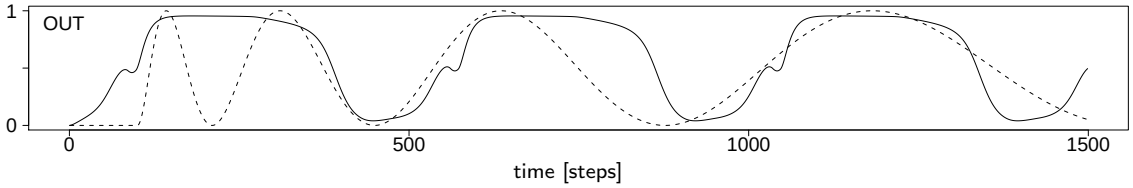



**Figure 4.2:** A pattern of concentration for which no valid solution was found. Behaviour of the best obtained network is shown; dashed line: the desired response.

## 4.3 Responding to external signals

In the following experiments, networks were evolved to generate certain expression patterns in response to continuous signals from the environment. Signal was provided as a concentration of an external factor (or two in some experiments) that was externally driven. Such an external factor can bind to promoters of genes exactly in the same manner as any other TFs produced by the cells (section 3.2), with the exception that it could not directly control the concentration of the output protein (see the model description, p. 67). This was a design decision enforcing some minimal complexity of the signal processing: any information that reaches the output of the system has to pass through at least a single internal node of the regulatory graph.

To facilitate evolution of general solutions, each genome was evaluated on a set of pairs of input and desired output (a training set). The GA was set to minimize averaged fitness function value (Eq. 4.2) over all test cases. To evaluate generality of the obtained solutions and test for overfitting, evolved networks were also tested against stimuli not present in the training set.

### 4.3.1 Doubling the oscillation frequency

In this problem, networks were expected to double the frequency of input oscillations (sine wave) and do so regardless of the frequency. The training set consisted of three training examples: 2 different frequencies of sine-like curve and an additional example with no input signal, requiring a silent response (Fig. 4.3). The silent input was introduced to facilitate evolution of networks that activated only when some external signal was provided.

Valid solutions (understood as displaying the desired behaviour) appeared already in the first few hundreds generations and continued to be refined over the rest of their much longer evolutionary history (Fig. 4.4). The best individuals obtained in all of the 10 independent evolutionary runs display desired behaviour on the training set (Fig. 4.3).

The best networks were found to generalize the problem, i.e., they would double any intermediate frequencies as well as those lower or higher than found in the training set (Fig. 4.5). Obtaining the proper behaviour for frequencies much lower than those present in the training set posed no problem (and for the best individuals, there seemed to be no minimum frequency, Fig. 4.5a). Generalization for frequencies above those present in the training set turned out to be more challenging, with

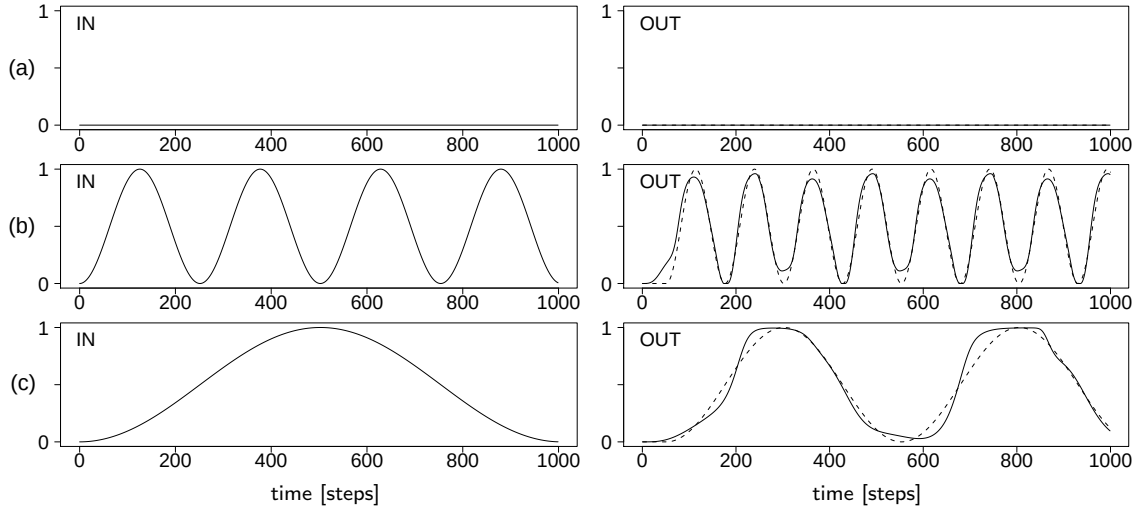



**Figure 4.3:** Training set and the behaviour of the best network evolved to double the frequency of the input oscillations. The best solution in 10 evolutionary runs, obtained after 6191 generations, $f = 0.01783$ is shown. Dashed line: the desired response.

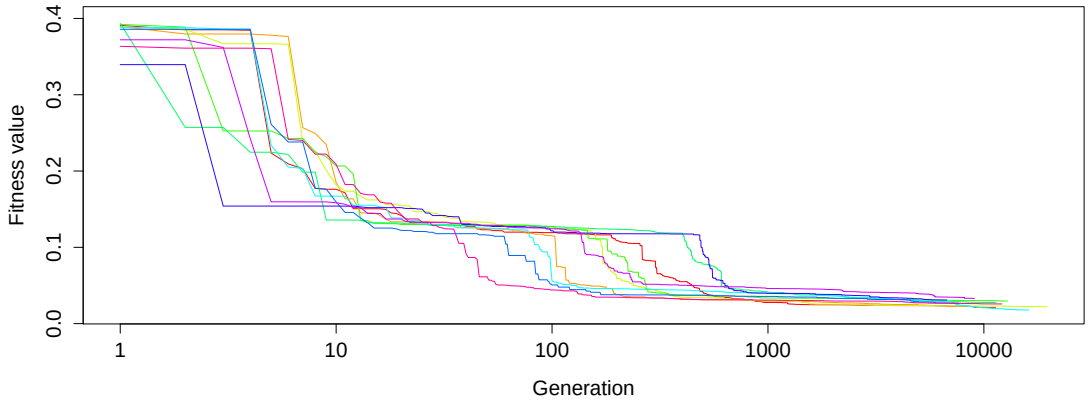



**Figure 4.4:** Fitness improvement over generations during evolution of networks doubling the frequency of the input oscillations (10 independent repetitions). In the experiments described in this chapter, lower values of the fitness function correspond to higher fitness. Note the logarithmic X axis.

networks tolerating usually no more than around 20-40% higher frequency, gradually making the spikes less pronounced as the frequency increases (Fig. 4.5b).

Poorer generalization for higher frequency signals is to be expected if one considers that maximum oscillation frequency of evolved networks is ultimately limited by how quickly concentration of a single product can build up and be degraded in the system (Eq. 3.2, p. 68). In practice, the limit will be even lower because of the latencies introduced by multiple genes involved in generating the response. To find out if the discussed experiment was indeed close to such limit, the training set was modified to employ two times lower frequencies of both the training stimuli and of the matching response. Interestingly, the networks would properly generalize very low frequencies, but again fail to respond to frequencies higher than the highest frequency in their training set. This was despite networks obtained in the previous experiment being able to generate frequencies this high. This means that, in this case, the frequency limit does not stem from limits on the speed of product accu-

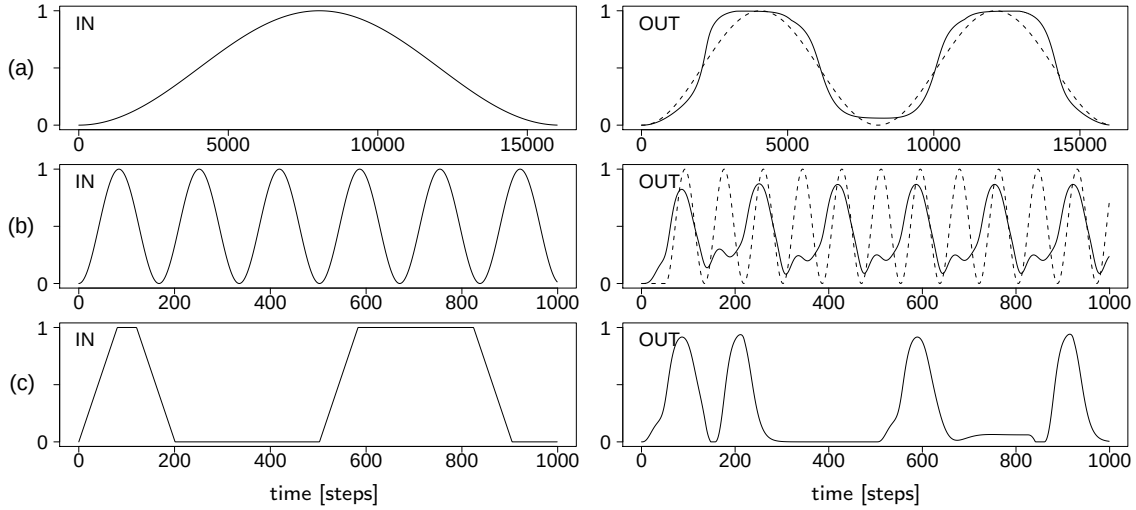



**Figure 4.5:** Problem generalization by the network evolved to oscillate at double of the input frequency (same individual as in Fig. 4.3): (a) the network behaves correctly for an input with 16x times lower frequency than in the training set, but fails to generalize for inputs with higher frequency (b). The response for the input signal in panel (c) hints on the way in which the output is calculated. Dashed lines in (a-b): the desired response.

.

mulation and degradation present in the system but rather results from networks adjusting their dynamics to the range of frequencies found in the training set. This is the case for the high frequency in particular, but the effect of this adjustment can also be observed for lower frequencies. When the input frequency is lowered, the output product starts to accumulate and degrade noticeably too fast (compare the difference in steepness of the slopes in Fig. 4.3c and Fig. 4.5a).

Evolved networks were further tested for their capability to adjust to changing input frequency "on the fly" rather than to maintain the frequency induced at the beginning. Indeed, the best networks would continuously match the output frequency to double that of the input. Given that for the training set the input frequency was constant over time, the ability to adjust dynamically demonstrates good generalization properties. However, this was not the case for some of the less fit individuals which would lock their output frequency to the double of the frequency observed initially on their input and would ignore any further change in the input frequency.

The analysis of the topologies of the evolved networks revealed that they have a very high density of connections. This makes inferring how they process information difficult (see example network on Fig. 4.6). However, a hint on the inner mechanics can be obtained by testing network behaviour for different stimuli. Figure 4.5c presents the response of this network when stimulated with trapezoid waveform of variable duty cycle. This shows that this particular network evolved to solve the problem by reacting to the raising and falling slopes of the signal, producing a spike in TF concentration for each.

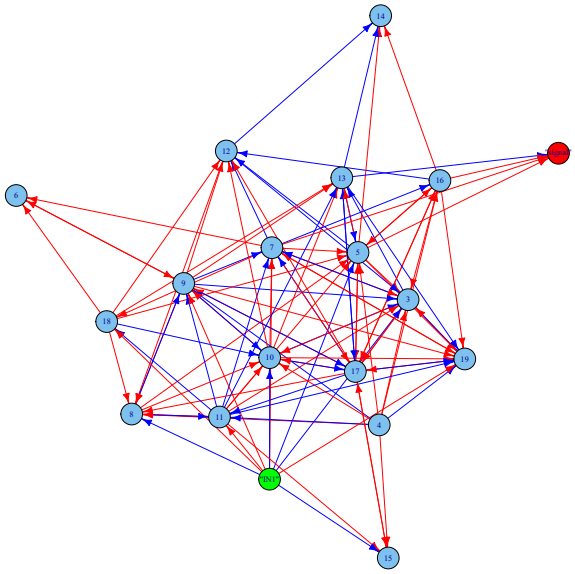


**Figure 4.6:** Regulatory graph of the best obtained individual evolved to oscillate at double of the input frequency (network of an individual seen in Fig. 4.3 and 4.5). Obtained networks were found to be very dense, only 20% of the strongest connections in this networks is shown (disconnected nodes are not drawn). Green node: signal input, red: output. Connections with positive weights are red, with negative are blue.

.

### 4.3.2 Low pass filter

Filtering high input frequency can be expected to be a problem well suited for regulatory networks, as the limited speed of accumulation and degradation of TF should tend to smooth out any high frequency changes just as a capacitance in an RC filter does. In this task, networks were selected for their ability to regenerate sinusoidal input oscillation, but only if its frequency was below certain threshold, thus implementing a form of a low pass filter.

For an initial experiment, 5 training input/output pairs were used (Fig. 4.7), two with frequencies below a threshold, two with above and one with no input signal (and requiring no response). GA was successful in finding individuals that would properly react to training pairs and also generalize for frequencies higher and lower than those in the training set. However, providing the evolved network with a sum of two sinusoids (one having a frequency above the filtering threshold) would result in blocking the output completely (not shown). This suggests that such networks filter the signal simply by detecting a quickly rising slope of the input and blocking the output if the slope raises too quickly.

In an attempt to evolve a more general solution, the training set was extended by 4 additional training examples (Fig. 4.8). They consist of inputs with two sine waves combined and require the network to filter out just the higher frequency component. Thus, ultimately, the networks were evaluated on 9 I/O pairs, shown in figures 4.7 and 4.8, and the behaviour of the best individual obtained in 10 evolutionary runs is overlaid on the both figures. When this network was tested on input patterns not present in the training set, although imperfect (Fig. 4.9a), some level of generaliz-

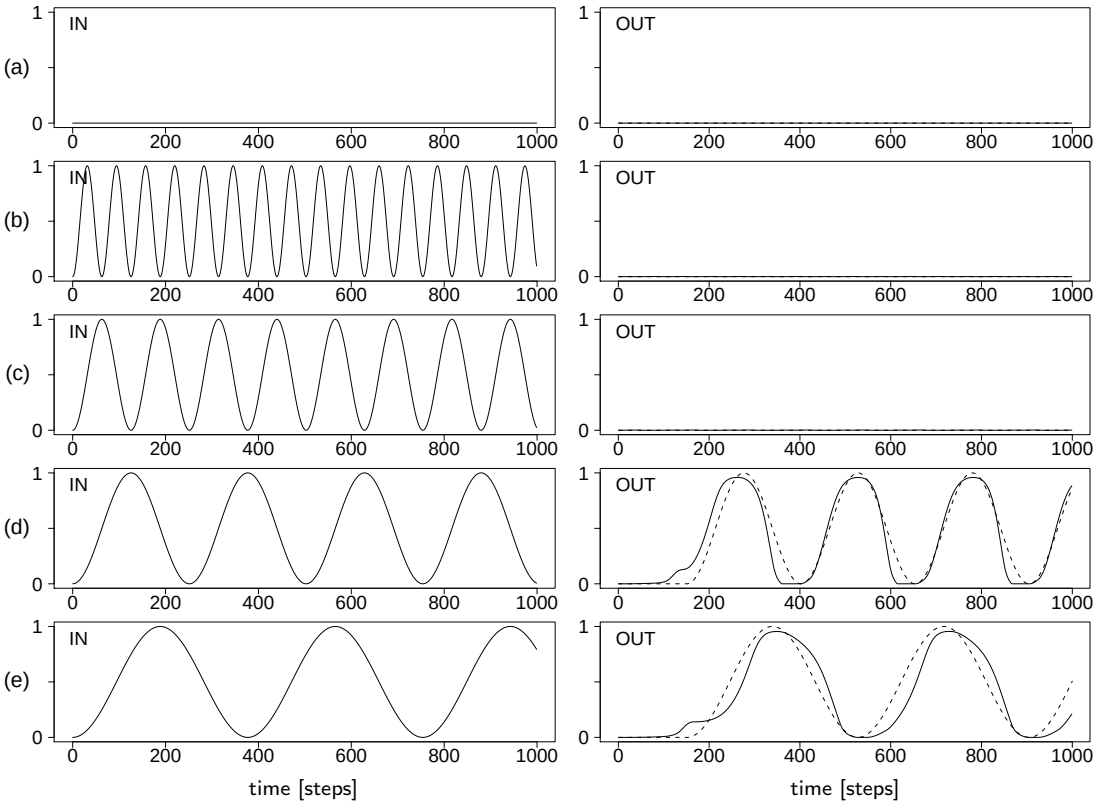



**Figure 4.7:** Part I of the training set used to evolve a low pass filter, with the behaviour of the best network from 10 independent runs overlaid (obtained in generation 8839,$f = 0.02659$). Dashed line: the desired response.

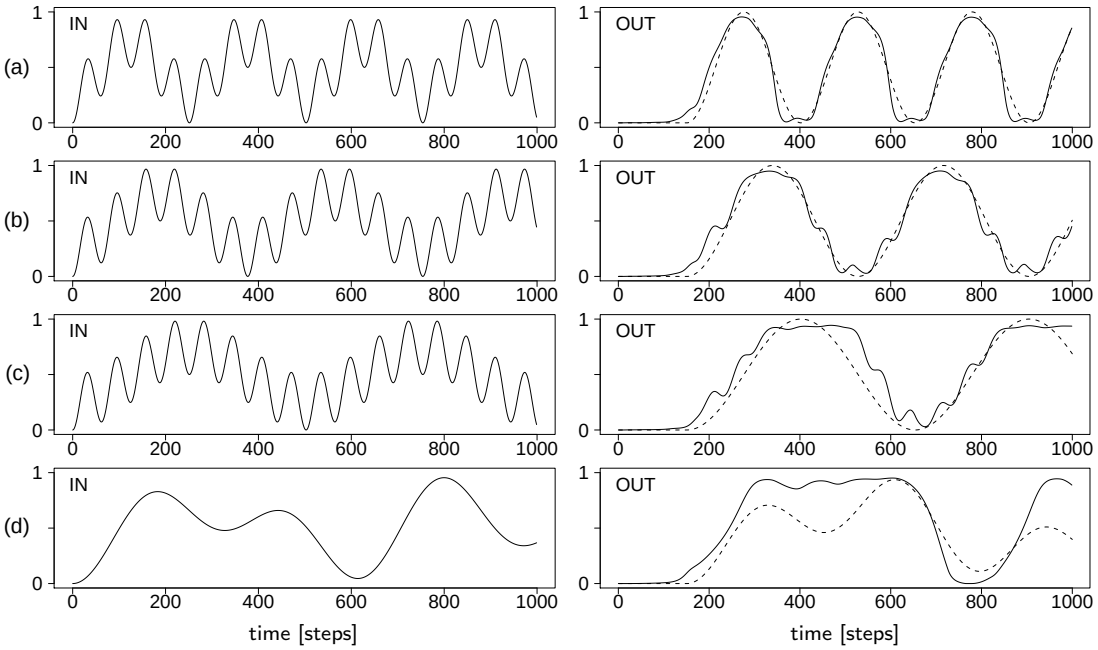



**Figure 4.8:** Part II of the training set used to evolve a low pass filter, with the behaviour of the network seen in Fig. 4.7 overlaid. Dashed line: the desired response.

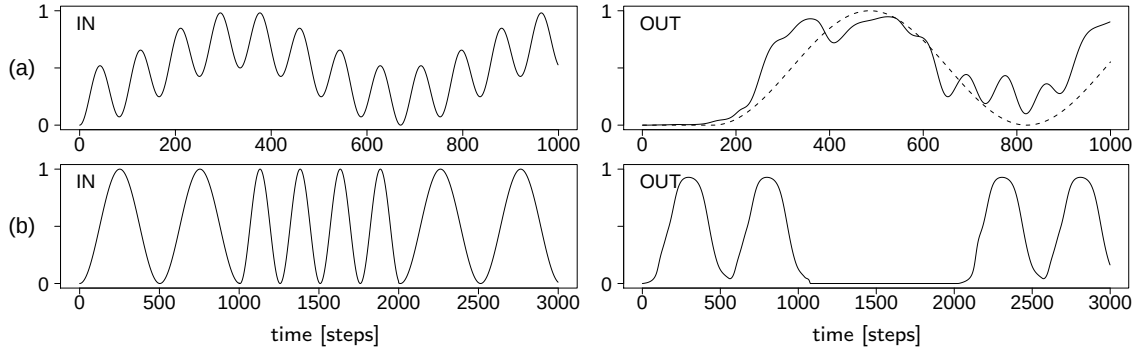



**Figure 4.9:** Problem generalization by the network evolved to act as a low pass filter (same individual as seen in Fig. 4.7, 4.8): (a) response to a combination of two sine stimuli, one above filtering threshold (dashed line in shows the ideal response), (b) the network reacts to the change in input frequency "on the fly".

ation could be seen as the high frequency component is largely filtered (and, given limited time for a response, perfect filtering would not be possible). Furthermore, the network is capable of adjusting its output to the changing frequency of the input signal "on the fly" (Fig. 4.9b), thus generalizing the problem in a desired manner, even though such scenario was not present in the training set.

### 4.3.3 Networks with signal memory: doubling the input pulse length

The problems described in previous sections (i.e., 4.2, 4.3.1 and 4.3.2) did not require explicit memory of the input signal. In this task, networks were evolved to respond with a square pulse twice the length of the square pulse on the input after 50 simulation time steps, hence requiring them to sustain some memory of its length. Four training pairs were used (Fig. 4.10). Desired behaviour was observed in all of the best networks from 10 evolutionary runs, suggesting that this is not a difficult problem. The best individual shown in Fig. 4.10 was also found to generalize the problem by properly responding to the pulses occurring on its input at any time and respond accurately to signals of intermediate lengths, as well as 50% shorter (Fig. 4.11). Pulses longer than present in the training set would lead to responses shorter than desired (Fig. 4.11d), exposing the leaky nature of the GRN-based memory.

When the network was stimulated with a pulse that had just half the height of those in the original training set (Fig. 4.11e), the length of the output pulse would be close to that of the input stimuli. This suggests that the network evolved to rely on some form of a simple integrator (e.g., by slowly building up concentrations) rather than, e.g., rely on detection of the raising and falling edge of the input signal.

To evaluate sustainability of this type of memory, the evolution was simulated with a modified training set in which the desired output, instead of being expected to start appearing 50 time steps after the input, was expected to appear 350 time steps later (Fig. 4.12). This required storing the memory of the length of the input pulse for a longer time. Although the best individual would still demonstrate the desired behaviour, obtained networks were less accurate and on average evolvability was worse, clearly demonstrating the increasing difficulty of storing information

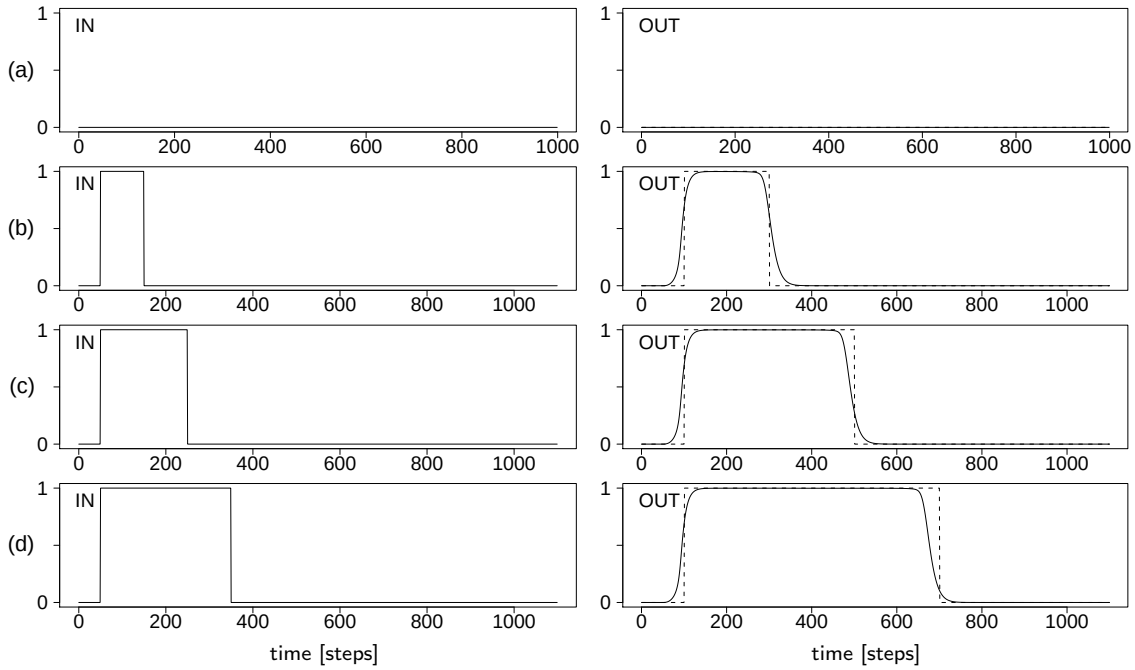



**Figure 4.10:** Training set and the behaviour of the best individual in 10 evolutionary runs evolved to double the input pulse length (obtained in generation 7195, $f = 0.01465$): dashed line shows the desired ideal response (note that a perfectly square response would be impossible to obtain).

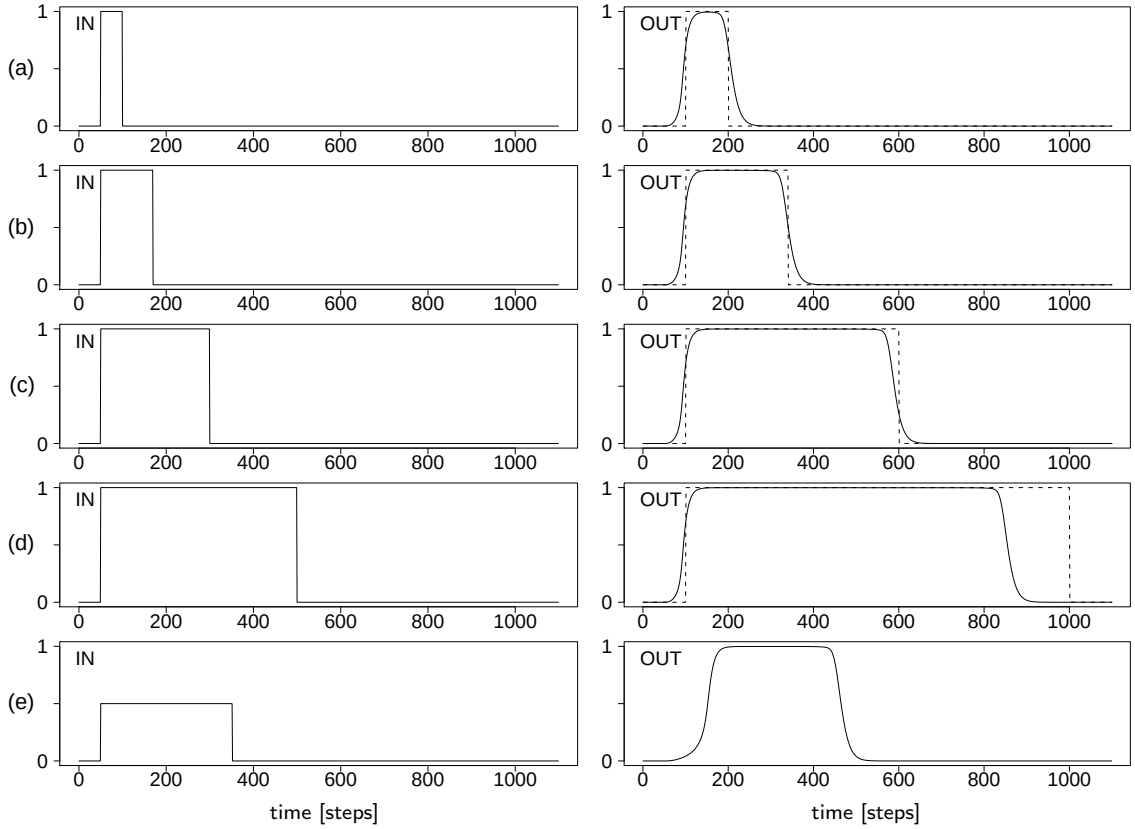



**Figure 4.11:** Problem generalization by the network evolved to double the input pulse length (same individual as in Fig. 4.10). Dashed line (a-d) shows the desired, ideal response.

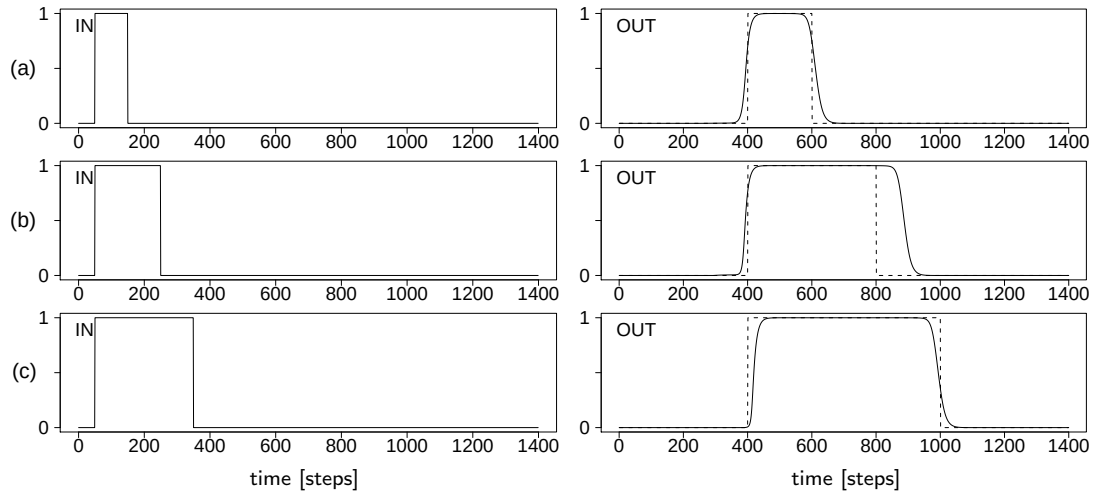



**Figure 4.12:** Behaviour of the best individual from 10 independent runs evolved to double the input pulse length, but with an increased delay of the response.

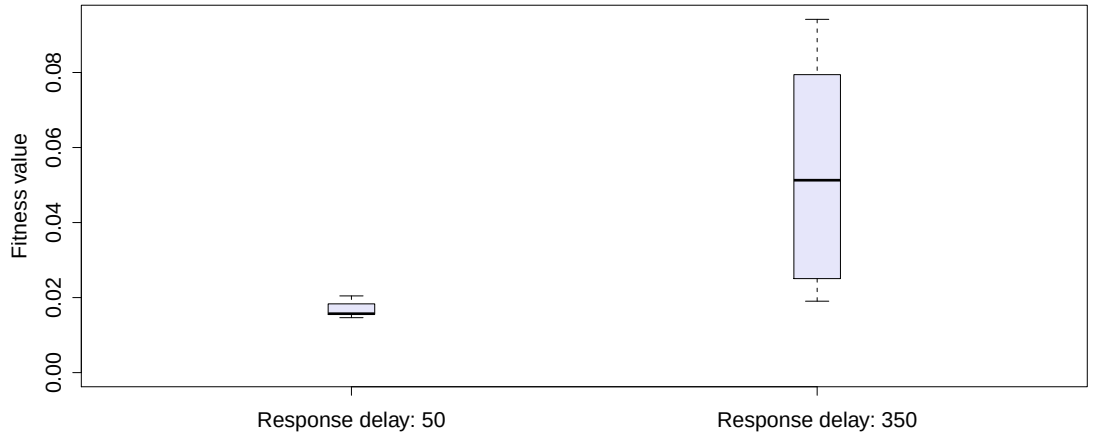



**Figure 4.13:** Comparison of fitness function values (lower is better) of the best networks evolved to double the input pulse length for a response delay of 50 and 350 time steps. The box plots show the median and quartiles for 10 best networks obtained in 10 independent evolutionary runs for each delay. Whiskers extend to the most extreme data point which is no more than 1.5IQR from the box (here, a maximum and a minimum).

in concentrations for longer times. Figure 4.13 compares fitnesses obtained in 10 repetitions of each type of experiment.

### 4.3.4 Doubling the number of the input spikes

The problems described in this and the following section (4.3.5) were conceived to test if it is possible to evolve GRNs that would be able to react to subsequent spikes in concentration (a sharp increase followed by a sharp decrease) and also present response as a consecutive spikes. This type of desired response is not well suited for GRNs since a single spike, despite carrying a single bit of information (i.e., spike or no spike), involves two processes: building up of the concentration of a product and then degrading it. Thus, the main purpose of defining the problems in this manner was to present the GA with more challenging tasks. Encoding information in pulses may also superficially resemble the type of processing performed by spiking neural

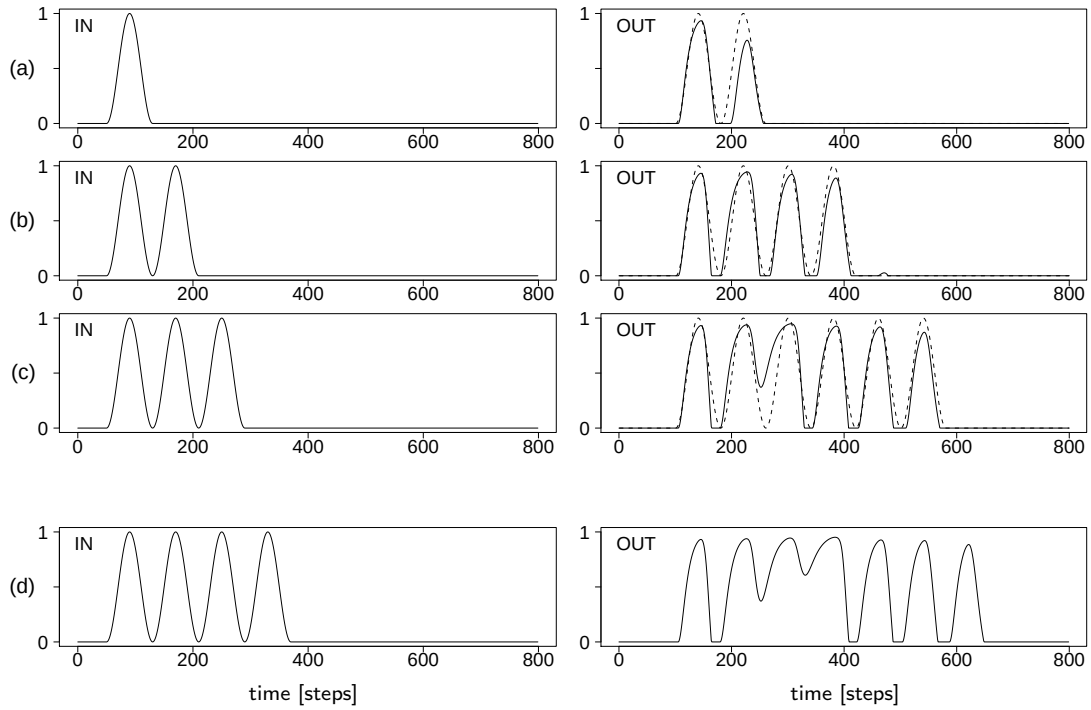



**Figure 4.14:** Behaviour of the best individual from 10 evolutionary runs evolved to double the count of concentration spikes on its input (obtained in generation 2794, $f = 0.02583$): (abc) training set with output obtained and desired, dashed line shows the desired response (d) testing for generality: the network responds with less spikes than desired when a higher number of spikes is presented on the input.

networks. However, in the case of spiking neural networks each spike represents changes of voltage across the membrane (which is caused by changes of ion concentration in the cell relative to the outside), whereas for regulatory network, each spike corresponds to a change of concentration of a protein resulting from a positive regulation followed by active degradation.

The first of the discussed problems required networks to respond to a series of pulses with the double of their number. The best solution obtained in 10 independent runs doubles the number of pulses for the training set as required (Fig. 4.14a-c). However, the response for the three pulses is imperfect (Fig. 4.14c). Note that, although the TF is not degraded completely, the concentration falls below 0.5, most likely an effect of a strong pressure to gain a reward provided by the additional term in the fitness function (section 4.1.1, p. 76). The best networks in most of the other runs would perform poorer, suggesting overall difficulty of this task.

Finally, the level of generalization of the problem was evaluated by stimulating the network with 4 consecutive pulses (Fig. 4.14d). The problem was not correctly generalized. The network would respond with only 7 pulses instead of the expected 8.

### 4.3.5 Integrating information from two separate signals: serializing pulses

The following problem was designed to test the capability of GRN to integrate information from two inputs rather than a single one. For this purpose, a second

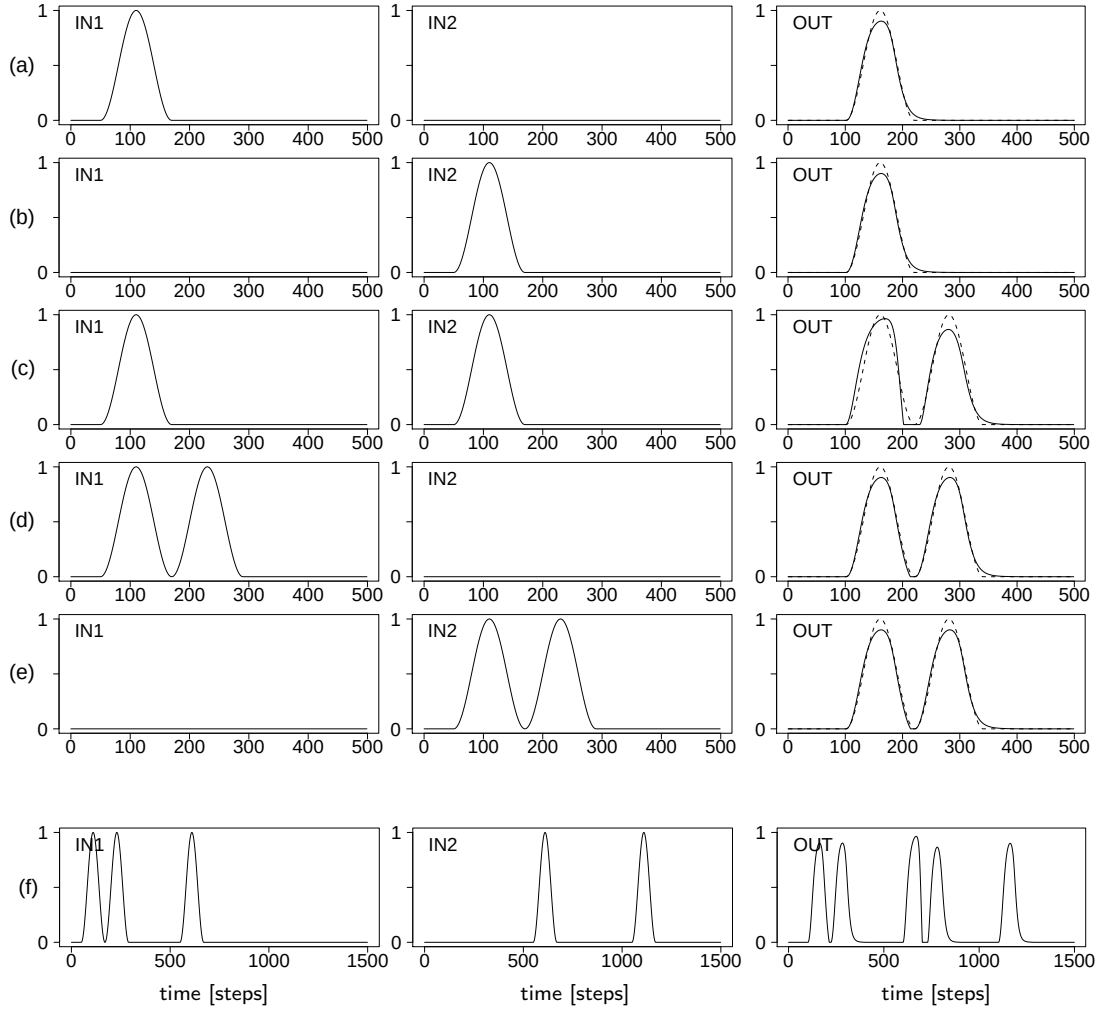



**Figure 4.15:** Behaviour of the best individual in 10 evolutionary runs evolved to count subsequent or simultaneous spikes on its input (obtained in generation 2068, $f = 0.01394$): (a-e) the training set with the output obtained and desired, (f) testing for generality.

external factor was enabled in the system, responsible for the second external input. The task was to respond with the number of output pulses equal to the number of pulses on both inputs observed within a certain time window. This can also be considered a form of simple GRN-based adder that presents response as subsequent spikes in concentration. Again, this is not the type of information encoding which is best suited for GRN-based processing: an adder operating on the levels of concentrations, rather than spikes, would be much easier to construct or evolve.

Nonetheless, a relatively high match to the desired response was obtained on the training set (Fig. 4.15a-e). The best individual not only properly serializes the pulses but also is general and works in a continuous manner: given sufficient time separation between the pulses, it can summarize correctly (Fig. 4.15f).

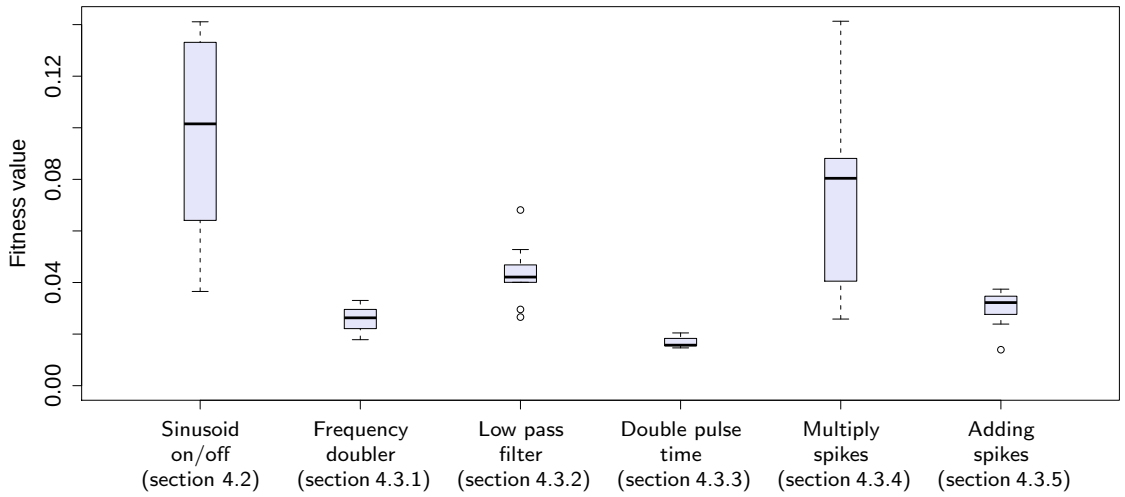



**Figure 4.16:** Fitness values of individuals obtained for each task. The box plots show the median and quartiles for the best 10 networks each obtained in an independent evolutionary run for each type of problem. Whiskers extend to the most extreme data point which is no more than 1.5IQR from the box. Circles indicate outliers.

## 4.4 Evolvability

The fitness values of the best individuals obtained in 10 independent repetitions of each discussed problem allow to partially infer the difficulty of each task (Fig. 4.16). However, the measure is a very imperfect one. The quality of solutions obtained for a given problem highly depends on the choice of the training set. Furthermore, because of the complexities of the fitness function discussed in section 4.1.1 (p. 76), the very formulation of the function influences the problem difficulty. It is perhaps the variance that represents the difficulty better than actual fitness value, as its low value suggests that the solution was easily discoverable by GA (as long as each run indeed did result in a desired solution). For example, both low fitness function values (i.e., the high quality solutions) and very low variance obtained for networks doubling the input frequency or doubling time of the pulses indicate the relative simplicity of these tasks.

### 4.4.1 Alternative fitness functions

The impact of the additional terms of the fitness function discussed in section 4.1.1 was investigated to find out whether such more complex function does indeed improve evolvability. An experimental setup used to evolve networks doubling the input signal frequency (section 4.3.1, p. 80) was selected for its simplicity and relatively low standard deviation of the end results of experiments (Fig. 4.16). To improve control over experimental conditions, genetic algorithm was set to run only for 2000 generations, instead of running as long as improvements were observed. Because of variation in fitnesses of obtained individuals in each independent run, each experiment was repeated 40 times.

Four experiments were performed, each using a different version of the fitness function listed below (refer to Eq. 4.2, p. 77 for the explanation of each term):

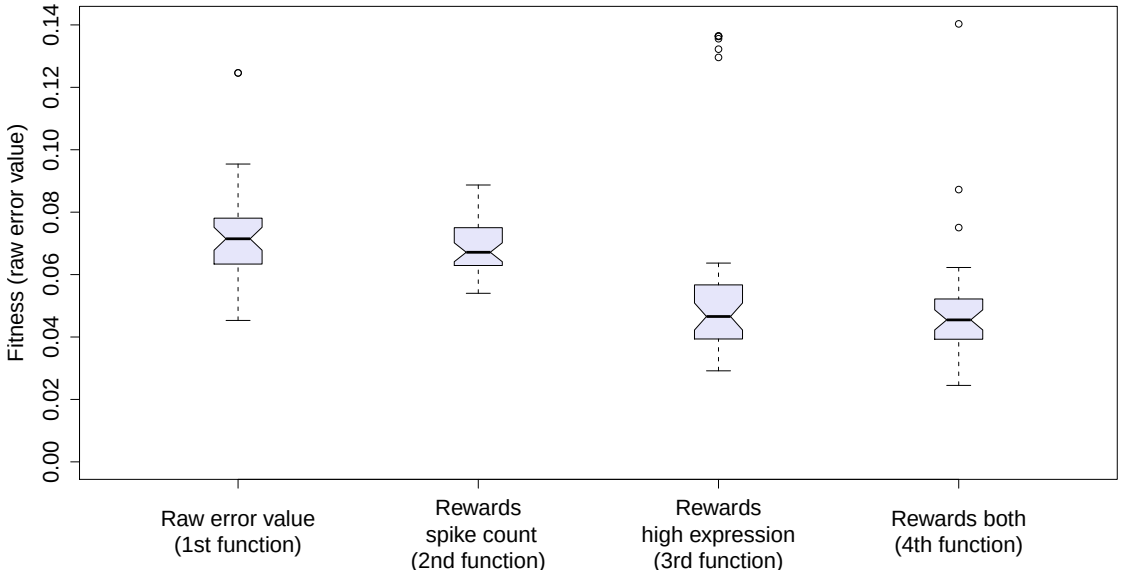



**Figure 4.17:** Comparison of evolvability with different versions of fitness function in signal processing experiments. The box plots show the median and quartiles for the unadjusted fitness value (1st function) of 40 best networks obtained in 40 independent evolutionary runs using each type of fitness function. Whiskers extend to the most extreme data point which is no more than 1.5IQR from the box. Circles indicate outliers. The notch marks 95% confidence interval for the median.

1. simple unadjusted error function: $\mu_1 \sum_{t=p}^{L-1} |o_t - d_t|$
2. error function that assigns higher weight to time points with a higher desired concentration level ($k = 1$): $\mu_2 \sum_{t=p}^{L-1} |o_t - d_t|(1 + kd_t)$
3. error function that rewards correct number of oscillations: $\mu_1 \sum_{t=p}^{L-1} |o_t - d_t| \frac{1}{2-S}$
4. error function that rewards for both of the above: $\mu_2 \sum_{t=p}^{L-1} |o_t - d_t|(1 + kd_t) \frac{1}{2-S}$ (the default one used for the experiments discussed in this chapter, Eq. 4.2)

where $\mu_1$, $\mu_2$ are normalizing terms:

$$\mu_1 = \frac{1}{\sum_{t=p}^{L-1} \max(1 - d_t, d_t)}, \qquad \mu_2 = \frac{1}{\sum_{t=p}^{L-1} \max(1 - d_t, d_t)(1 + kd_t)}$$

Although the GA in each of the four experiments would rely on a different definition of fitness, to allow for comparison between experiments, the fitnesses of the final individuals obtained in each of the repeated experiments were recalculated using the unadjusted error function (the first equation).

The comparison (Fig. 4.17) suggests that a more complex fitness function improves evolvability. To see if the average errors obtained with the adjusted function are indeed significantly different from those obtained with the unadjusted function, a Wilcoxon rank-sum test was performed on the results of 1st (unadjusted) and 4th experiment (adjusted). The test provides strong evidence that the adjusted function

(4th setting) improves evolvability in the presented problem ($p < 10^{-10}$, Wilcoxon rank-sum test, two tailed). Despite small apparent (Fig. 4.17) improvement in average value between 2nd and 1st setting, as well as between 4th and 3rd, the test did not provide evidence for the relevance of spike counting term for this task. However, this term would more likely be helpful in problems discussed in sections 4.3.4 and 4.3.5. In overall, the results suggest that adjusting the fitness function by incorporating additional knowledge about the GRN did allow for a more effective search and was most likely helpful in the experiments presented in this chapter.

### 4.4.2 Parameters of the model

Every model of artificial GRN is a compromise between biological realism and computational efficiency. Many features of the model depend on intuition driven choice of selection schemes, the sizes of populations or numerous other parameters, e.g., the constant that determines how the weight of a connection in the GRN is scaled with the distance between coordinates of genetic elements. Comprehensive exploration of the parameter space would be computationally prohibitive and it would be still difficult to draw definite conclusions on superiority of a particular setting, unless evidence from a representative sample of possible problems was gathered.

This subsection presents the results of comparison of the impact on evolvability of the three features of the model: the presence of multiplicative promoters (Eq. 3.4, p. 68), the selection scheme and the constant that controls how the distance between genetic elements translates into weights (Eq. 3.1, p. 66). Four settings were compared:

1. the original configuration (as used in experiments discussed so far)
2. the configuration in which multiplicative promoters are disabled
3. the configuration in which deflection of the affinity curve is modified ($\beta = 10$, instead of $\beta = 1$ used in experiments discussed so far, see Fig. 3.3, p. 66 for visualization)
4. the configuration in which binary tournament selection ($k = 2, p = 1$) was used instead of multiple individuals based tournament ($k = 10, p = 0.3$)

An experimental scheme for comparison was identical to the one described in the previous section (however, the usual, adjusted fitness function was used). There is no visible difference between the quality of obtained solutions for the modifications of the discussed parameters (Fig. 4.18). Indeed, no statistical evidence was found to suggest that any of the above modifications results in a different average of fitnesses of the best individuals obtained in multiple runs compared to the original setup ($p > 0.05$, two-tailed Wilcoxon rank test). Although one can reasonably assume that some differences in evolvability do exist, the lack of statistical significance indicates that the measured effect must be small and that the system is not very sensitive to these parameters. It also suggests that the multiplicative promoters are

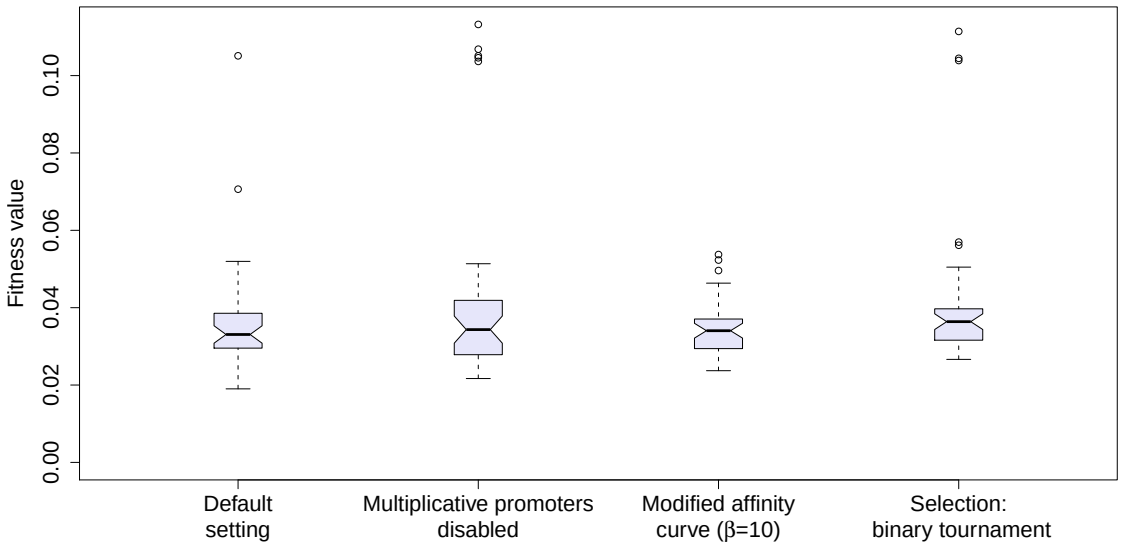



**Figure 4.18:** Comparison of evolvability for 4 modifications of selected parameters of the model. The box plots show the median and quartiles for fitness values (lower is better) of 40 best networks obtained in 40 independent evolutionary runs for each experimental setting. Whiskers extend to the most extreme data point which is no more than 1.5IQR from the box. Circles indicate outliers. The notch marks 95% confidence interval for the median.

not necessarily useful and thus, their use was limited in the later chapters for the sake of simplicity.

## 4.5 Discrete vs continuous dynamics

The overall structure of the artificial GRN represented as graphs with nodes performing computation bears strong similarity to perceptron-based neural networks with recurrent topologies, for example those evolved using the recently popular NEAT model (Stanley and Miikkulainen, 2002). However, two key differences exist between typical evolved perceptron networks and GRNs described in this work.

The main difference is that each regulatory node has a state that represents the current concentration of its associated TF and can only be changed by increasing the synthesis or degradation, whereas perceptrons are stateless (i.e., the output at the time $t + 1$ depends only on signals on its inputs at the time $t$). Such internal dynamics on one hand limits the reaction time of the network, but on the other may facilitate generation and processing of signals that always change gradually. The second difference is the indirect, genetic representation that assumes a linear and biologically inspired genome, rather than some form of a direct representation of the connectivity graph. This allows artificial evolution to shape the structure of networks in a manner more closely resembling the evolution of regulatory networks.

The no free lunch theorem (Wolpert and Macready, 1997) suggests that optimization strategy will outperform others if it is specialized for the problem to which it is applied. This suggests that one should not expect regulatory networks to outperform other biologically inspired computational models, but rather to have an advantage for a certain class of problems.

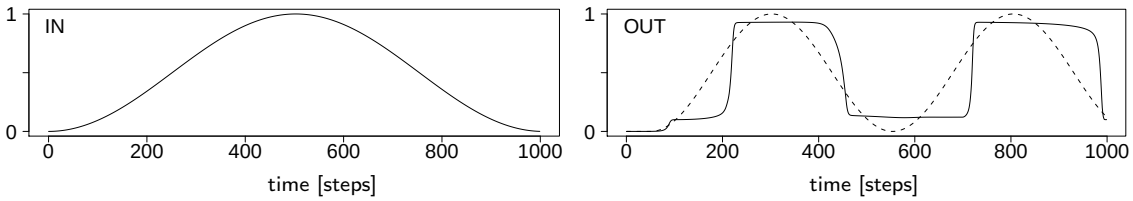



**Figure 4.19:** Behaviour of the network evolved to double the input signal frequency in which product build-up and degradation is not simulated. The best individual obtained in 10 independent runs (generation 1544, $f = 0.05091$) is shown. Dashed line: the desired response.

To test if continuous changes in product concentrations are indeed beneficial for the presented set of problems, the model was modified so that the concentration of a product was determined only by the activation of associated promoters in the previous time step. This means that the nodes in the regulatory graph became stateless and capable of changing their activation by any amount in a single time step. More precisely, the function of the activity of all promoters (Eq. 3.2, p. 68), instead of being treated as a current product synthesis level (with the range $[-1, 1]$), would be shifted right and scaled to $[0, 1]$ so that it could be treated as a new concentration level for the given time step. In such a configuration, the computation performed by the GRN becomes similar to that of recurrent networks of perceptron-like neurons, with the original method of network encoding maintained.

Networks were evolved to double the frequency of input oscillation, using exactly the same configuration of the system as in the original evolutionary run discussed in section 4.3.1 (p. 80). The best individual employing stateless nodes obtained in 10 individual runs (Fig. 4.19) does generalize the problem by correctly responding to frequencies not present in the training set, but it is measurably (Fig. 4.20) and visually worse than individuals obtained with the standard model setting in the original experiment using Eq. 3.2 (compare Fig. 4.19 with Fig. 4.3c, p. 81).

This result reinforces the initial hypothesis that inherent smoothness of regulatory nodes in the standard setting gives it an advantage in generating gradually changing outputs. One can also expect that such continuous changes in concentrations make it easier to handle noisiness of gene expression and noisiness of environmental signals, both in artificial and biological regulatory networks.

## 4.6 Robustness to noise

Noise is inherent in all biological systems. The numbers of molecules that are involved in regulation of a particular gene can often be relatively small (typically in the order of thousands, can be even as low as 10) with all reactions occurring in crowded cellular environment and subjected to thermal noise (Maheshri and O'Shea, 2007). Noise is also considered to be a significant source of phenotypic variation among cells (Blake et al., 2003; Munsky et al., 2012). Apart from intrinsic noise of cellular expression, variability of external environment further contributes to stochasticity of cell behaviour. Thus, all regulatory networks are constrained by multiple sources of randomness and evolved mechanisms that allow them to cope with it, such as,


**Figure 4.20:** Comparison of the evolvability between original model and the version in which product build-up and degradation is not simulated. The box plots show the median and quartiles for fitness values of 10 best networks obtained in 10 independent evolutionary runs for each experimental setting. Whiskers extend to the most extreme data point which is no more than 1.5IQR from the box (here, a maximum and a minimum).

for example, negative feedback loops in regulatory graphs or multi-site regulation (Bolouri, 2008). Transcriptional noise can also be used for cells' advantage and it is known to be exploited to generate diversity, e.g., among viruses (Arkin et al., 1998).

In the experiments presented so far, elements of the regulatory networks were not affected by random noise. Also, concentrations were represented as real numbers, thus even for very low concentrations, there was no negative effect of signal-to-noise ratio being reduced due to small numbers of molecules.

To simulate the effects of intrinsic gene expression noise, small random numbers were added to the calculated synthesis rates of each product. The original concentration update method (Eq. 3.2, p. 68) was modified, so that in every time step the change in expression included not only the produced and degraded product, but also a random component $R$:

$$\Delta L = (\frac{1 - e^{-A}}{1 + e^{-A}} - L)\Delta t + R \tag{4.3}$$

where R was drawn from a normal distribution.

Two types of experiments were performed. In the first one, networks were evolved without noisy synthesis rates, just as in experiments described previously. Then, the robustness of the networks to noise was analysed by re-evaluating their fitness after the noise in synthesis rates was enabled. This allowed to observe how their fitness degrades depending on the level of noise.

In the second experiment, networks were evolved already with noisy synthesis rates ($sd(R) = 0.01$) and tested for performance as well as robustness to noise higher than the one used during evolution.

The setup for the experiments was identical to that used for experiments comparing different versions of the fitness function (section 4.4.1), i.e., the problem of doubling input signal frequency was chosen and experiments were repeated 40 times. Then, the best networks obtained in 40 runs were re-evaluated with either noise disabled or random noise added to synthesis rates with $sd(R) = 0.01$ and

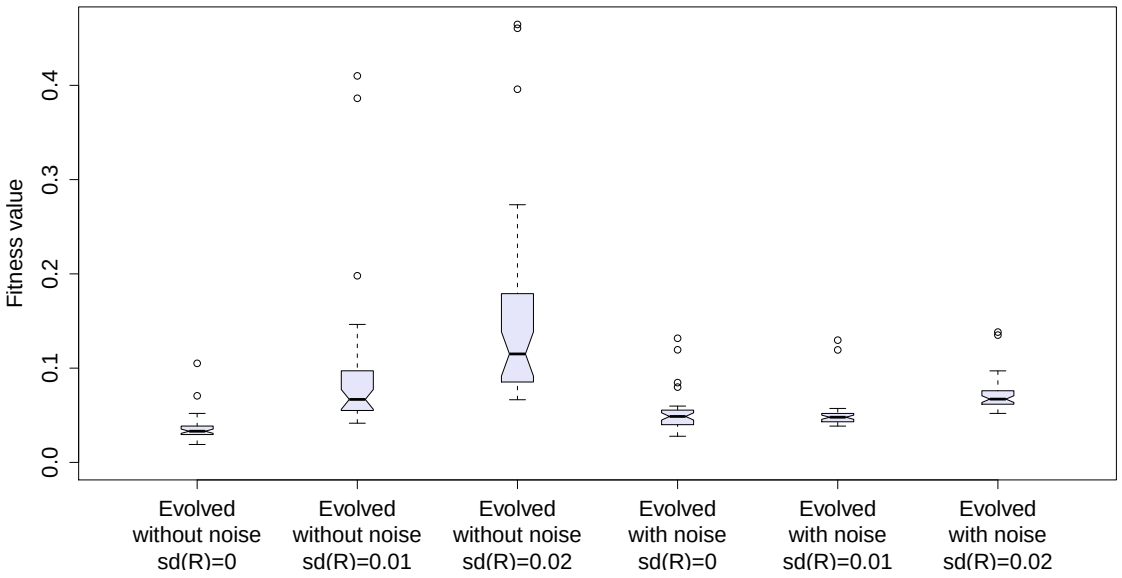



**Figure 4.21:** Comparison of robustness to noise of networks evolved without and with ($sd(R) = 0.01$) noise in gene expression. The box plots show the median and quartiles for fitness values of 40 best networks obtained in 40 independent evolutionary runs for each experimental setting. Whiskers extend to the most extreme data point which is no more than 1.5IQR from the box. Circles indicate outliers. The notch marks 95% confidence interval for the median.

$sd(R) = 0.02$. Because of the stochasticity in fitness evaluation, each network was evaluated 10 times and the average was used as a measure of its fitness value.

As soon as the noise in synthesis rates was enabled for the first experiment (evolution without noise), the error between the desired and generated output increased, even more so for higher levels of noise (Fig. 4.21). For the networks that evolved already with noisy elements, the error values are slightly higher than those of the best networks obtained in a noiseless setting. This was expected, since these networks would no longer be capable of producing perfectly smoothed sinusoidal output. However, even if the noise is disabled, their performance remained largely unchanged and is still slightly worse than that of noiseless networks. Nonetheless, the error is not increased much, indicating that the networks evolved to be robust to noisiness of their internal elements. Notably, under the same level of noisiness ($sd(R) = 0.01$), networks evolved with no noise considerably deteriorated in performance. The networks evolved with noise are also much more tolerant to noise amplitude that is twice higher than that used during their evolution. In other words, they were much more robust than the networks evolved in the default experimental setting.

To sum up, the noisiness of synthesis rates resulted in networks that are slightly worse in performance, but evolved mechanisms that allow them to greatly reduce the negative effects of noise. As soon, as noise was added to the system, they easily outperformed networks evolved without noise.

## 4.7 Summary

The goal of this chapter was to investigate the capability of the model introduced in Chapter 3 to evolve artificial regulatory networks that can generate desired patterns of gene expression or perform various forms of computation on continuously changing external signals, provided as externally driven concentrations of chemical substances. A number of problems was introduced, ranging from simple oscillators to a network that can perform additions. Generality of each solution was analysed and GRNs were found to easily generalize presented problems and perform computation on the information encoded as timing of events or in levels of concentration.

Various versions of the fitness function used to evolve GRNs were investigated and it was shown how the fitness function can be tuned to improve evolvability of GRNs, by incorporating intuition about the nature of the fitness landscape and potential local minima.

The impact of selected features and free parameters of the model on evolvability was investigated with no statistically significant effect found. In general, for model parameters that are selected by intuition (such as the deflection of the affinity curve), this is a desirable property, as it means the system works similarly across the range of parameters. On the other hand, for certain additional features, such as multiplicative promoters, the lack of evidence in favour of them can be used against it, as a simpler version of the model should be preferred. For this reason, the use of multiplicative regulatory elements was limited in the later chapters.

The structure of artificial gene regulatory network is in many ways similar to a recurrent neural network and can be considered to be a computational alternative to it. However, typically, neural networks employ stateless nodes (nodes whose state in the time $t + 1$ depends only on the state of its input at the time $t$ and not on the current concentration of a factor associated with this node), though continuous state recurrent neural networks have also been proposed (Beer, 1995; McClelland and Rumelhart, 1988) and would be a much closer computational equivalent of GRNs. The state of the nodes in GRNs is usually represented as a concentration which takes time to build up or reduce. Still, some of GRN models used in other works employ stateless nodes, e.g., Eggenberger Hotz (1997); Joachimczak and Wróbel (2008a). Stateless nodes are also a norm in models devised from RBNs. The results in this chapter show how the continuously variable state of each node can be advantageous in certain problem domains, such as those requiring processing continuous signals. This could mean that GRNs could find its usage, e.g., in evolutionary robotics, where their inherent smoothness of actuation and input processing would be a desirable property.

Another investigated property of GRNs was robustness to noisy TF concentrations. The networks were found to tolerate certain amount of noise which can be attributed to their high overall redundancy as well as noise being smoothed out by the integrating property of each node. It was found that if the networks were evolved already under assumption that TF concentrations are noisy, they were still able to solve presented problem, but evolved to be many times more robust to higher

noise levels. Hence, adding simulated noise to concentrations can be a simple way to evolve controllers that are robust to both imperfections of their components as well as that of input signals.

# Chapter 5

# Evolution of behaviour of GRN-controlled unicellular organisms

In this chapter, the applicability of evolving artificial gene regulatory networks to a control problem is investigated. Networks are evolved to control the behaviour of a simulated animat in an artificial environment. GRNs are expected to process sensory information (concentrations of chemicals) and respond with changing concentrations of external factors that influence animat's actuators.

Compared to the signal processing tasks discussed in Chapter 4, an artificial environment with simulated physics provides a more biologically plausible setting to investigate the evolution of GRNs. It allows for interactions between physical properties of the system and gene regulation. For example, the physical laws controlling the movement of the objects in the system are itself a form of computation and can be exploited by regulatory networks for their own computational needs. The interplay of physics and regulatory networks can be expected to generate more sophisticated behaviours for given network complexity (Eggenberger Hotz, 2003b) and is also an essential element of multicellular model introduced in Chapter 6.

## 5.1 Animat model and environment

Animats forage in an open 2D environment with simple Newtonian physics and simulated fluid viscosity in which energy sources are distributed. Each animat is a rigid circular object with a fixed diameter and a fixed mass and is equipped with two identical sensors located symmetrically on the front (Fig. 5.1). The sensors perceive concentration of a chemical signal present in the environment and the difference between left and right sensor can be used to extract information about the gradient. To allow the animat to move in the environment, two actuators $A_L$, $A_R$ are present at the sides of the animat's body. When any of them is active, a force is applied to the animat, causing it move. The force associated with each of the actuators is not

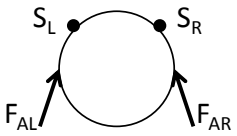



**Figure 5.1:** The model of the simulated animat. Sensors ($S_L$, $S_R$) of chemical signal (food scent) are placed on the front. Thrust generating actuators are on the back. When activated, an amount of force $F_{AL}$, $F_{AR}$ proportional to actuator activity is applied.

directed toward the centre of the animat, so the animat will turn if the activations of actuators differ. If only a single actuator is active, the animat moves in a loop. Hence, every turn an animat can take has a minimum curvature, dependent on its speed. Such actuators can be considered to be a simple model of thrust generating flagella. This type of actuation is also one of the simplest methods of locomotion in 2D environment and is very similar to the wheels driven Khepera mobile robot (Mondada et al., 1999), a cheap and popular platform that greatly contributed to the early development of the field of evolutionary robotics.

Switching the actuators off results in a motion continued due to inertia, until the fluid drag stops the animat. Steering the animat toward a particle requires not only orienting towards the food source, but also properly dealing with inertia while taking turns. In fact, at the early stages in many evolutionary runs, animats were observed to frequently overshoot the target by performing turns with too high speed.

The state of the animat at a given time step is represented by its position, direction, velocity $\vec{v}$ and the speed of angular rotation. Animats are subjected to a fluid drag $F_d = c_d d|\vec{v}|^2$, where $d$ is the diameter of an animat and $c_d$ is the drag coefficient. Since this type of drag acts only when an animat moves against the fluid, it does not act on animat's rotation. To prevent endless spinning, an additional rotation drag force (torque) is present in the system and acts against rotation of the animat.

Each food or poison particle is a point source of a chemical signal, which instantly diffuses into environment, generating a field of scent. The scent coming from a particle is proportional to the inverse of the distance to the source. The fields generated by all particles sum up and can be presented as a map of scent intensity (Fig. 5.2).

Whenever an animat gets in direct contact with a particle, the particle is consumed and removed from a map, together with the field of scent it generates (for the sake of simplicity, this happens immediately and temporal effects of diffusion are not simulated). Since only a scalar value representing scent intensity is perceived by the sensors, the direction of the gradient has to be extracted by comparing the difference between the two sensors and/or from the change over time during movement.

## 5.2 Sensors and actuators

The information from the sensors to GRN and from GRN to actuators is passed exactly in the same manner as input and output signals described in Chapter 4, i.e. through external factors and effectors defined in the genome (Table 5.1).

The only information about the environment that can be sensed by the animat comes from its two sensors $S_L$ and $S_R$. As the TF concentrations in the employed GRN model are in the range $[0, 1]$, whereas the strength of scent in the environ-

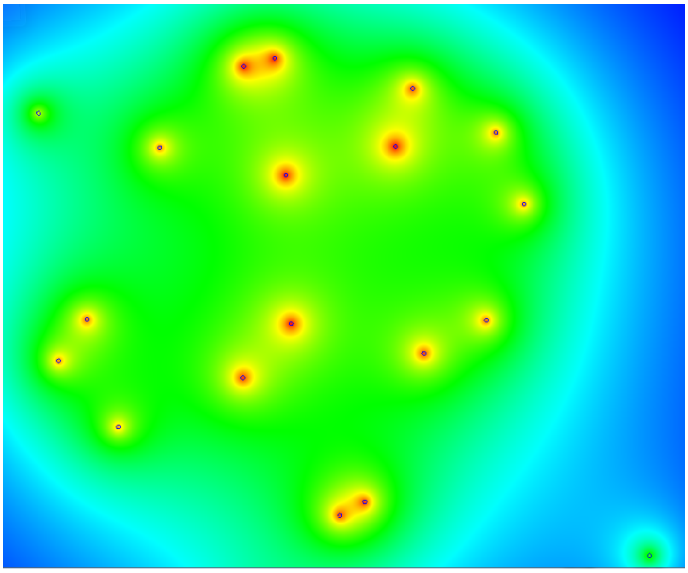


**Figure 5.2:** A fragment (the environment is open) of an example map of scent intensity that is locally perceived by animat's sensors. All food particles visible on this map have the same value and emit scent with the same strength. Colour map was normalized with blue representing zero, red maximum and green intermediate values of perceived scent levels.

ment can be arbitrarily high, some form of preprocessing of sensory information is necessary. The initial approach employed was to provide GRN with concentrations of input products $S_1$ and $S_2$ that would correspond directly to the values of $S_L$ and $S_R$ but were restricted to $[0, 1]$ with a sigmoidal function. In principle, this could have allowed for the evolution of simple controllers with sensors and actuators cross wired in the regulatory network, similarly to the controllers of Braitenberg vehicles (Braitenberg, 1986).

**Table 5.1:** Types of products and promoters enabled in the experiments on evolving GRNs for chemotaxis and the interpretations of subsequent input and output elements. $S_3, S_4, S_5$ where enabled only in the experiments with poisonous substances (section 5.6).

| **Promoter types** | **Product types** | **External factors** | **Effectors** |
|---|---|---|---|
| additive | transcription factor | "1" (fixed high concentration)<br>$S_1$ (a function of $S_R - S_L$)<br>$S_2$ (a function of $S_R + S_L$) | $A_L$ (left actuator)<br>$A_R$ (right actuator) |
| | | $S_3, S_4$ (same as above for the second type of food)<br>$S_5$ (signals the need to switch to a new food source) | |

However, initial experiments have shown that such signal preprocessing leads to very poor evolvability. The reason is that the diameter of the animat is very small compared to the steepness of the gradients in the environment, so both sensors perceive the scent at a very similar level. Unless the animat is very close to a food particle, the difference in signal levels is often less than 1%, making the use of this difference difficult. Although some animats capable to climb up the scent gradients were observed, their overall performance was poor. Similar reasoning explains why some bacteria (e.g., *E. coli*) employ only a single sensor and actuator and randomly

change the direction of motion: their size is too small to observe the gradients by measuring concentrations on each side of the cell. However, an order of magnitude larger eukaryotic cells are known to be able to detect gradients by employing sensors located on opposing sides of their cell (Alon, 2006).

Because the initial experiments did not result in high performance, the way in which sensory information is provided was modified. Instead of passing the value of activation $S_L$ and $S_R$ through a sigmoidal function, in the remaining simulations described in this chapter, the sensory information was presented to GRN through external factors whose concentration was determined by a sigmoidal function of a difference between $S_L$ and $S_R$:

$$S_1 = \frac{1}{1 + e^{-\alpha(S_R - S_L)}} \tag{5.1}$$

where $\alpha$ controls the steepness of the function, i.e., the amount by which small differences between $S_L$ and $S_R$ are amplified ($\alpha = 10$ was used). If $S_L$ is equal to $S_R$, the concentration of $S_1$ is 0.5. The concentration approaches 1 or 0 depending on the difference between $S_L$ and $S_R$.

Although using just $S_1$ was enough to evolve animats that actively search for particles, this way of providing sensory information does not allow to immediately infer the distance from the source. Because efficient turning requires taking the distance into account and the detection of direction of motion relative to the source, a second input product was introduced, with concentration dependent on the overall level of food scent at the animat location:

$$S_2 = \frac{2}{1 + e^{-\beta(S_R + S_L)}} - 1 \tag{5.2}$$

where $\beta = 1$ was used.

The thrust force generated by two actuators on the back of the animat is proportional to the concentration of a product associated with the effectors $A_L$ and $A_R$.

## 5.3 Fitness function

Each GRN was required to control an animat in its environment for a predefined number of simulation steps. A fixed number of food particles was deposited in the environment by drawing their positions from uniform distribution over the square area of size 45 (with animat diameter equal to 1). Fitness function was constructed to reward the amount of food gathered by an animat from its environment. An animat would start at the centre of the square area with randomized food particles, with the stored energy level of 0 and each consumed particle would increase this level by 1. The costs of movement were not taken into account in the following experiments.

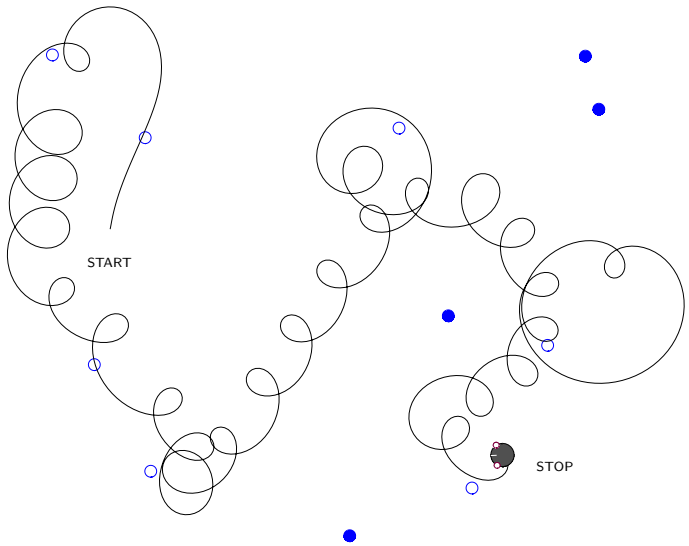



**Figure 5.3:** A common suboptimal solution in the fitness landscape of chemotaxing individuals: targeting food particles by performing circular motion. Despite low average speed, it can be quite effective at gathering particles. Particles consumed during lifetime are drawn as empty circles.

The genetic algorithm was set to minimize the following fitness function:

$$f_{fitness} = (1 - \frac{food}{food_{max}}) \cdot b \tag{5.3}$$

The $b$ term provides additional reward (*bonus*) to individuals that were observed to change the direction of their movement at least once during their lifetime (for them, $b$ was set to 0.9, otherwise it was equal to 1). The term was introduced after preliminary experiments when it was observed that the best animats would often evolve the same suboptimal behaviour that relied on circling towards food particles by always performing solely left or right turns (Fig. 5.3). The corresponding hill in the fitness landscape (here in an intuitive sense: more fit solutions occupying higher ground) is very easy to find and climb, but difficult to escape from: simply circling around a map allows to find some food particles by chance and the behaviour can be further optimized by evolving control over the diameter of the loop with only a single actuator (tightening the loops when the scent level increases). Thus, additional reward helps GA escape from this local optima and evolve more efficient control by promoting activity of the both actuators early on. Even so, as will be shown later, circling behaviour remains a strong attractor for the genetic algorithm.

Another issue that had to be addressed was map randomization. If the same map was used during the whole run of genetic algorithm, individuals overfit to this map would evolve. Such individuals would simply follow trajectories optimized for a particular map and fail or present greatly reduced fitness on any other map. To increase evolutionary pressure on general solution to the presented problem, each animat was evaluated on a randomized map: the number of particles would remain fixed, but their locations were always random. One side effect of this approach is that the fitness value is no longer deterministic and differs whenever it is recalculated

**Table 5.2:** Essential GA parameters used in the experiments on evolving GRNs for chemotaxis. Additional parameters are provided in the Appendix (Table C.2, p. 184)

| Parameter | Value |
|---|---|
| Population size | 300 |
| Elite individuals | 0 |
| Asexually created individuals | 300 |
| Individuals through crossover | 0 |
| Initial population | randomized genomes, 5 regulatory units each |
| Termination condition | 5000 generations |
| Selection | binary tournament ($k = 2, p = 1$) |

for the same genome. To reduce the effect of pure chance on fitness, each animat was evaluated on 4 random maps, and the average obtained fitness ($f_{avg}$) was used by the GA.

## 5.4 Genetic algorithm

Table 5.2 provides a summary of essential parameters used in the following experiments. The initial population was randomized just as in signal processing experiments (see Table C.2, p. 184 of the Appendix for details). Binary tournament selection (select two individuals from the population, keep the better one) was used. Elitism was disabled, so that the networks would not only need to optimize their fitness but also would have to evolve a level of robustness to mutations, hence creating a more biologically realistic evolutionary pressure. At the same time, the probability of deletions and duplications was set to be about an order of magnitude lower than in experiments in evolving signal processing networks (Chapter 4), since without elitism, any good solution could be too easily lost due to high mutations rates. Furthermore, recombination between genomes was disabled. This created conditions in which genetic elements could not be created *de novo* and all genetic elements in any individual were a result of duplication and divergence of elements from initial random genomes.

## 5.5 Foraging with a single type of food

In the first of the two presented experimental settings, the environment was created by placing 20 food particles at random locations. Then, each evaluated individual was simulated for 2000 time steps of its GRN activity. The scale of the physical environment was set so that the typical time necessary to cover the distance between furthest food particles, assuming both actuators were fully activated, would be in the range of 300 time steps. Because about 25 steps are needed for any TF to degrade from its full concentration (1) to 0.1, latencies in information processing in the GRN can noticeably impact reaction time in the physical simulation. This could be easily controlled by changing the ratio of GRN simulation steps to physics simulations steps (here 1:1), but the settings were chosen to introduce some evolutionary pressure on faster network reaction times.

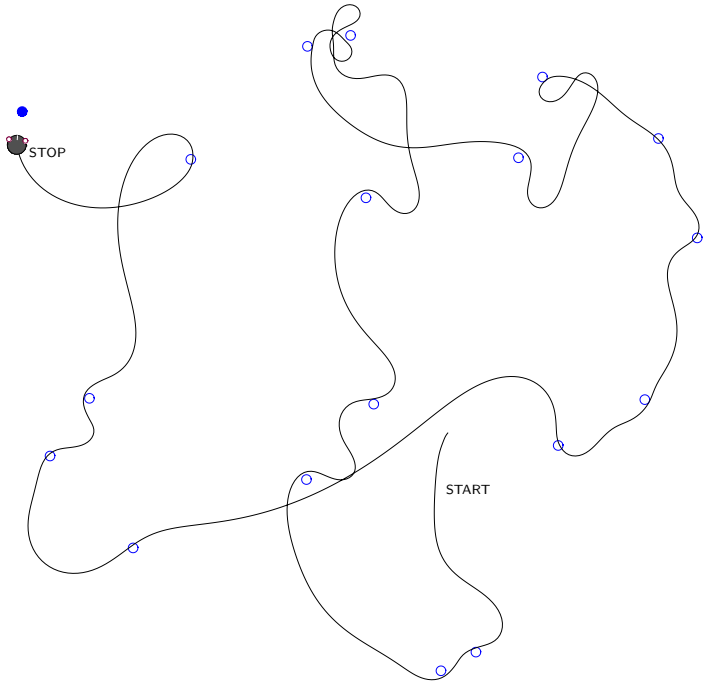



**Figure 5.4:** The best individual obtained in 10 evolutionary runs navigating a map with a single type of food source. Consumed particles are drawn as empty circles. Visualization of initial scent intensity on this map was earlier shown in Fig. 5.2.

Out of 10 independent evolutionary experiments (each using a different random seed), 7 runs resulted in solutions capable to gather around 75-95% food particles from the map (fitnesses between 0.05 and 0.25). The individuals obtained in remaining 3 runs would also target food sources, but do so by moving in loops that tighten near the food source (as in Fig. 5.3). This allowed them to consume around 30-40% food particles from the map. This behaviour is an example of the local optima in the fitness landscape discussed earlier (section 5.3).

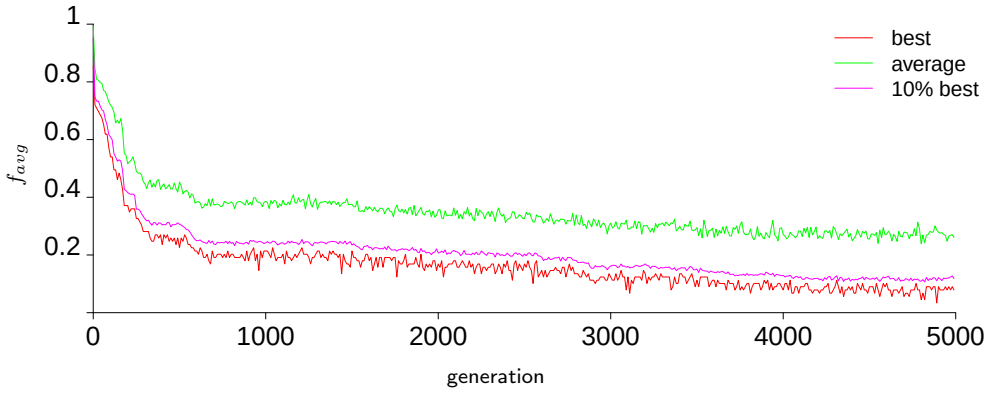
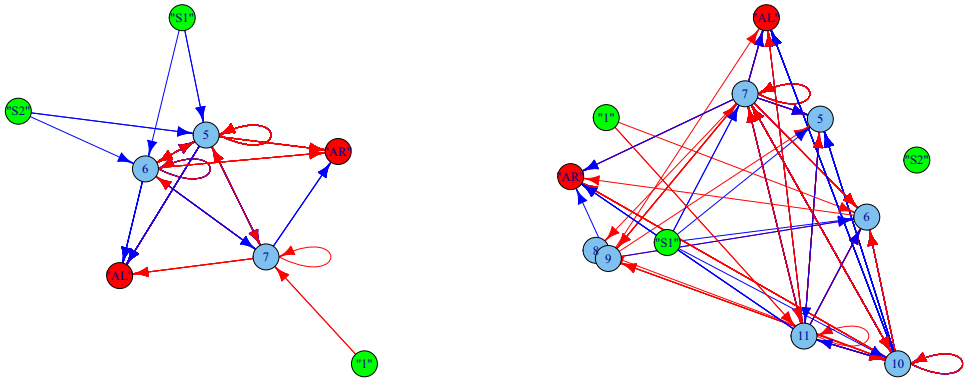



**Figure 5.5:** Fitness over generations for the problem with a single type of food source. The graph shows the history of the evolutionary run that resulted in the best performing individual out of 10 (its behaviour is shown in Fig. 5.4). Best and average fitness in every generation is shown and an average for 10% of the best individuals. Data points sampled every 10th generation.




(a) (b)

**Figure 5.6:** Topologies of evolved GRNs controlling behaviour in a problem with a single type of food source. (a) GRN of the best animat (generation 5000), (b) its ancestor in generation 3000. Multiple links between nodes have been reduced to a single line, red edges have positive weights, blue have negative.

### 5.5.1 Analysis of evolutionary history

In the 10 independent evolutionary runs, the capability to move towards food source was observed to evolve relatively early, during first few hundred generations (Fig. 5.5). Further evolution resulted in increasing the initially very slow speed of locomotion and improved targeting. The navigation of the best obtained individual is highly efficient (Fig. 5.4)[1], although it moves at around 60% of maximum speed and sometimes overshoots its target. However, this an expected trade-off given the inertia introduced by physics and latencies of the regulatory network (limited speed of product synthesis/degradation).

Visualization of the evolved regulatory network of this individual (Fig. 5.6a) shows a simple and largely symmetric topology with only 3 internal nodes. The GRN uses both the directional information ($S_1$) and the current level of scent concentration ($S_2$). The use of the latter is not critical for navigation: it was often seen disconnected in less fit solutions. Indeed, the ancestor of this individual from generation 3000 that was obtained by tracing the full evolutionary history of the final individual (Fig. 5.6b) does not make use of $S_2$ (it is disconnected from its GRN). It is very likely that the incorporation of this additional information was one of the sources of later improvement in navigation: notice that all connections from $S_2$ in Fig. 5.6a are inhibitory, hence most likely result in speed being reduced when close to a food particle.

During 2000 generations that separate the two individuals, the network has become smaller and less dense. That happened despite the size of the genome growing by around 70% (Fig. 5.7). Analysis of all mutations that occurred on the evolutionary path between the two individuals revealed that the number of duplications and deletions that occurred during that time was similar (6 and 5, respectively). However, the duplications were much longer (an average of 8.3 genetic elements, std dev: 3.4) with deletions having an average length of 2.3 (std dev: 2.1), despite

[1]Supplementary videos of animat behaviors are available at: `http://www.evosys.org/alife12chemotaxis/`

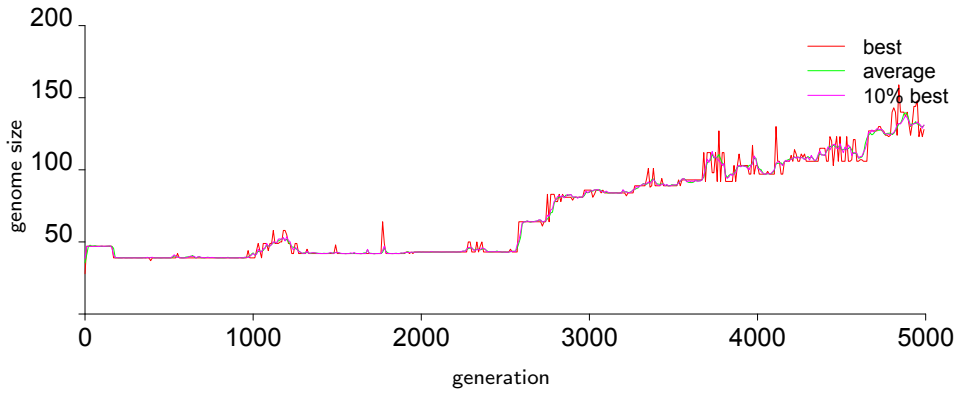



**Figure 5.7:** Genome size over generations for the problem with a single type of food source. The graph shows the history of evolutionary run that resulted in the best performing individual out of 10 (with fitness history shown in Fig. 5.5). Genome size for the best individual, an average of all individuals and an average of 10% most fit individuals in a population is shown. Data points sampled every 10th generation.

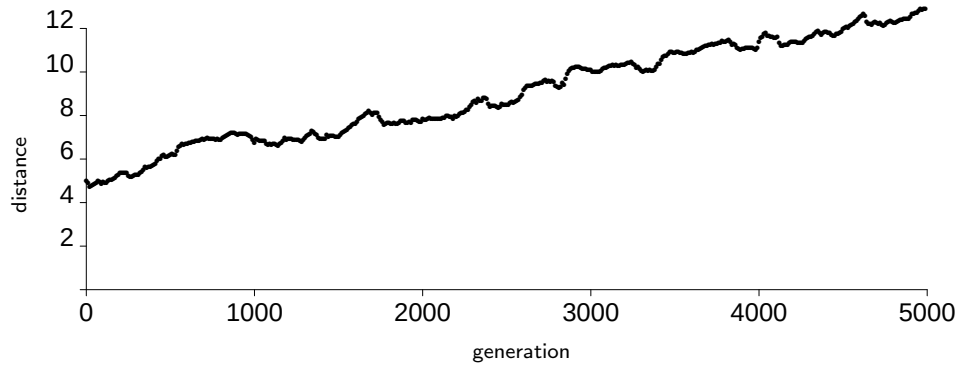



**Figure 5.8:** Spread of genetic elements in the genomes over evolutionary time for the problem with a single type of food source. Average distance from $(0, 0)$ of all genetic elements in all genomes in the population is plotted. The graph shows the history of evolutionary run that resulted in the best performing individual out of 10 (with fitness history shown in Fig. 5.5).

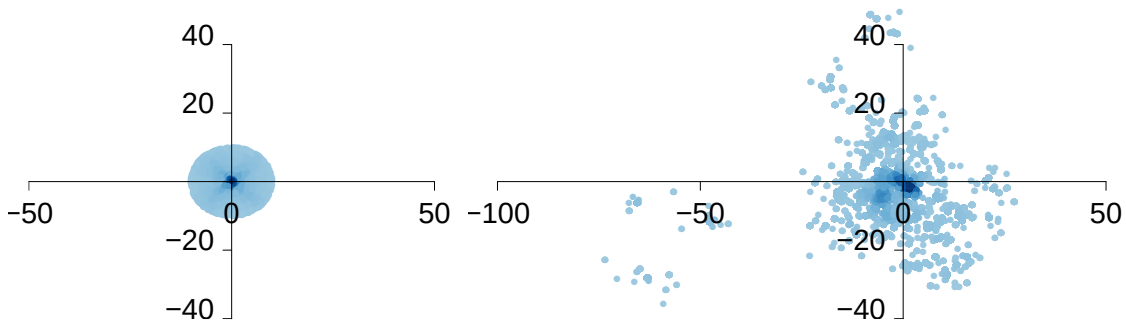



**Figure 5.9:** Comparison of the distribution of genetic elements from all individuals in first generation (left) and last generation (right) for the problem with a single type of food source. Dots represent locations in $\mathbb{R}^2$ of all genetic elements in the gene pool. Same evolutionary run as in Fig. 5.8.

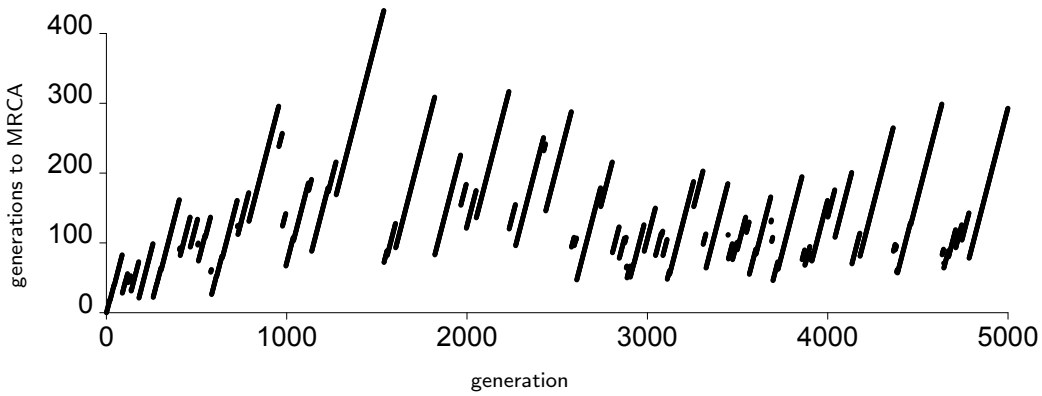



**Figure 5.10:** The number of generations from the most recent common ancestor (MRCA) for the entire population in each generation of the experiment with a single type of food source. Average: 148.7. The graph shows the history of evolutionary run that resulted in the best performing individual out of 10 (with fitness history shown in Fig. 5.5).

their randomized length being drawn from the same geometric distribution with an expected value of 10. The optimized structure of the final network is thus most likely not due to deletions but results from the coordinate mutations that occur to genetic elements over time.

Two processes are likely to stimulate optimization of an initially dense network. First, given latencies of product degradation and accumulation, it is advantageous to have a shorter signal path between sensors and actuators. The second process is the genetic drift caused by small coordinate mutations of genetic elements. Each such mutation results in a small change to weights of affected connections in regulatory network (and in biology is analogous to single nucleotide mutations in regulatory regions of DNA to which TFs bind). Small mutations have higher likelihood of remaining neutral or having an effect not significant enough to be picked up by positive or purifying selection. Over time, this leads to genetic elements spreading in the coordinates space (Fig. 5.8 and Fig. 5.9). Hence, unless their position is sustained by evolutionary pressure, all genetic elements perform a random walk in coordinate space, with elements that contribute little to the fitness drifting slowly beyond the interaction distance and reducing the overall density of the network. Similar process is known to occur during biological evolution: neutral mutations in duplicated genes or regulatory regions (now free from selection pressure) will lead to removal of redundant connections from biological GRNs.

Analysis of the number of generations separating all 300 individuals in a given generation from their most recent common ancestor (Fig. 5.10) reveals that it existed on average about 150 generations earlier. This means that all individuals in the final generation represent only a single successful lineage, rather than multiple independently evolving and competing lineages, and whenever mutation results in an improved individual, its descendants quickly take over the whole population. This suggests high homogeneity of the population and is further reinforced by the observation that the average of any measured genome parameter usually closely follows the value for the best individual.

## 5.6 Environment with food and poison

In order to create a more challenging environment for simulated evolution, a second type of food was introduced. In this scenario, one type of food would initially be edible and the other poisonous. At some point, their roles would swap. Consuming poison particles decreases the stored energy level and going below 0 is lethal for an animat. The switch occurs after a certain number of particles is stored internally by an animat, and the event is signalled with another input of the GRN. The purpose of such problem design was to see if it is possible to evolve GRNs that display radically different behaviour for the same sensory signals, depending on a single external switch. Whether this could be achieved is interesting from the point of view of scalability of GRN driven control to more complex environments, where different behaviours are required based on the same sensory data.

To allow sensing of the second type of food, a pair of two additional external factors $S_3$ and $S_4$ was added. They would behave in the same manner as $S_1$ and $S_2$ (equations 5.1 and 5.2), but react only to the second type of particles. The environment was randomly filled with 30 food particles of one type (blue) and 30 of the second type (red, initially poison). Each consumed food particle would increase energy level by 1, whereas each poisonous particle would decrease it by 1. Animat whose energy dropped below zero was immobilized. The poison would change into food and vice versa after the energy level reached 5 particles. Hence, the life of each individual became split into two phases: first, when it would have to gather blue particles while avoiding collecting red ones, and a second phase, when it would have to collect red ones and avoid blue ones. The higher density of particles (total of 60) compared to previous experiment (20) was found to be necessary for poison avoidance to evolve: if poisonous particles were too sparse, accidental consumption was too rare to negatively affect the fitness and the avoidance behaviour did not evolve.

The information about the need to switch behaviour was provided with another external input $S_5$. The concentration of $S_5$ would switch from 0 to 1 as soon as the stored energy level reached 5. Providing this signal was necessary, as the animats do not have any knowledge of their actual stored energy level and thus do not know when they should start looking for a new food source. A preliminary run of genetic algorithm without this signal resulted in animats that would move slowly in the environment, so that they would collect only about 5 food particles during their lifetime.

Ten independent evolutionary runs were performed, using the same GA configuration as in section 5.5, but with the lifespan of individuals extended to 7000 time steps to allow for collection of a larger amount of particles. Three runs resulted in best individuals with $f_{avg}$ between 0.19 and 0.26, thus extracting around 75% of accessible energy on the map. The animats show the desired behaviour, that is, they seek the first food source and switch to the second as soon as $S_5$ becomes 1. Another 3 runs resulted in individuals that gather only the blue particles and then stop. The remaining 4 runs resulted in individuals that gather around 50% of

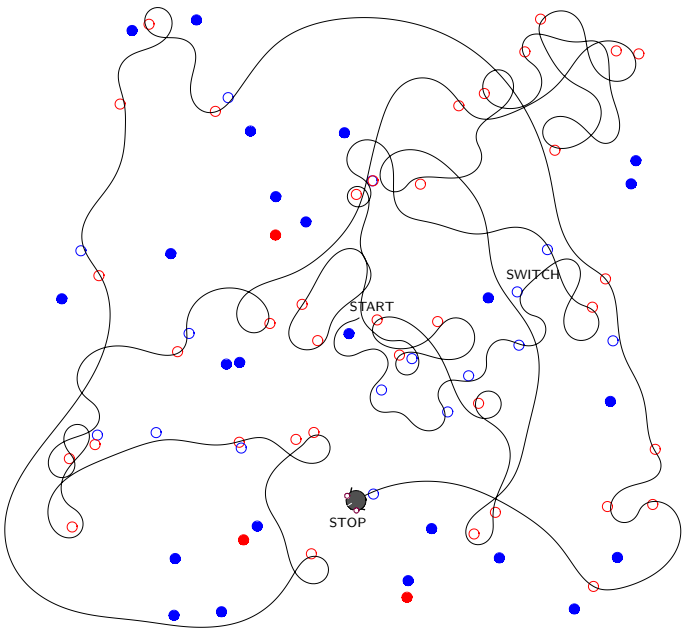



**Figure 5.11:** Behaviour of the best individual from the final generation (5000) for the problem with two switching types of food source. The switch between which food type is poisonous occurred after 5 blue particles were consumed. Consumed particles are marked as empty circles. The best individual obtained in 10 independent runs is shown.

food by efficiently collecting the blue particles and then gathering the red ones in a circular motion (Fig. 5.13), a manifestation of the attractor in the fitness landscape discussed earlier (see Fig. 5.3, p. 103).

### 5.6.1 Analysis of evolutionary history

Close inspection of the behaviour of the best individual obtained in 10 experiments (Fig. 5.11) reveals that it actively avoids red food particles while searching for the blue ones. However, after the behaviour switch is signalled, although it now actively seeks red particles, it consumes any blue ones that accidentally come its way. The difficulty to obtain additional avoidance behaviour in the second phase most likely can be explained by the fact that much lower evolutionary pressure exist for it. Accidental consumption of red particles at the beginning of animat's life (when it has energy of 0) is lethal. However, in the second phase, with energy level above 5, accidental consumption of blue particles does not lead to death. Thus, seeking only red particles and ignoring accidental consumption of now poisonous blue ones still leads to high fitness.

The analysis of the topology of the regulatory network also reveals that information from all externally provided signals ($S_1 - S_5$) is used (Fig. 5.12). The history of improvements in best and average fitness over generations shows that evolution was less gradual for this problem, with clearly visible stages (Fig. 5.14). Similar pattern was also observed in the remaining runs, although, naturally, the lengths of those stages would vary considerably. Inspecting the best individual at the first plateau (generation 2600) reveals that it is capable of seeking blue particles, while already

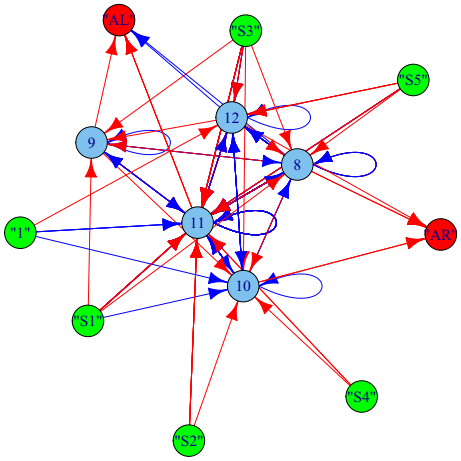



**Figure 5.12:** GRN topology of the best obtained animat for the problem with two switching types of food source. The behaviour of this individual is shown in Fig. 5.11. Multiple links between nodes have been reduced to a single line, red edges have positive weights, blue have negative.

actively avoiding red ones. However, after the signal to switch behaviour is activated, it only performs circular motion collecting particles at random (Fig. 5.13). This behaviour, on average, still results in a net gain of energy as there are now slightly more particles of the edible type of food left on the map. The best individual from generation 3100 is already capable of actively seeking the second type of food source, but does so very slowly. The third plateau in fitness is reached by successively improving the speed of navigation in the second phase of life.

The large improvement in fitness that occurred between generation 2900 and 3900 (Fig. 5.14) corresponds to an earlier minor increase in genome size (Fig. 5.15). The duplications that caused it did not create new nodes and the number of nodes would not change much during the evolution, dropping in the beginning and then increasing by one around every thousand of generations (Fig. 5.16). Hence, the duplications would result in the addition of new connections between regulatory units and mostly secondary to the already existing ones, thus changing the strength of the influence between the units that are connected. However, the fact that gene duplications result mostly in creation of new edges in a graph should not be surprising. To create a new connection, it is enough to duplicate a single product or promoter, whereas the creation of a new node requires creation of a pair of subsequent genetic elements encoding promoter and a product (see section 3.1.1, p. 64). Interestingly, observed two episodes of quick improvement in fitness cannot be causally linked with duplications observed during that time (arrows in Fig. 5.15), because these duplications did not directly precede these episodes. The duplications occurred either during these episodes or during the plateaus that separate them. This suggests that although duplications most likely provided the genetic resource for later improvement, the actual improvements occurred when already existing and possibly recently added genetic elements were acquiring new functions.

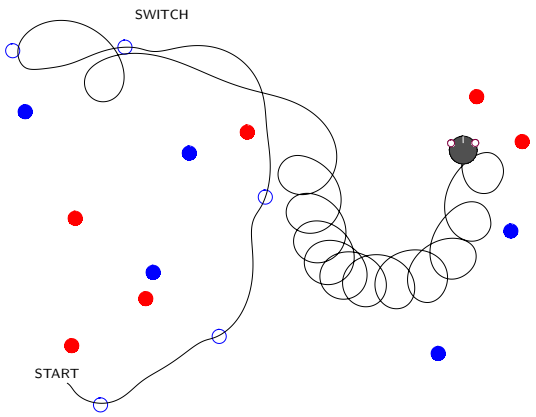



**Figure 5.13:** Trajectory of the best individual from generation 2600 for the problem with two switching types of food sources from the evolutionary run that resulted in the best performing individual (its behaviour shown in Fig. 5.11). After seeking blue particles, the animat switches to circular motion strategy, similar to that observed in the previous experiment (compare with Fig. 5.3). Consumed particles are drawn as empty circles.

## 5.7 Summary

The results of this chapter demonstrate how the model of regulatory network encoded in linear genomes can be applied to a control problem and used to evolve behaviours of an animat in a simulated physical world. The evolution would start with random genomes, growing over time through duplication of chunks of the genome and further divergence of duplicated elements. The evolvability was investigated by introducing a more complex foraging problem, when more than a single navigating behaviour would have to be presented by an animat and different behaviours would have to be employed for the same sensory input, depending on a single switch. This means that more complex tasks, including obstacle avoidance and foraging for multiple different resources, e.g., food and water, are possible (and, in fact, the latter was recently successfully demonstrated in a later work, Wróbel et al., 2012b). A more biologically realistic setting for the evolution of regulatory networks introduced in this chapter was used to investigate how the genomes and networks evolve and self optimize over evolutionary time. The system also served as a test bed for the creation of an open ended alife system, in which multiple individuals will compete for resources and which is currently under development (Erdei et al., 2011).

One of the unexpected results of the current configuration of the model was that the concentrations of TFs in the final generation turned out to be kept at low levels, mostly below 0.3. This may have been one of the approaches chosen by evolution to reduce the latencies in processing with genetic elements. Although the drop of TF concentration over time is exponential and so the relative drop in time $\delta t$ is independent on product concentration, the synthesis rate and increased degradation resulting from gene regulation has a fixed maximum efficiency. This means that the effect of relative change of concentration of a given TF coming from regulation is larger when the concentration is low. Performing the computation using low concentrations allows thus to react faster to changing environmental signals.

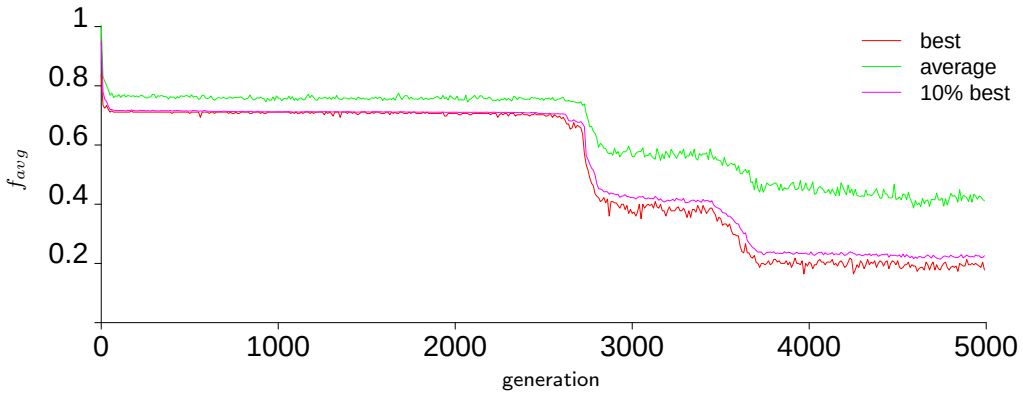



**Figure 5.14:** Fitness over generations for the problem with two switching types of food source. Three stages corresponding to improved behaviour are seen. The graph shows the history of evolutionary run that resulted in the best performing individual out of 10 (its behaviour is shown on Fig. 5.11). Best and average fitness in every generation is shown and an average for 10% best individuals. Data points sampled every 10th generation.

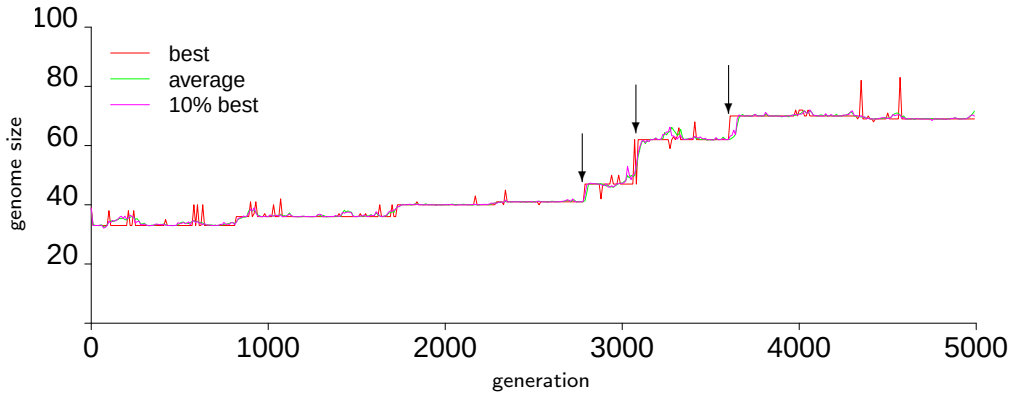



**Figure 5.15:** Genome size over generations for the problem with two switching types of food source. Arrows mark three larger duplications. The graph shows the history of the same run as in Fig. 5.14. Best and average fitness in every generation is shown and an average for 10% best individuals. Data points sampled every 10th generation.

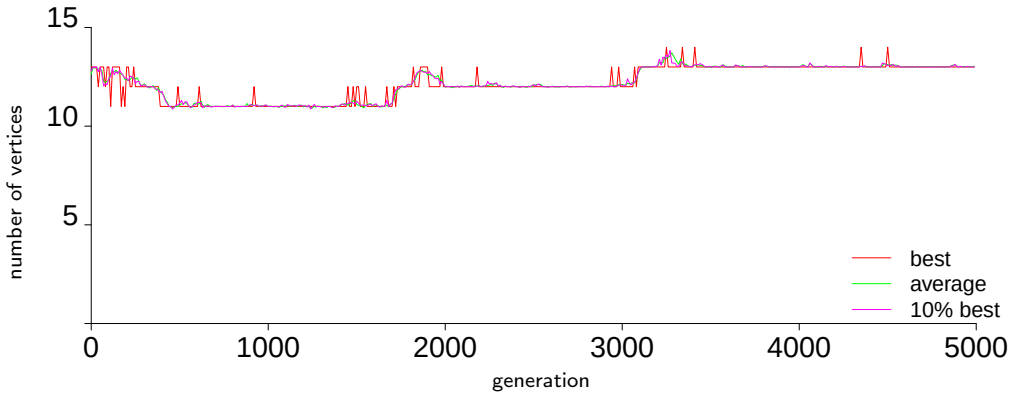



**Figure 5.16:** The number of nodes in the GRN during evolution for the problem with two switching types of food source. The graph shows the history of the same run as in Fig. 5.14. Best and average fitness in every generation is shown and an average for 10% best individuals. Data points sampled every 10th generation.

Certainly, this must be used to some extent by biological systems as well, but in a real world, lowering concentration will result in decreased signal-to-noise ration. Thus, for biological regulatory networks with noisy gene expressions, a trade off has to be made, which was not a case in the presented experiments. It would be interesting to investigate how addition of intrinsic noise of gene expression will affect the way evolved networks process and encode information and balance the signal-to-noise ratio.

# Chapter 6

# Evolution of multicellular development

This chapter presents the results of application of the model of gene regulatory network described in Chapter 3 to the control of 3D artificial embryogenesis.

Most of the models of artificial embryogenesis existing prior to this work rely on the assumption that each cell of a developing embryo is placed on a uniform grid, with some notable exceptions, such as the works based on Cellular Pots Model (such as Hogeweg, 1999, 2000; Knabe et al., 2008b), in which a cell of the embryo consists of multiple grid cells. The representation of a cell of a virtual embryo with a single, filled cell of a grid means that each cell is represented as a square or a hexagon in 2D or a cube in 3D. The actual shape of cells is not usually relevant: the type of the grid influences only the number of neighbours of each cell, which translates into the degree of freedom for placement of new cells during division as well as cell movement. The use of a grid is a popular modelling approach because it has numerous computational advantages. Grids are efficient to simulate since interactions between cells (such as the diffusion of substances between them) occur locally, between clearly defined grid neighbours. Also, typically, in grid based models, the embryo is not allowed to move in relation to the grid, as this would remove most of the computational advantages. Furthermore, as the main decision that each dividing cell has to perform is the direction in which the daughter cell will be placed, the limited number of possible locations simplifies the simulated development both computationally and from the point of view of the evolutionary search space.

On the other hand, the use of a grid creates a system that, from the very start, is far from biological realism. Immobile cells remove one of the key features of the developmental process, that is the physical interactions between cells in a developing embryo, their movement and reorientation in space, as well as their ability to vary in size and shape. Naturally, allowing for these effects comes at higher computational costs.

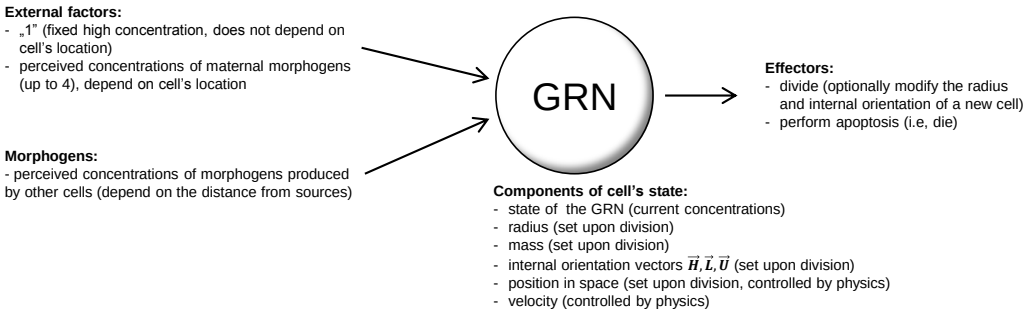



**Figure 6.1:** A summary of cell interactions with environment and properties of cell that constitute its state during development.

## 6.1 Developmental model

One of the goals of the work described in this chapter was to free the developmental process from the restrictions of a grid and to represent cells as objects in a continuous three dimensional space, interacting through simulated physics. In the developmental model introduced in this chapter, cells are represented as elastic, spherical objects which can (if desired) vary in size. Their motion is controlled by simple Newtonian physics with simulated fluid drag.

Developmental process starts from a single cell. Each cell is controlled by a gene regulatory network. When a cellular effector (i.e., a special transcription factor) crosses a predefined threshold of concentration, a cell will divide. The daughter cell will inherit the exact copy of the genome from the mother, together with the current state of regulatory network (that is, the concentrations of all TFs at the time of division). Cells can, however, differentiate from each other because the physical state of the daughter cell (that is its orientation and position in 3D space) can be different and leads to perceiving different signals from the environment. Finally, apart from producing TFs that regulate cell behaviour, cells can produce morphogens that diffuse into the environment. Morphogens can bind to promoters in other cells and influence their behaviour. Cells are also capable of performing apoptosis (i.e., programmed cell death) if the concentration of the corresponding effector crosses a predefined threshold.

The following sections provide the details of the developmental model. An initial outline of what constitutes the state of a cell and its input and output signals is provided in Fig. 6.1.

### 6.1.1 Configuration of the genome model

To facilitate multicellular development, new types of effectors and external factors were introduced to the model of the genome presented in Section 3.1.2 (p. 65). The external factors can now include up to 4 maternal morphogen gradients (Table 6.1). They represent substances (maternal factors) that are predeposited in the environment in which the embryo starts to develop (e.g., the egg). Such substances are known to be essential for the development of many organisms and their role was particularly well studied in the development of the *Drosophila melanogaster* (the

**Table 6.1:** External factors (inputs of the GRN) available in the developmental model.

| External factor | Description |
|---|---|
| "1" | a product that is perceived at the constant level of 1 |
| $P_1, P_2, P_3, P_4$ | morphogen gradients behaving as if they were morphogens diffusing from 4 fixed locations in space. Cells perceive them as products whose concentration falls off with the distance from the source. Exact location of each maternal morphogen source is defined at the beginning of the experiment and remains fixed during evolution |

**Table 6.2:** List of effectors (outputs of the GRN) available in the developmental model.

| Effector | Description |
|---|---|
| Divide | if above a preset threshold, a cell divides |
| Die | if above a preset threshold, a cell dies and is removed from an embryo |
| Change radius | increases the radius of a daughter cell beyond the default value upon division |
| $\alpha_H$, $\alpha_U$, $\alpha_L$ | 3 effectors controlling the amount by which internal cell orientation is rotated in the daughter cell after division |

common fruit fly), where their gradients determine the polarity of the egg and the embryo (see, e.g., Carroll et al., 2004). By sensing the concentration of such substance, a cell can determine its approximate location in space. The diffusion of maternal morphogens in the presented system is not simulated explicitly. Instead, they are assumed to form a static gradient of concentration in space and are perceived by cells simply as an external factor whose concentration falls off with the distance from a predefined point in 3D space (the "source" of this morphogen).

The new cellular effectors are related to the actions that can be taken by cells (Table 6.2) and their detailed explanation is provided in section 6.1.4. Depending on the experiment, actions that are not essential to development, such as modification of cell size or apoptosis, could be disabled.

To allow for communication between cells, a new type of genetic elements was introduced: morphogens. They are treated as gene products, i.e., similarly to transcription factors. A single promoter followed by a single morphogen in the genome can form a regulatory unit and thus a node in a regulatory graph (see Fig. 3.1, p. 64). Just like TFs, each morphogen has a concentration (from 0 to 1) associated with a cell in which it is produced. However, contrary to TFs, the range of action of morphogens extends beyond the cell they are produced in: they diffuse into the environment and can influence promoters of other cells. The process of diffusion is simulated in a simplified manner, and morphogen molecules are not simulated explicitly. Instead, cells perceive morphogens as products, whose concentrations diminish with a distance from the cells they are produced in. If multiple sources of a morphogen exist, their effect on promoters sums up. A more detailed explanation is provided in section 6.1.3. Since morphogens are encoded in the same manner as TFs, it is left up to the evolution how many and whether any at all will be used by the embryo.

### 6.1.2 Simulated physics

As mentioned earlier, cells are represented as spheres and can vary in size within a single embryo. Each cell has associated real-valued coordinates in 3D space, a radius, a mass and velocity. At every time step of physics simulation, force vectors acting upon cells are calculated and their sum is used to obtain cell accelerations according to Newtonian laws. Accelerations are used to update velocities and velocities are used to update cell positions. The next time step of simulation is calculated using the Runge-Kutta 4th order integrator, due to its higher precision and increased stability of the physics over the classical Euler's method (see, e.g., discussion in Bourg, 2001).

Three types of forces act on the cells during development: repulsive force, adhesive force and fluid drag. The repulsive force is calculated whenever two cells overlap with each other, i.e., when the distance between their centres is smaller than the sum of their radii. The force acts on both cells in the direction defined by their centres (with an opposite sign for each cell), pushing the cells away, with the value proportional to the square of the overlap, formally:

$$F_r = \begin{cases} 0, & \text{if } ||P_1 - P_2|| \geq r_1 + r_2 \\ c_r\ (||P_1 - P_2|| - r_1 - r_2)^2, & \text{if } ||P_1 - P_2|| < r_1 + r_2 \end{cases} \tag{6.1}$$

where $P_1, P_2$ are of positions cells' centres in 3D space, $r_1, r_2$ are their radii and $c_r$ is the repulsion coefficient ($c_r = 5$ was used).

To maintain coherent structure of the embryo and prevent cells from disconnecting from the embryo, an adhesive force acts between cells. The adhesive force, like the repulsive force, acts along the direction of cell centres, but has an opposite sign. It is only non zero for any pair of cells whose surfaces are within small interaction distance equal to $d_a \cdot min(r_1, r_2)$ and only when cells do not overlap.

$$F_a = \begin{cases} 0, & \text{if } ||P_1 - P_2|| - r_1 - r_2 \geq d_a \min(r_1, r_2) \\ 0, & \text{if } ||P_1 - P_2|| - r_1 - r_2 \leq 0 \\ c_a\ (||P_1 - P_2|| - r_1 - r_2 - d_a \min(r_1, r_2))^2, & \text{otherwise} \end{cases} \tag{6.2}$$

where $c_a$ is an adhesion coefficient ($c_a = 1$ was used). Adhesive force was active as long as the cells remained within 1/4 of the radius of the smaller cell ($d_a = 0.25$).

To prevent erratic cell movements as well as oscillations caused by interactions of the repulsive and the adhesive force, all motion is dampened with a drag force that is proportional to the velocity of a cell. It can be thought of as an effect of fluid viscosity:

$$F_d = c_k ||\vec{v}|| \tag{6.3}$$

where $c_k$ controls viscosity of the fluid ($c_k = 1$ was used). The force acts in the direction opposite to the cell's velocity $\vec{v}$ ($\vec{F_d} = -F_d \frac{\vec{v}}{||\vec{v}||}$).

A summary of the above and remaining essential parameters related to physics is provided in the Appendix (Table C.3, p. 185). The visualizations of resulting

dynamics of cellular motion are provided in results section (see, e.g., Fig. 6.6, p. 126).

### 6.1.3 Morphogens and diffusion

The typical approach to simulate diffusion of substances in 3D space is to divide the space into regions, either uniform (cubical regions) or non uniform (e.g., using structures such as octrees), and then to iteratively simulate the flow of substances between adjacent regions of space (e.g., using the Fick's laws of diffusion). The precision of the simulation depends on the resolution of a grid, which is a trade-off between computational cost and desired accuracy of the simulation. Since the development in the presented model does not occur on a grid to avoid introducing grid just for the purpose of simulating diffusion, an alternative, simplified approach is used. It assumes that perceived concentration of a morphogen in a given position in space depends on the distance from a source cell and the concentration of this morphogen in this cell. To prevent information from propagating instantaneously in the system, a historic value of morphogen level at the source is used. Each morphogen source stores a complete history of its morphogen concentrations in previous time steps. The longer the distance, the older the value used (see the visual explanation in Fig. 6.2[1]). Formally, the concentration of the morphogen $m$ that is perceived by the cell $c$ at the simulation time step $t$ is:

$$L_{c,m}(t) = \sum_{i=1}^{I} l_{i,m}(t - \lfloor \frac{D_{i,c}}{d_s} \rfloor) \frac{1}{1 + D_{i,c}} \tag{6.4}$$

where $I$ denotes the number of cells of a developing embryo (potential producers of $m$), $D_{i,c}$ is the distance from the cell $c$ to the source cell $i$ in 3D space of the developing organism, and $l_{i,m}(t)$ is the concentration of the morphogen $m$ in the source cell at the time $t$. The value of $l_{i,m}$ is delayed in time: the past concentration from the simulation time step $t - \lfloor \frac{D_{i,c}}{d_s} \rfloor$ is used, where $d_s$ is the distance the information about morphogen propagates in a single simulation step ($d_s = 0.1$ in the experiments in this thesis).

Although such approach does not conserve mass and thus does not simulate diffusion realistically from the physical point of view, it has the essential spatiotemporal properties of diffusion, i.e., the perceived concentration is decreasing with the distance from the source and information about changes in concentration propagates through the system with a limited speed. In this way, it is similar, e.g., to the propagation of sound waves at large distances. It was chosen because of its relatively low simulation cost for small numbers of cells: the computational cost is proportional to the number of locations in space where concentrations are being read (i.e., the number of cells) and to the number of sources producing the morphogen (also the number of cells), thus has the computational complexity $O(n^2)$ (where $n$ is the number of cells). This means that for small numbers of cells (in the order of hundreds) it will still be relatively fast compared to simulating a uniform diffusion

[1] Video of this figure is available at: `http://youtu.be/l5_fBCR7ncM`

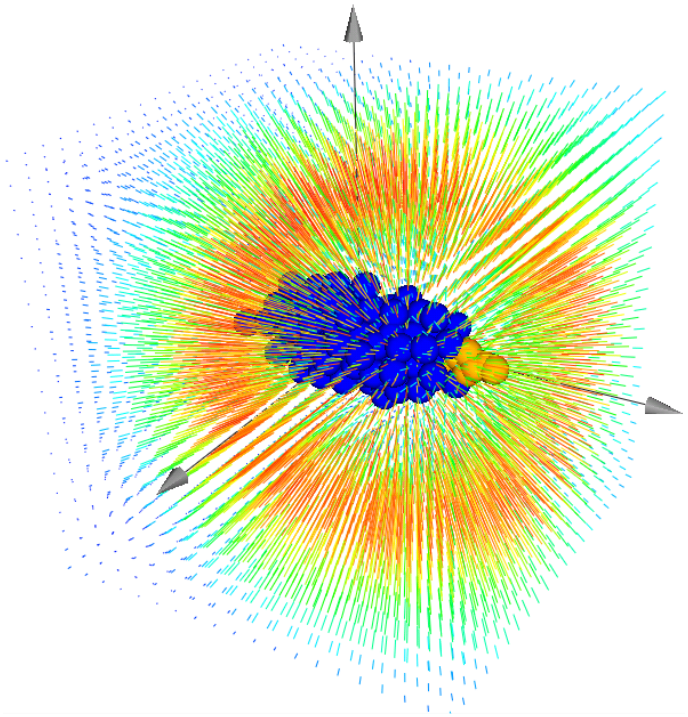


**Figure 6.2:** Visualization of the simplified diffusion model. Diffusion is modelled without the use of a grid, by using past values of morphogen expressions. In this example, an outburst of morphogen production in the far right region of the embryo, followed by its degradation soon after, resulted in a ring (red) of increased concentration travelling away from the source. Lines show the direction of the gradient, their colour shows the perceived concentration of this morphogen (blue - low, green to yellow - intermediate, red - high).

grid which, given moderate spatial resolution of 100, in 3D would already result in a need to update 1 million grid cells every time step. However, since the cost of a grid based system is less sensitive to the number of cells in the embryo, it should be preferred if the model is scaled up to much higher numbers of cells.

### 6.1.4 Cellular actions: division, death and growth

For the development to occur, each cell has to perform discrete actions such as division, based upon continuously changing product concentrations. Each action has a constant threshold (defined at the level of experiment) associated with it and is performed after the threshold concentration of corresponding effector (Table 6.2) is reached.

#### Cell division

When the effector responsible for division crosses a threshold, a cell divides. At this moment, a new cell is added to the system, containing a copy of the GRN, together with its current state, i.e., the concentrations of all products in the mother cell.

Each cell maintains a vector representing its orientation ($\vec{H}$) that determines direction of cellular division. At division, a daughter cell is placed in the direction $\vec{H}$ from the mother, at the distance of $\frac{1}{3}$ of mother's radius. This means that initially

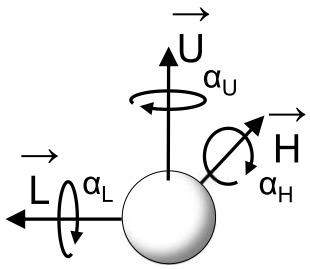



**Figure 6.3:** A vector $\vec{H}$ along with the two auxiliary perpendicular vectors for up and left ($\vec{U}$, $\vec{L}$) define the internal orientation of a cell in 3D space. Cells divide in the direction of $\vec{H}$ and can modify their internal orientation by rotating around any of these vectors.

the daughter overlaps with the mother and, in the following time steps after division, the forces of simulated physics will push them away, until equilibrium between the adhesive and repulsive force is reached. If the area in which the daughter cell is being placed is occupied by other cells, physical forces will continue to push away the cells until they no longer overlap. To prevent another division in the very next time step (since both cells now have the level of division effector above the threshold), the concentration of the effector responsible for division is set to 0 in both cells.

To allow cells to control the direction towards which they divide, a method of incrementally modifying orientation of a 3D vector is used. It is based on one of the approaches used in 3D L-systems for modelling plant development (Prusinkiewicz and Lindenmayer, 1996). Orientation is represented by three perpendicular unit vectors $\vec{H}, \vec{L}, \vec{U}$ indicating the heading, the direction to the left and the direction up of the internal orientation of a cell (Fig. 6.3). A cell can modify its internal orientation by rotating around each of the three vectors. This way, the rotations are always relative to the current internal orientation and do not depend on the absolute coordinate system. The angle of rotation is controlled by 3 cellular effectors (Table 6.2), whose concentrations are translated into an angle between $[0, 2\pi]$. Rotation of the current cell orientation by angles $\alpha_H, \alpha_L, \alpha_U$ is expressed by the equation:

$$[\vec{H}'\ \vec{L}'\ \vec{U}'] = [\vec{H}\ \vec{L}\ \vec{U}]\ R_H\ R_L\ R_U \tag{6.5}$$

where $R_H$, $R_L$ and $R_U$ are rotation matrices for rotation about three vectors by angles $\alpha_H, \alpha_L, \alpha_U$ respectively:

$$R_H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha_H & -\sin\alpha_H \\ 0 & -\sin\alpha_H & \cos\alpha_H \end{bmatrix}$$

$$R_L = \begin{bmatrix} \cos\alpha_L & 0 & -\sin\alpha_L \\ 0 & 1 & 0 \\ \sin\alpha_L & 0 & \cos\alpha_L \end{bmatrix}$$

$$R_U = \begin{bmatrix} \cos\alpha_U & \sin\alpha_U & 0 \\ -\sin\alpha_U & \cos\alpha_U & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

#### Cell death (apoptosis)

When a cellular effector responsible for cell death (Table 6.2) crosses a threshold, the cell is removed from the system. Apoptosis was allowed only in some of the simulation experiments.

#### Cellular growth

The number of cells necessary to fill a desired shape in 3-dimensional development is proportional to the volume of the shape and increases with the cube of its size. The ability to enlarge some cells can thus be an efficient way to grow larger structures using smaller number of cells.

If the cellular effector responsible for cellular growth is enabled in the system, cells are able to influence the size of a cell created after division. Its radius is calculated according to:

$$r_m = r_d \cdot (1 + kE_s) \tag{6.6}$$

where $r_d$ is the default cell radius (see Table C.3 in the Appendix), $E_s \epsilon [0, 1]$ is the concentration of cellular growth effector and $k$ is the maximum cell expansion factor (values of $k = 1$ and $k = 2$ were used in this thesis).

## 6.2 Evolution of a desired 3D morphology

This section presents the results and the approach used to evolve genomes that can control multicellular development, so that it leads to the creation of desired 3D morphologies.

### 6.2.1 Fitness function

The simple and intuitive way to compare how closely a given multicellular morphology matches a target shape is to count how many cells fit inside the desired shape, penalising for each cell outside the shape (as used, e.g., by Kumar and Bentley, 2003). The desired shape can be defined mathematically, for example, as a union of various primitives. This approach works well when cells are placed on a grid, but can have undesired consequences if cells are allowed to overlap, even temporarily, and when a fixed number of developmental steps is used (as is the case of the system described in this thesis). If cells divide at the very last time step of the development, the morphology will change only minimally (since the simulation would be stopped before the physics pushes the cells away). This allows to potentially double the number of cells inside a target shape, without changing the morphology. This is something likely to be exploited by evolution unless additional care is taken, for

example, stipulating that a certain number of physics simulation steps always occurs after the last cellular divisions or by making it impossible to improve fitness by overlapping cells.

The approach used in this thesis is based on the division of a cuboid in 3D space that embeds the target shape into cubical voxels. Each voxel is marked either as belonging to the shape or external to it. The morphology of an obtained embryo is voxelized in the same manner, and the number of matching voxels is interpreted as similarity. By assigning higher weight to some of the voxels in the target shape, it is possible, in principle, to improve evolvability of some of the morphologies that otherwise do not evolve easily (e.g., connecting parts of some structure that account for only a minor fraction of the volume). For the experiments presented in this thesis, the size of the voxels was chosen to be $\frac{1}{3}$ of the diameter of the smallest possible cell. Formally, the fitness $f_m \in [0, 1]$ for an obtained morphology is maximized by the GA and defined as:

$$f_m(D, M) = \max(0, \frac{1}{s_x s_y s_z} \sum_{x=0}^{s_x-1} \sum_{y=0}^{s_y-1} \sum_{z=0}^{s_z-1} r_{xyz}) \tag{6.7}$$

where $s_x$,$s_y$,$s_z$ are the dimensions of the cuboid in which the target shape is embedded, $D$ is the target shape ($D_{xyz} = 1$ for voxels that belong to the shape, 0 otherwise), $M$ is the obtained embryo shape ($M_{xyz} = 1$ for occupied voxels), and finally, $r_{xyz}$ is the reward for a given voxel:

$$r_{xyz}(D, M) = \begin{cases} 1 & \text{if } D_{xyz} = M_{xyz} = 1 \text{ (match)} \\ 0 & \text{if } D_{xyz} = 1 \text{ and } M_{xyz} = 0 \text{ (undergrowth)} \\ -1 & \text{if } D_{xyz} = 0 \text{ and } M_{xyz} = 1 \text{ (overgrowth)} \end{cases} \tag{6.8}$$

### 6.2.2 Embryo viability criteria

In many situations it is helpful to introduce certain minimal requirements on the embryos (see also section 3.3.4, p. 72). In the evolutionary runs described in this chapter, embryos were required to finish their development with at least two cells. Embryos in which cells never divide are not good candidates for further evolution and may not even have the necessary effector gene. A penalty fitness 0 is assigned to such individuals, and they are not used during the creation of a new generation. Additional criteria are also introduced later to obtain embryos that self terminate their development (section 6.2.8, p. 131).

### 6.2.3 Settings for the genetic algorithm and development

The limit of cells in the embryo was set to 150 and divisions would no longer occur after the limit was reached (but the physics and the dynamics of the regulatory network would still be simulated). Development was simulated for 500 time steps during which both the physics and the state of the regulatory network were updated.

**Table 6.3:** Essential GA parameters used in the experiments on evolving GRNs to control 3D development. Additional parameters are provided in the Appendix (Table C.4, p. 185).

| Parameter | Value |
|---|---|
| Population size | 100 |
| Elite individuals | 5 |
| Asexually created individuals | 85 |
| Individuals through crossover | 10 |
| Initial population | randomized genomes, 5 regulatory units each |
| Termination condition | no improvement for 500 generations |
| Selection | binary tournament, with increasing selection pressure ($k = 2$, $p$ increasing linearly from 0.6 to 1 during first 2000 generations) |
| Cell limit during development | 150 |
| Developmental time steps | 500 |

**Table 6.4:** Types of products and promoters enabled in the experiments on evolving GRNs to control 3D development and the interpretation of subsequent input and output elements.

| Promoter types | Product types | External factors | Effectors |
|---|---|---|---|
| additive | transcription factor<br>morphogen | "1" (fixed high concentration)<br>maternal morphogen at (8,0,0) | divide (threshold 0.8)<br>rotation $R_H$<br>rotation $R_L$<br>rotation $R_U$ |

GA settings were similar to those used in the previous chapters (Table 6.3), although a smaller population was used due to a higher computational cost of the experiments. Binary tournament selection was used in which the probability of selection was increasing over time to protect initial genetic diversity. A single maternal gradient was enabled, with a maximum of concentration at $(8, 0, 0)$ (this position in space corresponds to the tip of the right pointing axis arrow in Fig. 6.4). Apoptosis and modification of cell size were disabled (see Table 6.4 for a summary of types of genetic elements enabled in this experiment).

### 6.2.4 Evolution of an ellipsoidal morphology

In an initial experiment, an attempt was made to evolve a symmetric, ellipsoidal morphology (Fig. 6.4a). The development would start from the first cell located at the centre of the coordinate system shown. Ten evolutionary runs with the same parameters were executed. Largely owing to the simplicity of the target morphology, all of the obtained embryos visually resembled the target shape. The least fit embryos were more spherical. The best individual obtained in 10 runs (Fig. 6.4b) reached the fitness of 0.6, with a clearly ellipsoidal morphology. The evolutionary runs lasted around $1000 - 3000$ generations (Fig. 6.5) and started with spherical embryos that were result of random genome initialization (with fitnesses around 0.35). In all runs, the quick improvement occurring during the first 200-300 generations is followed by a slower improvement in later generations.

The developmental process of the best obtained individual was investigated (Fig. 6.6). Initially, cells divide in a straight line, while the effectors responsible for rotation during division are not yet expressed. During that time, the cells remain com-

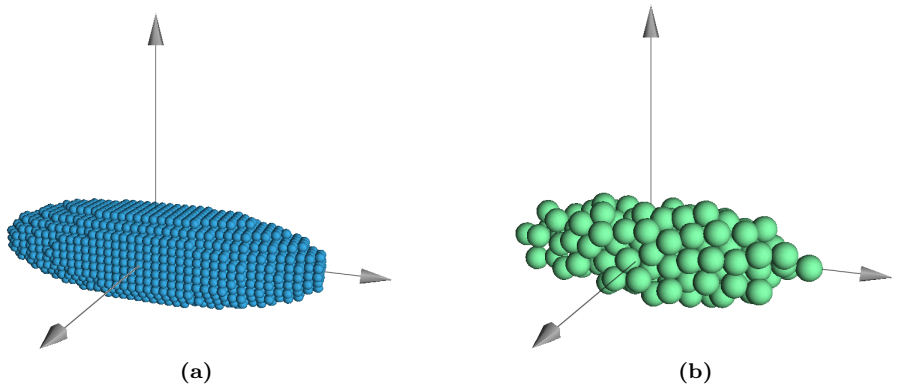



**Figure 6.4:** Target shape and the best obtained morphology of an ellipsoidal embryo: (a) voxelized target morphology, each voxel is drawn as a small, blue sphere, (b) the best matching morphology obtained in 10 independent evolutionary runs, each green sphere represents a cell. Development starts with initial cell located at the centre of the coordinate system.

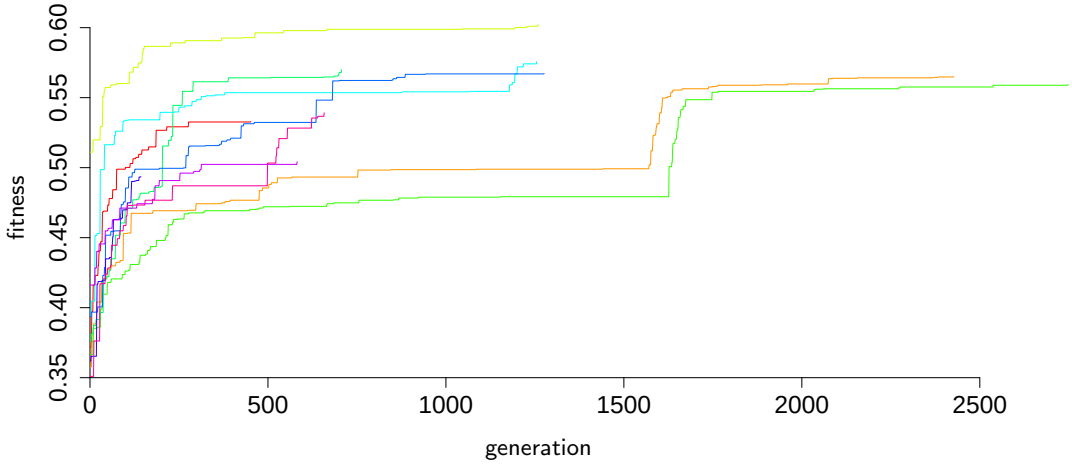



**Figure 6.5:** History of fitness of the best individual in a population for 10 independent runs evolving ellipsoidal morphology. Each GA run lasted until no improvement for 500 generations was observed.

pressed, because cell divisions occur faster than they are pushed away from each other. Then, on the left side of the embryo, new cells start to divide to the side of the line (Fig. 6.6d). Next, a wave of cellular divisions propagates towards the right side of the embryo. During the remaining 150 time steps of development, physical forces equilibrate the structure of the embryo and cells reduce their overlap.

### 6.2.5 Evolution of an asymmetric morphology: a stem-cap shape

For a second series of evolutionary runs a more complex morphology was specified with left-right asymmetry, consisting of an elongated stem ending with a cap (Fig. 6.7a). As the initial cell is located in the middle of the structure, the embryos had to grow in both directions, asymmetrically. Similarly to simulation for the ellipsoidal target in the previous section, GA was running between 1000 and 4000 generations and fitnesses rapidly improved in a first few hundreds of generations (Fig. 6.8).

Most of the 10 runs ended with individuals evolving only the cap of the structure,

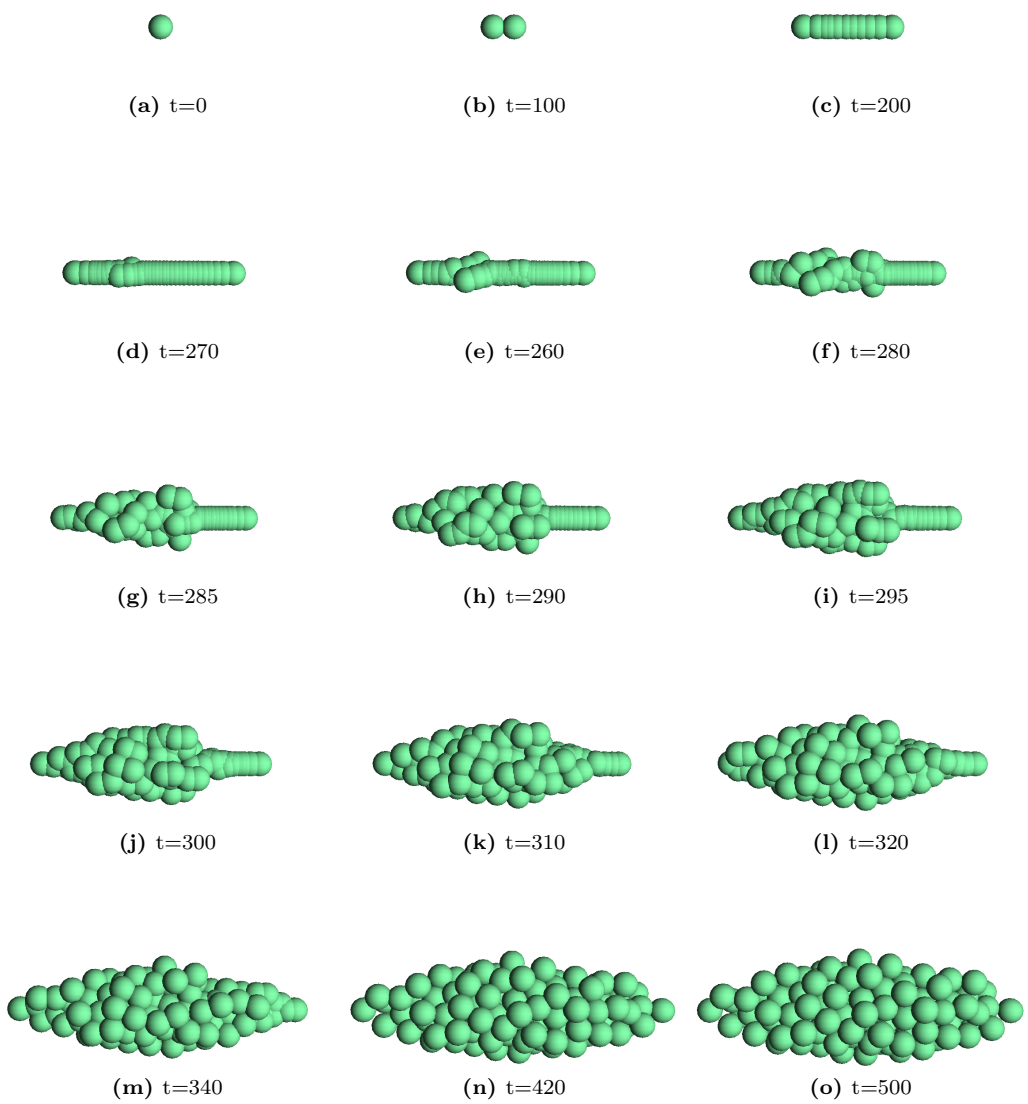
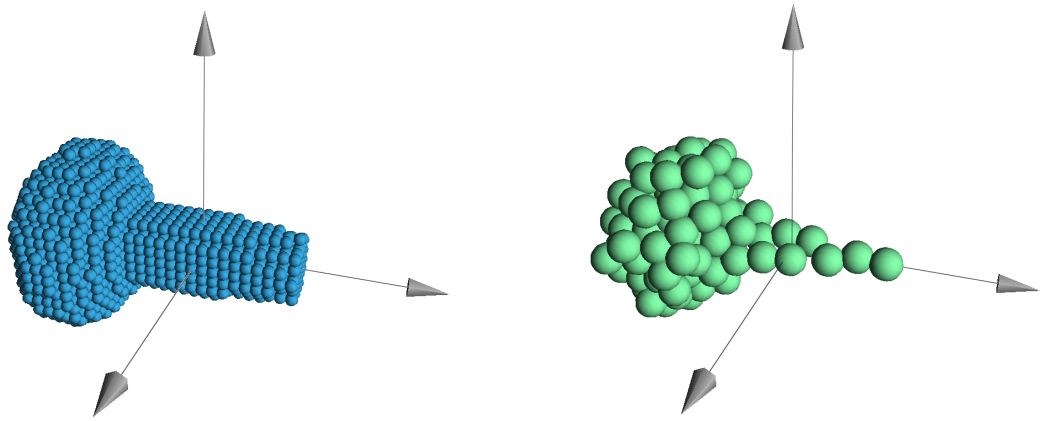



**Figure 6.6:** Evolved developmental process of an ellipsoidal morphology. Frames show consecutive stages in the development of the best individual from 10 independent evolutionary runs (same embryo as in Fig. 6.4b)

with stem either non existent or poorly pronounced. However, the best individual consists of both stem and a cap and reached the fitness of 0.58. The final stage of its development is shown next to the target shape in Fig. 6.7b.

Similarly to the development of the ellipsoidal embryo (section 6.2.4), development of this best stem-cap individual initially proceeds with cells dividing in a line (Fig. 6.9) until, at the stage of 14 cells, the cells at one extreme of the embryo start to reorient during division to form a cap. Some of the cells that formed the stem at the very beginning divide further to the side.

Closer inspection of this individual reveals that the genome has grown from 5 regulatory units found in initial, random individuals, to 13 regulatory units. At that time, the number of genetic elements increased from an average of 18 in the initial population to 57 in the best individual. All new regulatory units formed



(a) (b)

**Figure 6.7:** Target shape and the best obtained morphology of a stem-cap shape: (a) voxelized target morphology, each voxel is drawn as a small blue sphere, (b) the best matching embryo obtained in 10 independent evolutionary runs, each green sphere represents a cell. Development starts with initial cell located at the centre of the coordinate system.

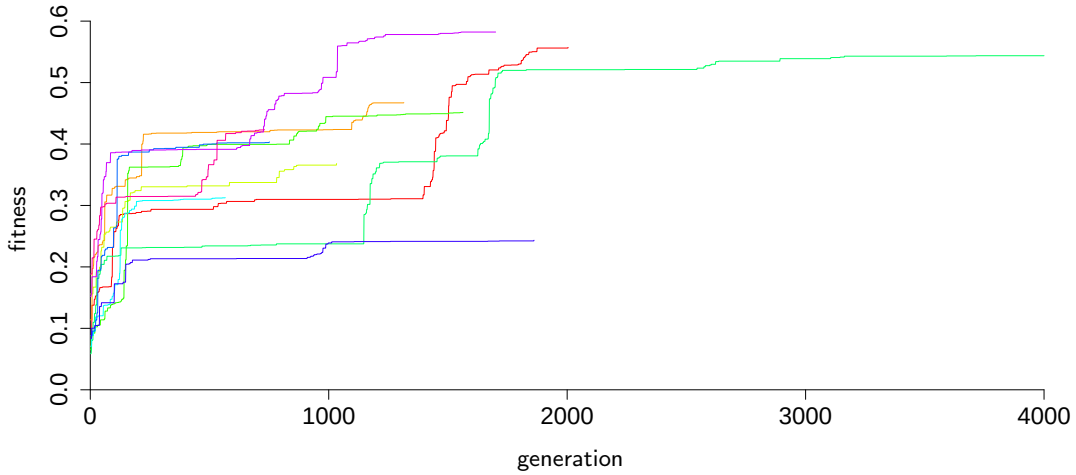
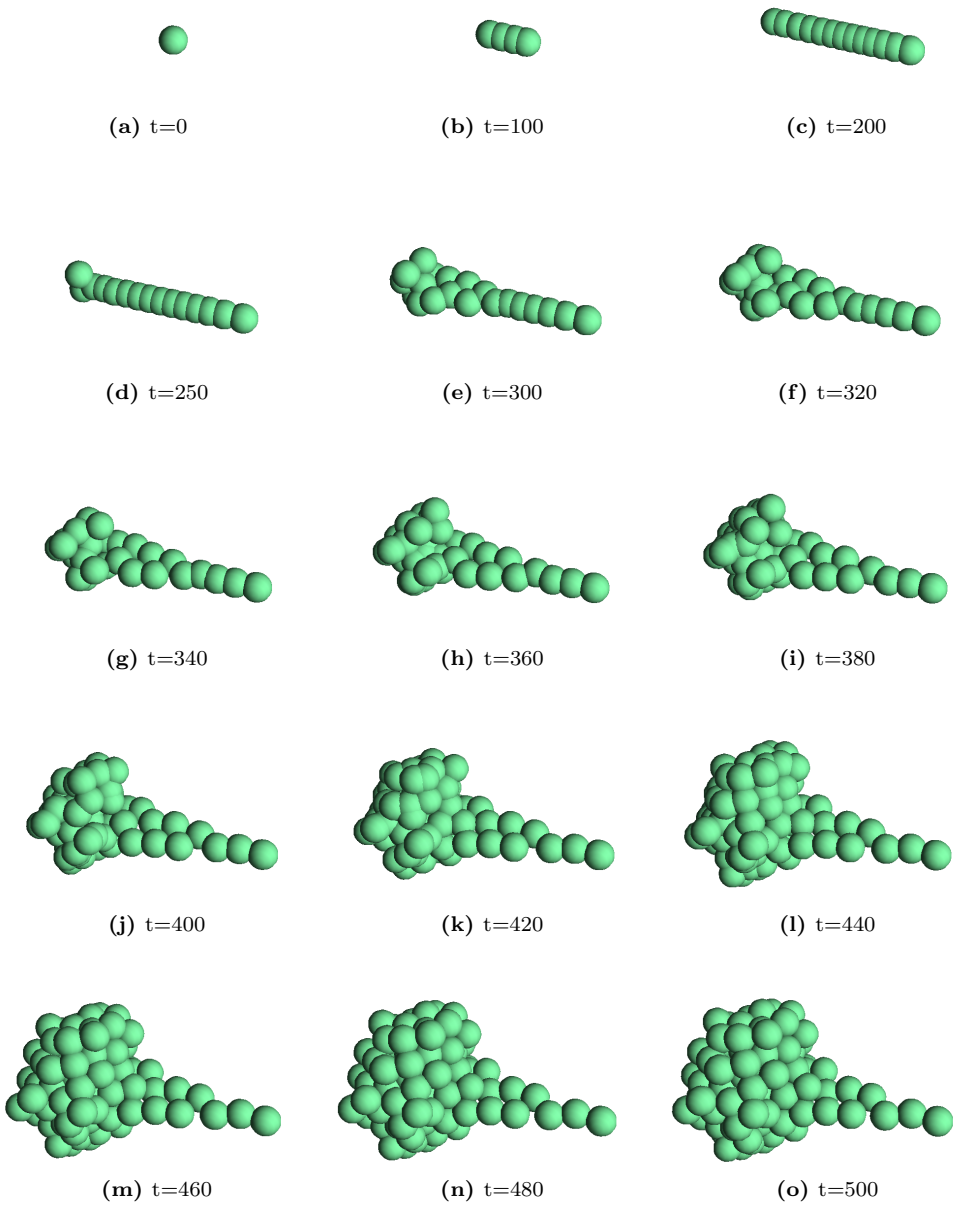



**Figure 6.8:** History of fitness of the best individual in a population for 10 independent runs evolving a stem-cap shape. Each GA run lasted until no improvement for 500 generations was observed.

through duplications of existing genetic elements, either through the operator of gene duplications or via sexual recombination with another individual.

### 6.2.6 Knock-out experiments on the evolved stem-cap shape

In principle, the maternal concentration gradient predeposited in the environment and emitted from a single point in space provides enough spatial information for cells to form the presented stem-cap shape. Cell could however also produce their own morphogens, if the genome encoded them. Apart from having some small probability of being present in initial random genomes (Table C.4 in the Appendix), genetic elements encoding morphogens could appear through a mutation of a type. This means that element that encoded a TF could mutate into a morphogen and start to diffuse outside of a cell, while retaining its original affinities to the same regulatory regions.



**(a)** t=0 **(b)** t=100 **(c)** t=200

**(d)** t=250 **(e)** t=300 **(f)** t=320

**(g)** t=340 **(h)** t=360 **(i)** t=380

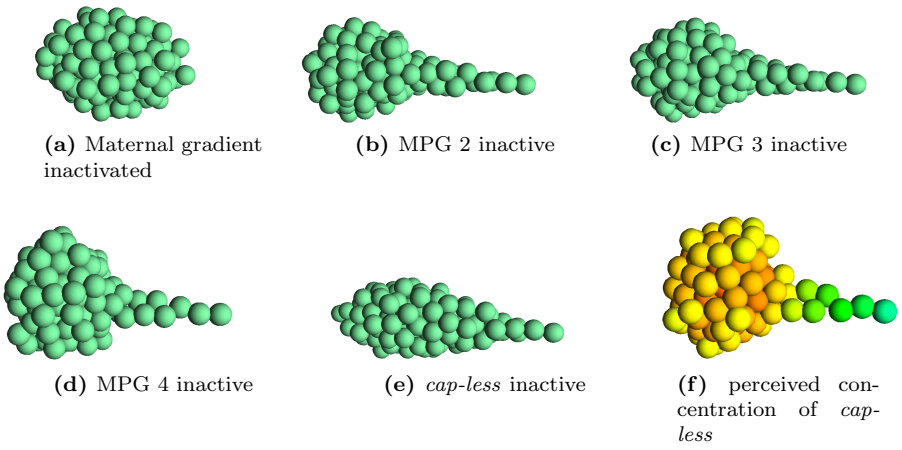
**(j)** t=400 **(k)** t=420 **(l)** t=440

**(m)** t=460 **(n)** t=480 **(o)** t=500

**Figure 6.9:** Evolved development of the stem-cap shape. Frames show consecutive stages in the development of the best individual from 10 independent evolutionary runs (the same embryo as seen in Fig. 6.7)

The genome of the investigated individual was found to encode 6 morphogens that were produced during development. To find out how both maternal gradient and these self-produced morphogens influence the development of an obtained embryo, a series of knock out experiments was performed. This type of experiments corresponds to the approach used in molecular genetics, where a given gene is inactivated to elucidate the role of its product in a particular process. In the case of the model presented in this thesis, one can easily render certain genetic element inactive, by setting its associated coordinates to a very large value (so that its product can no longer bind to anything) and then simulating the developmental process again.

When the maternal morphogen gradient was disabled, the development would



**(a)** Maternal gradient inactivated **(b)** MPG 2 inactive **(c)** MPG 3 inactive

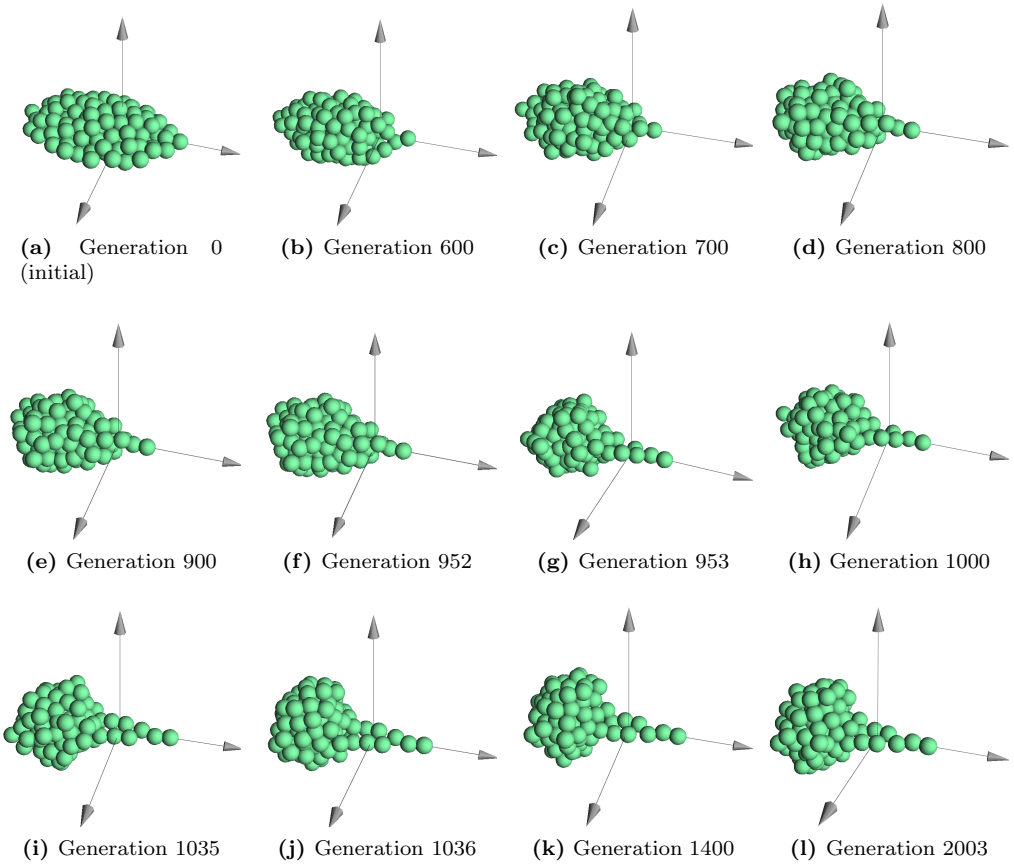
**(d)** MPG 4 inactive **(e)** *cap-less* inactive **(f)** perceived concentration of *cap-less*

**Figure 6.10:** Effects of gene knock-outs on the development of the evolved stem-cap shape. (a) shows the effect of removing the maternal morphogen gradient, (bcdef) show the effects of inactivation of different morphogens produced by this embryo (g) demonstrates the concentration of *cap-less* in the cells of normally developed embryo (blue - zero, green - low, yellow and red - high).

result in forming a mostly spherical clump of cells (Fig. 6.10a). This suggests that it was essential to the breaking of embryo's symmetry. Since this maternal gradient was most likely the only source of asymmetry in the environment at the beginning of evolution (before self produced morphogen gradients could emerge), it is perhaps not surprising that it evolved to be central for the development.

The effects of inactivating self produced morphogens were investigated next. One morphogen (MPG 1), despite being expressed, was found to be completely neutral (there was no phenotypic effect of a knock out). The loss of another two morphogens (MPG 2 and 3) results in a much less pronounced "cap" in the structure (Fig. 6.10bc). Inactivation of another morphogen (MPG 4) results in a slightly degraded shape, but still consisting of a clearly defined stem and a cap (Fig. 6.10d). Thus, although the morphogens are not essential for the development, they evolved to assist in improving the shape of the embryo. However, when the 5th morphogen was inactivated, a cap structure would fail to develop almost entirely (Fig. 6.10e). Following the convention for naming biological genes, the gene coding for this morphogen was named *cap-less*. Interestingly, when its perceived concentration in the cells was overlaid on the structure of the normally developed embryo, *cap-less* was found to be mostly influencing the cells in the cap of the structure (Fig. 6.10f). A similar relation between a morphological structure and a gene encoding a morphogen was also observed in an independent set of experiments based on the version of the presented model. There, a dose effect was also observed: a higher concentration would lead to a more pronounced structure (Joachimczak and Wróbel, 2008a).

However, it should be noted that the fact that *cap-less* can be demonstrated to be necessary for development of the morphological structure, does not imply a causal relationship between presence of the *cap-less* product in the cells and the formation of the cap. Instead, *cap-less* should be seen as one of many genes that are involved in the formation of the cap which, when disabled, will result in the partial or full degradation of the structure. The situation in biological development



**(a)** Generation 0 (initial) **(b)** Generation 600 **(c)** Generation 700 **(d)** Generation 800

**(e)** Generation 900 **(f)** Generation 952 **(g)** Generation 953 **(h)** Generation 1000

**(i)** Generation 1035 **(j)** Generation 1036 **(k)** Generation 1400 **(l)** Generation 2003

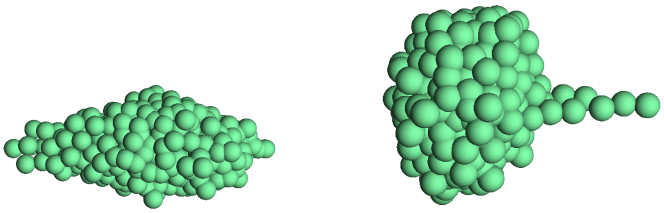
**Figure 6.11:** The best matching shape in a population over generations during evolution of the stem-cap shape.

is analogous and, unfortunately, often misrepresented by popular media eager to associate complex traits with single genes (e.g., announcing the discovery of the "obesity gene" or the "intelligence gene").

### 6.2.7 Change of the morphology over evolutionary time for the stem-cap shape

To see how the morphology changes during evolution, best individuals from populations at various periods of evolutionary history were sampled. The most fit individuals that existed at a given generation are not necessarily direct ancestors of the best obtained morphology, but they are closely related to the direct ancestors, and so, can be expected to be representative for their morphology. By the same reasoning, fossils provide us with an insight into how ancestors of existing animals could have looked like, despite the fact that we cannot hope to find fossils of the actual ancestors.

The analysis shows (Fig. 6.11) that evolution started from a highly flattened, ellipsoidal shape which, over subsequent generations, progressively shifted its centre of weight towards the final location of the cap. At the same time, the structure would continue to elongate its stem.



(a) (b)

**Figure 6.12:** Development of the ellipsoidal (a) and the asymmetric morphology (b) with a cell limit of 150 disabled. Termination of growth did not evolve and cells would continue divisions as long as the development was simulated.

### 6.2.8 Evolving self-termination of division

For the experiments with both target shapes, cellular divisions were set to stop when the embryo reached the size of 150 cells. In all runs, evolved embryos were found to rely on this hard limit to terminate their development. That is, if the limit was removed, divisions would continue to occur, most likely leading to an unlimited growth (Fig. 6.12).

To test if it is possible to obtain embryos self terminating their development, viability criteria were extended to penalize embryos with the maximum possible number of cells. This limit was increased to 400 cells, and whenever individual grew up to 400 cells, it was considered non viable, i.e., it was not used to generate new generation (see also section 3.3.4, p. 72). As expected, such runs resulted in obtaining embryos that did not reach 400 cells during their development and thus did not rely on the hard limit of embryo size.

However, closer inspection of obtained embryos revealed that if the development is continued beyond the original length of the simulation (i.e., 500 time steps), divisions continue to occur, soon reaching the hard limit. What happens is that instead of evolving a genetic mechanism that would stabilize and reduce production of TFs activating the division effector, evolution tends to find solutions that have a slow, but steady division rate, so that the hard limit is never reached during the fixed time of the simulation of development. This is one of many examples of how evolution is likely to find the simplest and the "cheapest" solution within provided constraints.

In order to obtain individuals that are truly capable of self terminating their development, viability criteria had to include an additional condition. The developmental process was extended from 500 to 600 time steps, but if divisions or apoptosis occurred during the last 100 time steps of development, the individual would be considered non viable. This ensured that all viable embryos were capable of stopping their division for at least 100 time steps and not simply because of the exploitation of the hard limit.

The experimental setting with the ellipsoidal target (section 6.2.4) was reused but with the two additional viability criteria described above. All 10 runs resulted in individuals that terminate their development before 500th time step. To test if solutions are indeed stable, the development was simulated for another 1200 steps (3

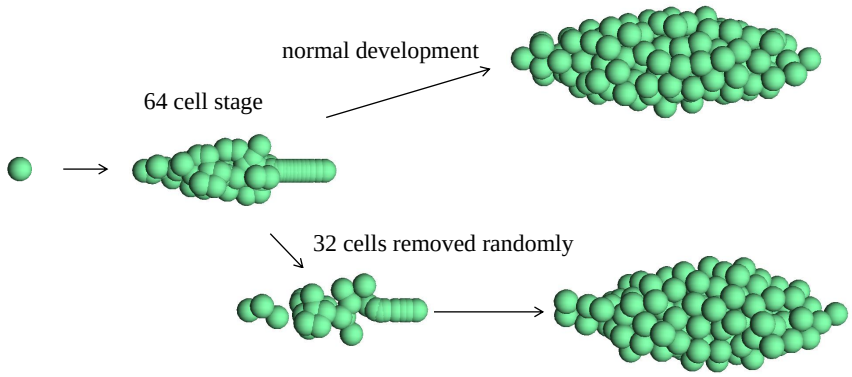



**Figure 6.13:** Illustration of the approach used to evaluate robustness to cellular damage.

times longer than their original lifetime). Cell divisions were observed after the final developmental step 600 only in 1 of the 10 best individuals obtained in independent runs.

This result suggests that the two proposed viability criteria are an effective way to evolve self termination of the development. Importantly, both criteria have to be enabled, as none of them is sufficient by itself.

## 6.3 Robustness to cellular damage

Robustness to perturbations such as mutations or damage is one of the principal reasons of interest in developmental systems (Stanley and Miikkulainen, 2003) and, with the aim of better understanding which properties of such systems contribute to their resilience, has been investigated in multiple existing developmental systems (see, e.g., Andersen et al., 2009; Eggenberger Hotz, 2003a; Miller, 2004; Streichert et al., 2003, for some recent examples). Biological embryos are known to develop properly in a wide range of variations of external environment and can tolerate various levels of cellular damage. For example, removing a single cell during initial stages of development typically has negligible effects on further development. Multicellular organisms are also capable of self-repair even after the development had been completed, with some higher animals (e.g., newts) capable of regrowing a lost limb. Hence the hope that by recreating some of the elements of biological development *in silico*, we can design systems that are robust to damage and capable of self-repair.

### 6.3.1 Robustness during development

To investigate how robust are the embryos evolved using the introduced embryogenesis model, cellular damage was applied at various stages of the development. After the number of cells in the embryo reaches a threshold (e.g., 64), a percentage of existing cells is removed randomly and the development is continued (see Fig. 6.13 for visual explanation). The damaged embryo is then compared to that obtained during unperturbed development, and its quality is represented as a fraction of its fitness

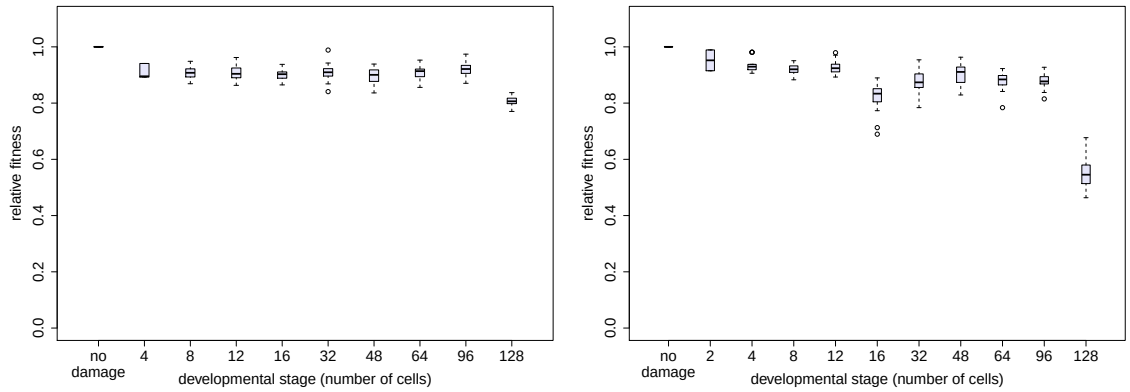



**Figure 6.14:** Robustness of the evolved ellipsoidal embryo to cellular damage at various developmental stages. (a) effect of removal of 25% cells, (b) effect of removal of 50% cells. Fitness is relative to the fitness obtained in unperturbed development. The box plots show the median and quartiles for the relative fitnesses obtained in 40 independent developments with random cell removal. Whiskers extend to the most extreme data point which is no more than 1.5IQR from the box. Circles indicate outliers.

.

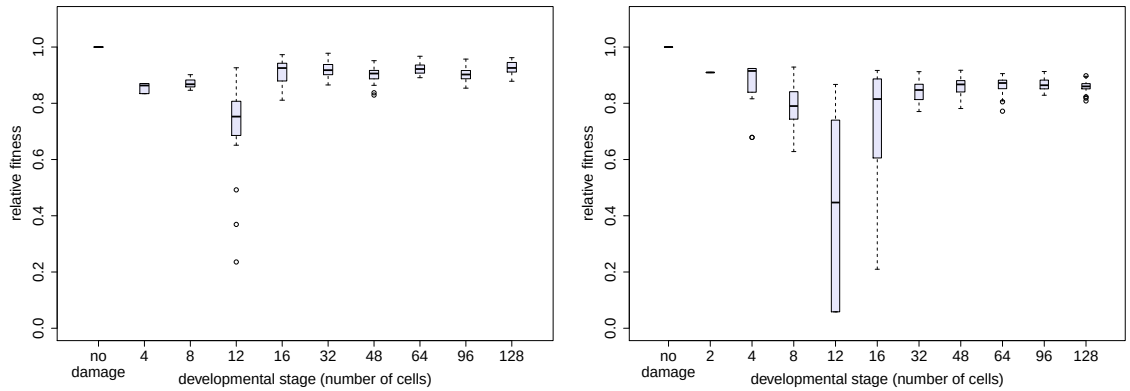



**Figure 6.15:** Robustness of the evolved stem-cap embryo to cellular damage at various developmental stages. (a) effect of removal of 25% cells, (b) effect of removal of 50% cells. The box plots show the median and quartiles for the relative fitnesses obtained in 40 independent developments with random cell removal. Whiskers extend to the most extreme data point which is no more than 1.5IQR from the box. Circles indicate outliers.

.

over the original fitness. As cells were selected randomly during the removal, the process was repeated 40 times for each developmental stage to measure the average effect.

The ellipsoidal and asymmetric embryos described in sections 6.2.4 and 6.2.5 were evaluated for the effects of random cell removal at the stages of 2, 4, 8, 16, 64, 96 and 128 cells (the development would continue up to the limit of 150 cells). Embryos were tested for their robustness to removal of either 25% or 50% cells at every stage.

Considering that removal of 25% cells results in a considerable damage to the embryo, the development of the ellipsoidal embryo is clearly very robust (Fig. 6.14a),

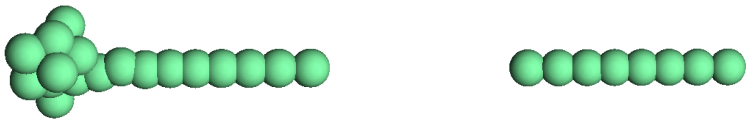
being able to usually regain close to 90% of the original fitness for most of the developmental stages. Only after reaching the size of 128 cells, further loss of ability to regrow is seen.

The robustness to removal of 50% of the cells at each stage is only slightly lower (Fig. 6.14b). The ability to restore the structure is very high, with the removal of one of the two initial cells having the smallest effect on development. Interestingly, the embryos consisting of 16 cells seem to be less resilient to cell removal than further developmental stages, indicating that this is a more sensitive stage. One likely interpretation is that cells at this stage have to break from the development that initially occurs in a line. Damaging the cells that are essential to this reorientation has higher impact on fitness. Still, the embryos are capable of regaining more than 80% of their original fitness (Fig. 6.14b). Nonetheless, such high robustness to damage can be attributed partly to the simplicity of the target shape: obtaining an ellipsoidal shape does not require precise control during development and mostly relies on physics that pushes cells away from each other.

When robustness of development of an asymmetric embryo was analysed in the same manner it was found to be less resilient to damage than the ellipsoidal embryo (Fig. 6.15). However, it is still surprisingly robust. It is able to restore around 85% of the original fitness when 25% cells are removed. The development clearly appears to go through a bottleneck during which the impact of cell removal is much higher than during initial and later stages of development. This is most pronounced at the stage of 12 cells and can be observed both for removal of 25% and 50% cells.

Further investigation of the issue shows that at this stage of development cells were still dividing in a line (Fig. 6.9a-c, p. 128). Divisions occur only in the leftmost cells, towards the side where the cap will be formed later. By manually and selectively removing cells in the embryos of size 12, it was observed that removal of rightmost cells does not influence fitness much. The remaining cells do not divide any more and remain to be part of the stem of the final structure, but a shortened stem does not influence fitness considerably as it amounts to only a small fraction of the volume. However, the leftmost cells drive the divisions that result in the bulge. Removal of a single such cell was not found to have a large impact on fitness, but removal of 3 or 4 cells is enough to delay the development of the bulge so that it never forms during the allowed simulation window or forms only partially. This explains the observed developmental period of increased sensitivity to cellular damage. At the early stages of development, the divisions are driven by a small fraction of leftmost cells, similarly to a meristem in plant growth (see, e.g., Evert and Eichhorn, 2004). Removal of a single cell does not influence the development, but if due to random removal too many leftmost cells are lost, the embryo will no longer form the cap of the structure (Fig. 6.16).

All of the individuals described in this chapter evolved and developed in the absence of any developmental stochasticity, but they are still very robust to cellular damage. If the randomness was inherent to the development, some level of robustness would be expected to emerge, just as randomness in gene expression leads to the evolution of networks more robust to noise (section 4.6, p. 94). This means that



(a) (b)

**Figure 6.16:** The effect on the development of the stem-cap embryo of removing cells when it reaches the size of 12 cells (at this point in development, cells are still aligned in a line). The final stage of development is shown after 3 leftmost cells were removed (a) or 4 leftmost cells (b).

the robustness shown above does not stem from the fact that embryos had to be robust to environmental variation and can be considered to be an emergent feature of the evolving, developmental system, independent of environmental stochasticity. One suggested explanation for the emergence of fault tolerance is believed to be mutational robustness. Under the mutational pressure, genomes have to develop certain level of robustness in response to genetic mutations and as a side effect, this gives them certain level of robustness to environmental perturbations as well (Federici and Ziemke, 2006). Preliminary results for embryos evolved with noise in gene expression (using the method discussed in section 4.6, p. 94) suggest that such embryos display an even increased robustness to cellular damage as a side effect (Joachimczak and Wróbel, 2012a).

### 6.3.2 Embryo regrowth

Although robustness to cellular damage during development was observed to be an emergent feature of the model described in this thesis, fully developed embryos were not observed to display the ability to regrow removed parts of their structure. One reason for this is the lack of genetic control of growth termination in the embryos discussed in the section 6.2.8. In these embryos, typically a large fraction of cells would have their division effectors above the threshold at the end of development, but these cells would not divide because of the hard limit set on the number of allowed cells in the embryo. When a group of cells is locally removed from an embryo (rather than randomly, as in section 6.3.1), cells will immediately start dividing again, but there is no reason why this should happen locally at the site where cells were just removed: any cell in the embryo that was ready to divide can now do so.

The situation is different for embryos that evolved self termination of development (section 6.2.8). These embryos either do not respond at all to damage or initiate regrowth in the whole structure, sometimes even falling into uncontrolled growth. Such uncontrolled growth most likely occurs in embryos in which self-termination depends on the concentrations of self produced morphogens. As soon as a large enough fraction of cells is removed, the inhibiting effect of those morphogens vanishes, and divisions continue until enough cells produce this morphogen and stop the process again. However, even for the simplest embryos, divisions are not likely to result in restoration of ellipsoidal morphology.

To allow for local regrowth to occur, a given cell has to be able to detect a need to do so. This information, to some extent, is provided by the change in concen-

trations of morphogens in its surrounding, but due to the lack of observed regrowth ability, in another series of experiments, an additional, more explicit external signal was provided to each cell. This signal carries the information about the number of neighbours in cell's surrounding. A cell that has direct contact with 8 other cells would perceive a maximum concentration of this external signal (i.e., 1). More precisely, the external factor would have a concentration of $\frac{k}{8}$, where $k$ is the number of adjacent cells. This way, cells were able to obtain accurate information about their direct surroundings and could detect the number of lost neighbours after removal. Unfortunately, adding this external factor did not result in embryos capable of regrowing lost fragments.

It is possible that local regrowth does not evolve if embryos are not rewarded for it. To test if this is a case, the ability of regrowth was made an explicit part of the fitness function. The target shape was set to an ellipsoidal morphology, but each embryo was now evaluated 3 times. In the first evaluation, the development is unperturbed and is simulated for 800 time steps. The morphology of the embryo is then compared to that of the target shape twice. First at the time step 400 and for a second time at the time step 800. During the second evaluation, slightly more that the right half of the embryo is removed at the time step 410. Development is allowed to continue unperturbed until the time step 800. The shape of the embryo at the time steps 400 and 800 is compared to the target. The third evaluation is analogous to the second, but the leftmost part is removed. Finally, the fitness value of an individual is calculated as a weighted average of the shape similarity in the 6 evaluations, with the 2 evaluations that follow cell removal having twice the weight. The reason for this higher weight was to put higher pressure on regrowth, as of the 6 comparisons to the target shape, only two would actually depend on the ability to regrow lost fragment. Another approach could be to simply compare the morphology of the embryo 3 times, each at the final developmental step 800. However, this was found to be easily exploited by evolution. Embryos evolved using this approach were found to develop very slowly, so that when half of the embryo is removed at simulation step 410, the embryo is still very small and only a few cells are removed. Only after simulation step 410, embryos continued to develop to their full size, sidestepping the effect of cell loss.

The new setup proved to be more successful, but only 1 in 10 runs resulted in an individual able to partially regrow after either its left or right part was removed. The remaining individuals can regrow only one of their sides, but not both. The best individual self terminates the development around time step 300 and remains stable up to step 800 (Fig. 6.17a). When the left part of this embryo is "sliced off" (Fig. 6.17a-f), the regrowth starts to occur very quickly, in around 20 time steps. Only the cells that were located next to the lesion divide. The final stage of the embryo after developing with the damage to the left part does not fully resemble the original, but a large part of the structure is regrown. The development of this individual is less robust to damage to the right side (Fig. 6.17g-l). The regrowth starts a bit later, about 90 time steps after the lesion. Similarly to the situation after damage to the left, the cells divide only locally, as is desired, but the end result

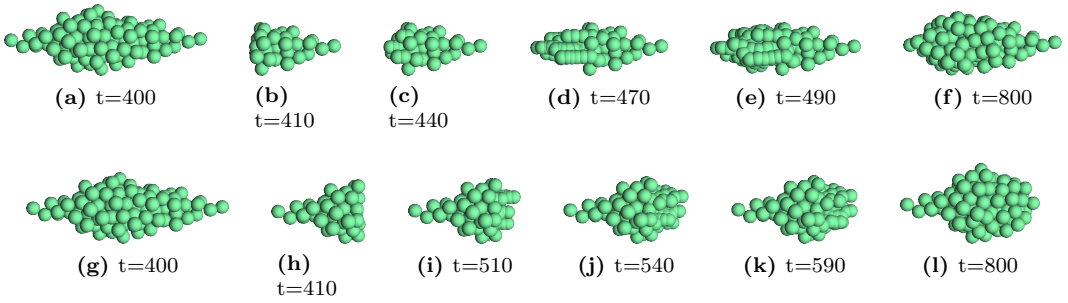



**Figure 6.17:** An ellipsoidal morphology evolved with a fitness function promoting ability of regrowth, responding to a removal of left part of the structure (a-f) and right part of the structure (g-l).

is more spherical.

The initiation of regrowth in the best individual obtained in 10 runs has to depend entirely on the ability of cells to detect when, and where the damage occurred. Since under this experimental setting cells are not provided with explicit information about their neighbours (as discussed earlier), the only way they can detect the change in the structure of the embryo is through the change in concentrations of morphogens. When cells are removed, concentrations change and the GRNs in adjacent cells have to react to this change.

The quality of self repair obtained in the presented experiments is far from perfect, but demonstrates that it is indeed possible to evolve capability to regrow fragments of the embryo, if the embryos are rewarded explicitly in the fitness function. The growth occurs locally, at the site where the cell lesion is applied. The fitness function had to be constructed carefully, to avoid evolution of simplified and undesired solutions. It is possible that further tweaking of the parameters would result in improved robustness to this type of damage. However, the experience so far suggests that the main obstacle for improved evolvability may be the need to detect when and where the lesion occurs, based solely on changes of morphogen concentrations. Perhaps the results could be considerably improved if this information was provided in a more explicit manner. This could be done, for example, by providing cells with additional external factor that would signal that a damage has occurred to cells adjacent to the lesion. Such signals indeed exist in living organisms and are sent (chemically) by cells in the tissues that are located at the site of tissue damage, signalling the need to activate regrowth.

## 6.4 Evolution of 3D patterning

In the evolutionary runs described in this section the capability of the introduced developmental model to evolve both 3D morphology and cellular differentiation is investigated. The cells are allowed to differentiate by changing their colour. The colour of a cell results from the increased concentrations of new type of pigment-like effectors. The fitness function is designed to reward reaching a desired morphology as well as a desired colour pattern.

### 6.4.1 GA settings and genome configuration

The model was extended by three new types of effectors responsible for colouring a cell: *red, blue* and *green*. When the concentration of these effectors becomes non zero, a cell becomes coloured. The exact interpretation of their concentrations as a colour depends on the experiments and a few different approaches are investigated. In the first experiment (section 6.4.3) only the *red* and the *blue* effector is enabled, and their effect is thresholded, i.e., the cell becomes coloured only if the effector has a concentration above 0.5. This leads to 4 possible colour states for each cell. When none of the effectors crosses the threshold the cell is marked as colourless (white). If only one effector crosses the threshold, the cell is coloured accordingly. When both the *red* and the *blue* cross their threshold, the cell becomes pink.

Alternative scenarios in which effectors are not thresholded (so a cell can display intermediate levels of red and blue or their mix) or when a third effector, *green* is enabled, are explored in sections 6.4.4, 6.4.5, 6.4.6.

Of the external factors listed in the Table 6.1 (p. 117) only the signal "1" was kept. In particular, maternal morphogen gradients were not used to see whether the embryos can still differentiate their cells based only upon morphogens that are endogenous. The GA used in the evolutionary runs presented in this section reused most of the settings used for experiments in Chapter 4, e.g., it allowed for both additive and multiplicative promoter types (see Table 6.5 and 6.6 for a summary of the settings).

**Table 6.5:** Essential GA parameters used in the experiments on evolving GRNs to control patterning of 3D embryos. Additional parameters are provided in the Appendix (Table C.5, p. 186).

| Parameter | Value |
|---|---|
| Population size | 300 |
| Elite individuals | 5 |
| Asexually created individuals | 195 |
| Individuals through crossover | 100 |
| Initial population | randomized genomes, 5 regulatory units each |
| Termination condition | no improvement for 500 generations |
| Selection | tournament ($k = 10$, $p = 0.3$) |
| Cell limit during development | 200 |
| Developmental time steps | 300 |

**Table 6.6:** Types of products and promoters enabled in the experiments on evolving GRNs to control patterning of 3D embryos and the interpretation of subsequent input and output elements.

| Promoter types | Product types | External factors | Effectors |
|---|---|---|---|
| additive<br>multiplicative | transcription factor<br>morphogen | "1" | divide (threshold 0.8)<br>die (threshold 0.8)<br>cell radius<br>red pigment<br>blue pigment<br>green pigment (depends on the experiment)<br>rotation $R_H$<br>rotation $R_L$<br>rotation $R_U$ |

### 6.4.2 Fitness function

The fitness function was derived from the function used for the evolution of 3D morphology (Eq. 6.7, p. 123). It relies on the comparison of a voxelized definition of the target shape with a voxelized phenotype of a developed individual. However, since the goal of the evolutionary simulations described in this section is to simultaneously evolve desired morphology and patterning, the fitness function is designed to reward for both the correct morphology as well as patterning.

The reward for the expression of colour effectors in a particular voxel is computed so that only expressing the right colour effector and silencing effectors for other colours results in a maximum reward. The exact form of this equation depends on the number of allowed colour effectors and the choice of colour coding. In the first experiment (section 6.4.3) with 2 colour effectors (*red* and *blue*), the following equation is used:

$$f_{col}(D,M,R,B) = max(0, \frac{1}{s_x s_y s_z} \sum_{x=0}^{s_x-1} \sum_{y=0}^{s_y-1} \sum_{z=0}^{s_z-1} \frac{(r_{xyz} + c_p c_{xyz})}{2}) \in [0,1] \qquad (6.9)$$

$$c_{xyz}(D_{xyz}, R_{xyz}, B_{xyz}) = \begin{cases} (R_{xyz} + 1 - B_{xyz})/2, & \text{if } D_{xyz} \text{ is } \textit{red} \\ (B_{xyz} + 1 - R_{xyz})/2, & \text{if } D_{xyz} \text{ is } \textit{blue} \\ 1 - (R_{xyz} + B_{xyz})/2 & \text{if } D_{xyz} \text{ is } \textit{white} \\ 0, & \text{if } D_{xyz} \text{ is } \textit{empty} \end{cases} \in [0,1] \qquad (6.10)$$

where $D$ is the desired, voxelized target pattern of size $s_x \times s_y \times s_z$, $M$ is the obtained morphology ($M_{xyz} = 1$ if a voxel is filled by some cell), whereas $R$ and $B$ are the thresholded levels of *red* and *blue* effectors in a cell that occupies a given voxel. For example, if the concentration of the *red* effector in the cell is above 0.5, $R_{xyz}$ becomes 1, otherwise it is 0, and the concentration of this effector does not influence cell's colour. $r_{xyz}$ is the reward for a correctly occupied voxel (Eq. 6.8) and $c_{xyz}$ is the reward for expressing colour effectors at a desired level. Half of the possible

total reward can be reached by correctly filling the target morphology with their cells, but an additional increase requires the expression of the right colour genes.

Since the fitness function rewards both the morphology and the correct patterning, it is possible for two coloured individuals to have a higher fitness than a three-coloured one as long as their morphology matches the target shape more closely. This, however, can be deceptive for the search algorithm and GA can become stuck in a local maximum in which individuals have a highly fit morphology, but do not have the genetic circuity allowing them to produce all colours. For this reason, the term $c_p$ was added to promote expression of colours early on during evolution. For any individual that would have at least one red, blue and white cell, no matter where, $c_p$ was equal to 1. For individuals with cells in two different colours $c_p$ was equal to 0.5. For individuals with just one colour $c_p$ was equal to 0.33. This approach was found to improve evolvability by highly rewarding existence of at least some ability to differentiate into different colours. Without it, the embryos would often evolve to have only cells with one colour. A similar approach was used in by Knabe et al. (2008b).

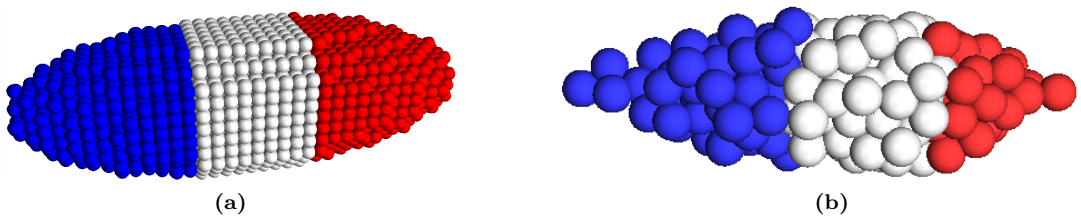
### 6.4.3 French flag problem in 3D: the tricolour embryo

One of the popular toy problems used to test the evolvability of a developmental system in the field of artificial embryogenesis is the so-called French flag problem. In this task, the embryo develops to recreate a pattern of 2D French tricolour, i.e., the cells have to differentiate into three spatially segregated types (see e.g., Chavoya and Duthen, 2008; Fontana, 2008; Knabe et al., 2008b; Miller, 2004, for recent examples). The French flag model itself derives from an explanation, proposed more than 40 years ago by Wolpert (1968), of how signalling molecule (a morphogen), emitted from a localized source in the tissue, allows the cells to differentiate spatially based on detecting levels of its concentration.

To evaluate the ability of the presented system to evolve cellular patterning, the two dimensional French flag problem was scaled up to the third dimension. The target morphology was an ellipsoid and the cells had to differentiate into regions with three different colours (Fig. 6.18a). Two colour effectors were enabled, and their effect on a cell was thresholded (Table 6.7). As in the previous sections, each experiment was repeated 10 times, and the best obtained embryo was investigated. Of the 10 runs, three resulted in individuals with three colours. Remaining runs

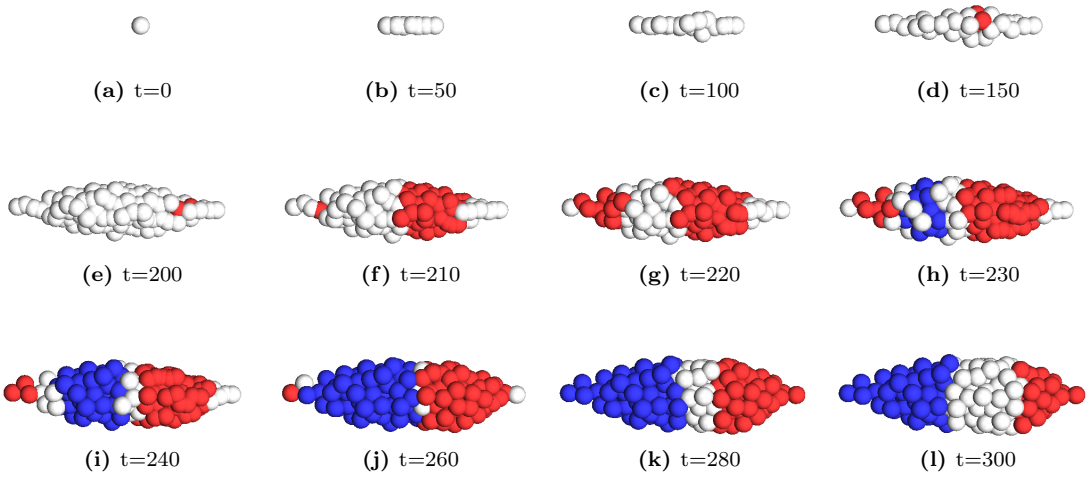
**Table 6.7:** Colour effectors and their effect on a cell used to evolve 3D French tricolour embryos. Thresholded value of concentration determines its effect on colour (and is used when calculating the fitness).

| **Colour effectors** | | **Cell colour** |
|---|---|---|
| *red* | *blue* | |
| $\geq 0.5$ | $\geq 0.5$ | pink |
| $\geq 0.5$ | $< 0.5$ | red |
| $< 0.5$ | $\geq 0.5$ | blue |
| $< 0.5$ | $< 0.5$ | white |



(a) (b)

**Figure 6.18:** Evolving 3D French tricolour embryo. (a) voxelized target pattern (small spheres represent voxels) (b) best embryo obtained in 10 independent evolutionary runs (spheres are cells).



**(a)** t=0 **(b)** t=50 **(c)** t=100 **(d)** t=150

**(e)** t=200 **(f)** t=210 **(g)** t=220 **(h)** t=230

**(i)** t=240 **(j)** t=260 **(k)** t=280 **(l)** t=300

**Figure 6.19:** Snapshots of developmental stages for the best obtained tricolour embryo (shown in 6.18b).

would become stuck in a local maxima in which embryos differentiated only into two colours or remained white, but had a matching morphology.

The best individual in 10 runs (Fig. 6.18b) appeared in generation 1944 and reached the fitness 0.792. This individual develops in a manner similar to the one discussed in section 6.2.4, with cells initially dividing in a line and later starting to divide slightly to the side (Fig. 6.19[1]). All cells begin as white, i.e., initially none of the colour effectors crosses the threshold concentration. First coloured cells appear halfway through the development ($t = 150$) and are red. Blue cells appear around the time step 230. For some of the cells, their colour changes more than once during development.

The *blue* effector remains only slightly above the threshold of 0.5 and is produced only in the cells that are indeed blue (compare Fig. 6.20a and Fig. 6.18b). In contrast, the *red* effector evolved to have its concentration increasing continuously, starting from the white region of the embryo (Fig. 6.20b). Since the *red* effector appears not to be expressed at all in the region of the embryo where the *blue* effector is expressed, this suggests that a mechanism of mutual exclusion may have evolved

[1]Videos of the results presented in this chapter can be found at `http://www.evosys.org/ecal09patterning`

(a) (b)

**Figure 6.20:** Concentrations of the colour effectors at the end of development in the best obtained tricolour embryo. (a) concentration of the *blue* effector (b) concentration of the *red* effector. Blue colour represent zero concentration, red maximum, green-yellow intermediate values.

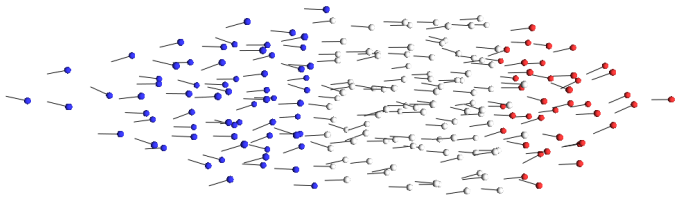


**Figure 6.21:** The orientations of division vectors at the end of development of the best obtained tricolour embryo. Small variations to orientations of divisions generate an elongated shape by exploiting the physics of the system. Small spheres represent cells' centres.

between both colour effectors. Precise analysis of the topology of the GRN turned out to be difficult due to its size (43 regulatory units), but simply looking on the connections with the highest absolute weight to each of the effectors revealed that both are most strongly influenced by the same TF. However, the connection to the *red* effector is highly excitatory (weight 1.56), whereas the blue one is inhibitory (weight -2.24).

The morphology of the investigated individual forms in the manner analogous the ellipsoidal embryos discussed in the earlier section (6.2.4) and does not pose a challenge for the evolutionary algorithm. This is largely because of the ability of the developmental process to exploit the artificial physics. A close inspection of how cells change orientations of their division vectors reveals that cells apply only minor variations to their orientation. This is enough for cells to start to divide slightly to the side of the embryo and to allow repulsive forces to take it from there (Fig. 6.21). This illustrates how both in artificial and biological systems the complexity, which would have to be otherwise present in the genome, is replaced by simple exploitation of the laws of physics.

Since the embryos were intentionally disallowed to rely on maternal morphogen gradients predeposited in the environment, the genome was investigated for the presence of genes coding for morphogens. Two morphogens with clear asymmetric production centred in the posterior and anterior of the embryo could be identified (Fig. 6.22). Although this is not the only possible solution for generating anterior/posterior axis (a single morphogen at one extreme of the embryo would suffice, in principle), all analysed individuals were found to have differential production of at least two morphogens. The presence of these morphogens is necessary for the proper development and for inducing the cell colour, but they are not sufficient by themselves and instead remain part of a dense network of interactions (as was in

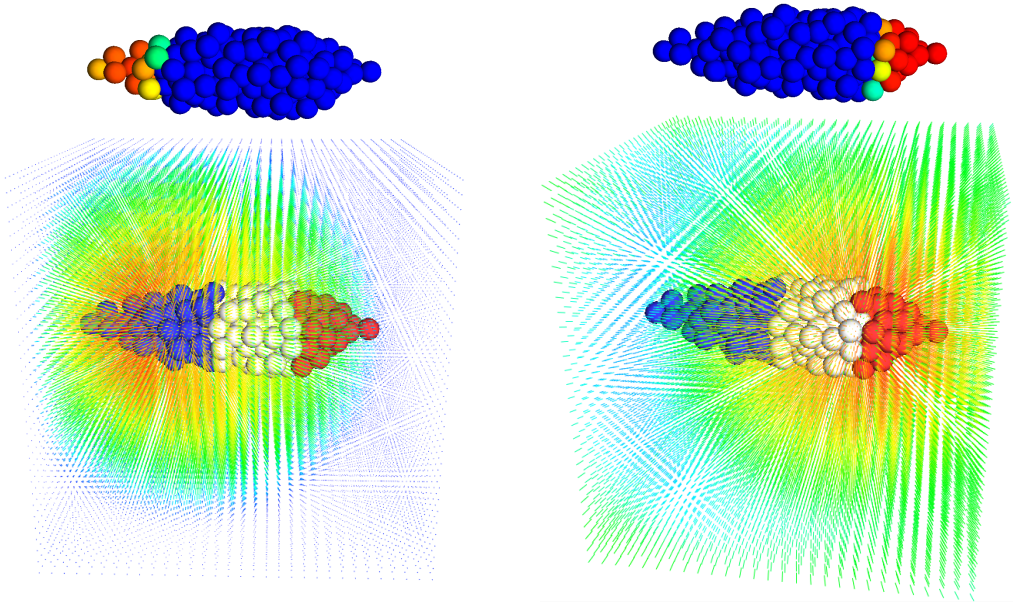


**Figure 6.22:** Self-generated gradients of positional information in the best obtained tricolour embryo employing two different morphogens (left and right column). Upper row: the concentrations of morphogens in every cell (blue-low, red-high) for each of these morphogens, Bottom row: normalized morphogen density maps in the space surrounding the embryo. Only the gradients for two selected morphogens are shown in the figure, two other morphogens with similar, localized expression were found.

the case of a *cap-less* gene, p. 129).

One of the problems identified in the initial experiments that was not an issue in experiments which focused solely on evolution of the morphology was the stability of the patterns. When evaluated by their similarity with the target pattern after 300 steps, the morphology would remain stable, but the colour pattern not necessarily so. Typically, the pattern would sweep through the embryo (driven by diffusing waves of morphogens) or oscillate. Thus, to obtain the investigated individual an average of originally proposed fitness function (Eq. 6.9) over multiple time steps was used (at times 250, 260, 270, 280, 290 and 300). Evolution with such fitness function resulted in more stable patterns, though in most of the cases the pattern would still degrade if development was allowed to continue beyond its default lifetime of 300 steps. Note however, that it is a common feature of living systems to degrade if their lifespan is extended beyond what they were selected for by evolution. The yield of individuals with stable patterns is expected to raise as the selection pressure on stability is increased (by, for example, requiring the pattern to remain stable for a longer period of time).

In another series of experiments, the stabilizing role of the maternal concentration gradients present in the environment of the developing embryo was investigated. Two maternal gradients with sources external to the embryo and located at its two extremes were added. They can be considered analogous to the gradients of *Bicoid* and *Nanos* morphogens determining the anterior/posterior axis in *Drosophila melanogaster* (the fruit fly) egg before the development starts (see, e.g., Carroll et al., 2004). In simulations performed in this work, such gradients greatly increased the

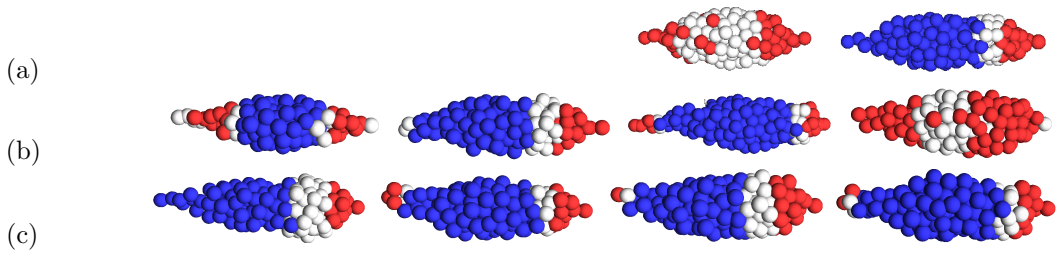



**Figure 6.23:** Robustness to cell removal of the best obtained tricolour embryo: the effects of removing a single cell from the embryo at the 2- (a), 4- (b), or 8-cell stages (c) of development; in (c) only half of possible cases is shown.

yield of stable individuals, but only if the cells were additionally prevented from producing their own morphogens (which made these maternal factors the only inducers of differential expression). This suggests the dominant role of self-produced morphogens, which in this case can be most likely explained by the diffusion model used. In the current setting, the influence of morphogens on regulation in a particular cell can be much higher than that of maternal gradients, because the effect of multiple sources (cells) sums up.

The robustness to damage of the presented individual was also briefly investigated by removing single cells at the initial stages of development (when the embryo size was 2, 4 and 8 cells). Although the morphology would remain ellipsoidal, the removal of cells would often result in a shifted colour pattern, yielding low objective fitness. However, by investigating the individuals that were damaged during development visually, one can observe that despite the shifts, three-striped patterning is usually maintained (Fig. 6.23).

### 6.4.4 Four colour embryo

To test the limits of the current setup, a more challenging target shape was defined: an embryo with 4 colour stripes. The overall configuration of the system remained identical with the previous experiment (section 6.4.3), but the target shape was modified (Fig. 6.24a). In 10 independent evolutionary runs, only a partial success was achieved, with blue stripe reduced to a single or just a few cells in the two best obtained individuals (Fig. 6.24bc).

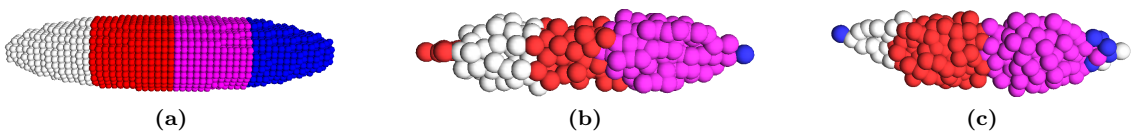
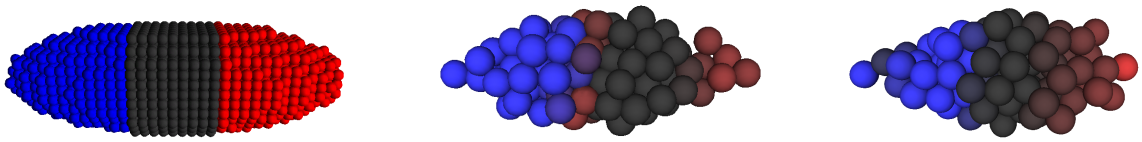



**Figure 6.24:** Four colour embryos evolved using the same setup as for the French flag embryo: (a) the target pattern, (b,c) two best individuals obtained in 10 independent evolutionary runs.



(a) (b) (c)

**Figure 6.25:** Three colour embryos evolved without colour thresholding: (a) target pattern, (b,c) two best individuals obtained in 10 independent evolutionary runs.

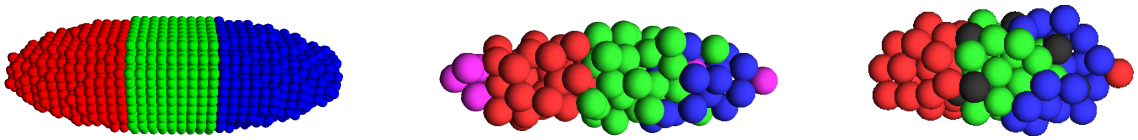
### 6.4.5 Continuous colour representation

The popular approach to generate coloured embryos in an artificial developmental system assumes that a cell changes colour only after a certain factor had crossed a predetermined threshold (employed, e.g., in Chavoya and Duthen, 2008; Fontana, 2008; Knabe et al., 2008b; Miller, 2004). This is aesthetically pleasing as it always yields clearly defined colours, but it also makes generation of sharp transition between bordering regions of different colours much easier. To see if the system can evolve patterning without colour thresholding, the same setup was used to repeat the experiment, but with the colour thresholding disabled. The equation for colour reward (Eq. 6.10) remained valid, with the only difference that $R_{xyz}$ and $B_{xyz}$ would now take values from the interval $[0, 1]$ directly corresponding to the concentrations of the *red* and *blue* effectors. The colour of every cell was determined by converting the concentrations into RGB values and was equal to: RGB($red$,0,$blue$).

Evolution under those assumptions turned out to be a harder problem. This is understandable if one considers that it was now necessary for every cell to not only reach a certain threshold of colour effector concentration, but also to maximize (or fully repress) it. Rather surprisingly, evolutionary runs (terminated after 500 generations without improvement) took much longer, taking even up to 30 000 generations (compared to about 3 000 in the experiments with thresholding). This indicates that the fitness landscape of this problem is actually much less rugged and provides many more opportunities to fine-tune individuals. However, even the two best individuals obtained in 10 runs would fail to maximize concentration of the *red* effector (Fig. 6.25).

### 6.4.6 Three colour effectors

In this setup, a third type of colour effector was enabled which, when expressed, would produce green colour of a cell. As in the experiment discussed in section 6.4.3, colour effectors are thresholded, i.e., a cell gains the colour only if the given effector has concentration above 0.5. Because cells could produce 3 types of colour effectors, they could take 8 different colour combinations (instead of 4). Colouring was based simply on the RGB combination of the effectors, which means that if none crosses the threshold a cell is black and when all cross the threshold, a cell becomes white. The target pattern (Fig. 6.26a) was defined so that cells in every area have to express only a single colour effector and repress the two others. The colour reward used in



(a) (b) (c)

**Figure 6.26:** Three colour embryos evolved to use three distinct colour effectors: (a) target pattern, (b,c) two best individuals obtained in 10 independent evolutionary runs.

the fitness function had to be modified accordingly and took the following form:

$$c_{xyz}(D,R,G,B) = \begin{cases} \frac{(R_{xyz}+(2-G_{xyz}-B_{xyz})/2}{2}, & \text{if } D_{xyz} \text{ is } \textit{red} \\ \frac{(G_{xyz}+(2-R_{xyz}-B_{xyz})/2}{2}, & \text{if } D_{xyz} \text{ is } \textit{green} \\ \frac{(B_{xyz}+(2-G_{xyz}-R_{xyz})/2}{2}, & \text{if } D_{xyz} \text{ is } \textit{blue} \\ 0, & \text{if } D_{xyz} \text{ is } \textit{empty} \end{cases} \in [0,1] \qquad (6.11)$$

where R,G and B contain the corresponding thresholded colour components for each voxel at position $(x,y,z)$. $c_{xyz}$ becomes 1 if a single colour crosses the threshold in match with the target, whereas the other two remain below the threshold. If all effectors cross the threshold, the colour reward is equal to 0.5. If the desired effector does not cross the threshold, but the two others incorrectly do, the colour reward is 0. Due to higher problem difficulty, lower quality of patterning was achieved, but indeed, the best two individuals would evolve approximately correct patterning (Fig. 6.26bc).

## 6.5 Summary

The chapter introduced a novel model of multicellular development controlled by gene regulatory networks. The development occurs in 3D environment with simulated physics and contrary to most existing developmental systems, the model does not rely on placing cells on a grid and allows them to move freely and interact through physical forces. The evolvability of the model was demonstrated by challenging the genetic algorithm with a task of evolving simple 3D structures such as an ellipsoid or an asymmetric shape, wider on one end. Providing the developing cells with an environment in which simple physics is simulated allowed them to exploit it and to simplify some of the genetic machinery that would be otherwise necessary to create the desired morphologies. After introducing cellular differentiation to the system, it was also demonstrated how the model can be used to simultaneously evolve morphology and patterning of virtual embryos, in what is likely to be a first reported version of the 3D French flag problem. When available, the embryos would rely on maternal morphogen gradients to define their morphology. However, even when maternal factors were not present, by using only morphogens produced by cells, the embryos were successful in developing into desired morphology and patterning and generated their own gradients of morphogen concentrations.

Since the system allows for full control over which genes are active, as well as allows to trace full evolutionary histories of evolved individuals, it can be used as a research platform to investigate the evolution of morphology and its relation with evolution of gene regulatory networks, essentially allowing to perform *evo-devo* experiments *in silico*. Examples of knock-out experiments that allowed to determine genes associated with certain morphological features were demonstrated.

It was further demonstrated how an ability to self-terminate development can be evolved in such embryos, by incorporating additional term in the fitness function.

Interestingly, even though not evolved to be resilient to damage, the embryos were found to be highly robust to random removal of even large fractions of cells during their development. These results show how robustness to physical damage is an emergent property of a developmental system. Furthermore, by promoting the ability to self-repair in the fitness functions, fully grown embryos capable of regrowing removed parts were obtained.

# Chapter 7

# Open ended evolution of 3D morphologies

The experiments presented in chapters 4-6 relied on the existence of the objective definition of a quality of a phenotype (i.e., the fitness function), which would guide the search algorithm towards the better quality solutions. However, this is not how biological evolutionary process works. Biological evolution progresses without a goal beyond replication of individuals or their genes. If it would be possible to observe ancestors of any given organism existing today, one would not see a continuum of forms more and more adapted to the current environment. Instead, one would see individuals that lived in different environments and that were shaped by different evolutionary pressures. Most importantly, every ancestor would appear to be highly adapted to the environment it happened to live in. The complex and intertwined history of all living organisms is one of the main sources of life's incredible complexity and variation. It has been a Holy Grail of artificial life systems to recreate this complexity *in silico*. Artificial life systems that attempt to do so are referred to as open ended evolution systems (see, e.g., Adami et al., 1994; Channon and Damper, 2000; Ray, 1992; Yaeger, 1993, for some examples of such systems).

The problem that any optimization method has to face is the fact that fitness landscape can have many local maxima. What may initially seem a good way to improve existing solution may (and usually will) turn out to be an evolutionary dead alley. The defining feature of evolutionary algorithms is the use of a population of solutions in an attempt to deal with local optima. Even though a single individual may be trapped in such an optimum, other individuals can explore other areas of the fitness landscape. Finding the balance between exploitation of the local gradient towards improvement and the amount of exploration performed in other areas of the search space has been for years the subject of research in the field of evolutionary computation. Many problem specific algorithms and various genome encodings have been and are continuously proposed to address it. Still, many real life optimization problems exhibit fitness landscapes that are so complex that genetic algorithms perform very poorly. This is especially the case for problems where fitness landscapes

are deceptive, i.e., finding the optimal solution would require the search to proceed in a different direction than directly towards the best solution for extended periods of time.

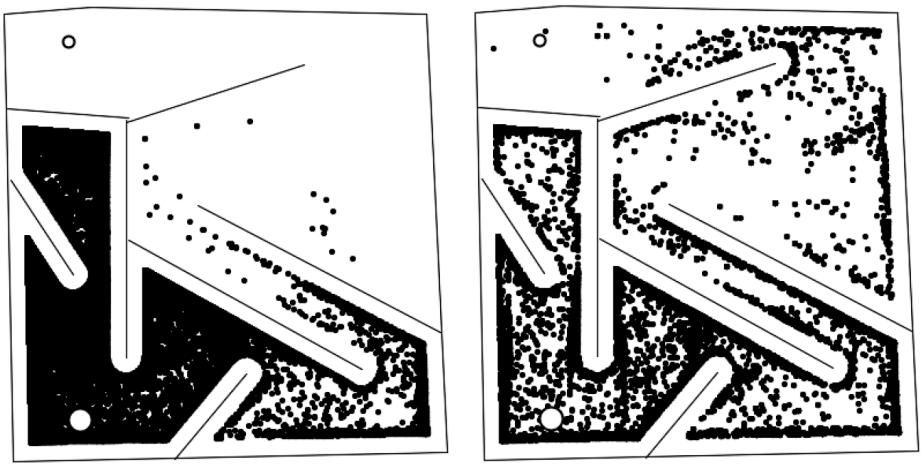
## 7.1 The Novelty Search algorithm

The novelty search algorithm, proposed by Lehman and Stanley (2008) is an evolutionary algorithm that attempts to solve optimization problems by employing a radical idea of abandoning the fitness function. The fitness function in the original evolutionary algorithm is replaced with a measure of phenotypic novelty. The algorithm favours those individuals in the population that have the most novel phenotypes. This is done even when it is possible to formulate a well defined fitness function. However, in a deceptive search space, attempting to optimize the fitness poses a risk of solutions getting trapped in a local suboptima. The method of Lehman and Stanley proposes to focus on exploring novel areas of the search space rather than on heading directly towards the solution. This is achieved by assigning a higher score not to highly fit individuals, but to solutions that are different from those that have already been found. Perhaps surprisingly, novelty search does not result in a blind walk through the space of solutions. The method has been shown to outperform classical fitness based-search for problems such as guiding a robot through a labyrinth and evolving bipedal locomotion (Lehman and Stanley, 2011). Several methods that help evolutionary search by promoting genetic diversity (e.g., by rewarding individuals for having genomes different from those of other individuals) were previously proposed (see, e.g., Mahfoud, 1995; Sareni and Krahenbuhl, 1998). However, compared with other methods of sustaining diversity in the population of solutions, novelty search focuses entirely on the diversity of the phenotypes, not on the diversity of the genotypes.

The core of the novelty search algorithm is the measure of phenotypic novelty which requires a measure of distance between any two phenotypes. For an algorithm to function, a method must exist, that will allow to calculate the distance between any two phenotypes. First, a matrix of distances between phenotypes of individuals in the population is calculated. If a given individual is very similar to others, its average distance to its closest neighbours will also be small. Novel individuals will tend to be located further from the others, thus having higher average distance. Lehman and Stanley (2011) define the measure of novelty as the average distance of an individual $x$ from its $k$-nearest neighbours. Thus, novelty is a measure of sparseness of the phenotypic space surrounding the individual:

$$\rho(x) = \frac{1}{k} \sum_{i=0}^{k} \text{dist}(x, \mu_i) \tag{7.1}$$

where $\mu_i$ is the $i$-th nearest individual in the behaviour space with respect to the distance metric *dist* and $k$ is the number of nearest individuals used to compute



(a) (b)

**Figure 7.1:** Comparison of novelty-based (a) and fitness-based (b) search in a deceptive maze problem. Points represent final positions of the robots trying to escape the maze in subsequent generations. Reproduced from Lehman and Stanley (2011).

sparseness of the phenotypic space surrounding individual $x$.

The candidates used to compute the distance are recruited from the current population as well as from the past. The latter is important. Otherwise, the evolving population could backtrack in the search space, rediscovering phenotypes found earlier. However, computing distance from all the past individuals can be computationally prohibitive. This is why only a selection of past individuals is used. This selection is known as the archive. Whenever a considerably novel phenotype (i.e., with novelty above a threshold) is discovered, it will be copied to the archive, and it will remain there as a representative of its type. Furthermore, setting a valid threshold for adding individuals to an archive is domain-specific and it can change during evolution. The method proposed by Lehman and Stanley (2011) uses an adaptive algorithm. The threshold is increased if too many individuals are added within given number of generations and lowered if no new individuals are added for an extended period of time. In one of the examples investigated by Lehman and Stanley (2011), robots that can guide through a deceptive labyrinth were evolved using the novelty-based and fitness-based evolution (Fig. 7.1). The distance function between phenotypes in a novelty-based evolution was the Euclidean distance between the positions of robots at the final steps of their lifetime. As can be observed, fitness based approach (Fig. 7.1a) gets trapped in a deceptive local suboptimal solution, whereas novelty based search successfully explores the space of possible robot behaviours until it finds the solution (Fig. 7.1b). In this particular toy problem, the novelty search algorithm benefits from the closed environment as the walls stop it from exploring the infinite space outside of the labyrinth. As authors point out, for some domains it may be necessary to constrain the space of possible solutions.

## 7.2 Novelty Search for 3D Morphologies

The novelty search algorithm presents an interesting opportunity to create an open-ended system for evolving 3D morphologies. On one hand, this creates a system with more realistic evolutionary pressures. On the other, it can be seen as a generic way of exploring what kind of morphologies are reachable in a given artificial embryogeny system. The necessary changes to the system are limited to redefining the way fitness value of each individual is computed by replacing it with the calculation of novelty value (Eq. 7.1), based on a definition of distance function for two morphologies. The developmental model and the genetic algorithm remain unchanged.

### 7.2.1 Distance function

To assess the novelty of each new phenotype, a voxel based approach was used, similar to the approach used for fitness-directed shape evolution (Chapter 6). Equation adapted from Eq. 6.7 (p. 123) was used to calculate the distance between two individuals $A, B$ by counting the number of voxels that are different between the two discretized shapes:

$$d_m(A_{xyz}, B_{xyz}) = \frac{1}{s_x s_y s_z} \sum_{x=0}^{s_x-1} \sum_{y=0}^{s_y-1} \sum_{z=0}^{s_z-1} |A_{xyz} - B_{xyz}| \tag{7.2}$$

where $s_x$,$s_y$,$s_z$ are dimensions of the cuboid containing the shapes, and $A_{xyz}, B_{xyz} \in \{0, 1\}$ is the voxel state at position $x, y, z$ (1 is filled, 0 is empty). $s_x, s_y, s_z$ are set at the beginning of an evolutionary run and have to be sufficiently large to encompass a large spectrum of possible morphologies. This also means that a typical calculated value of $d_m$ will tend to be small, as the volume of each morphology is much smaller than that of the cuboid. However, this does not affect the novelty search algorithm, because it relies only on the relative distances between phenotypes.

One of the limitations of voxel-based comparison of morphologies is that it is sensitive to rotations and shifts. This is not a serious problem for goal directed evolution, but could be easily exploited by evolution searching for novel individuals. For example, novel individuals could be formed by introducing small rotation at the initial stage of development. This is considered undesirable, so a further refinement to the distance function based on the Principal Component Analysis was used.

The goal of the Principal Component Analysis (PCA) is to rotate and shift the original coordinate system for a given set of $n$-dimensional observations so that the variance of the first coordinate is maximized in the new coordinate system, then the variance of the second coordinate, and so on. In the refined method of fitness calculation, PCA is applied to a set of cell positions, which results in a shape rotated so that the $X$ axis is aligned with the longest axis of the embryo (Fig. 7.2). Realigning all morphologies before comparison removes the effects of rotations and shifts. However, it has its limitations. For example, if two very similar shapes that are elongated in different directions are compared, PCA will rotate them completely

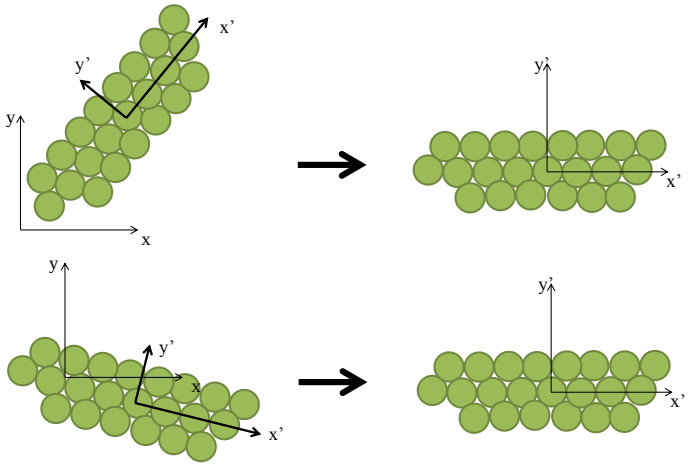



**Figure 7.2:** Illustration of PCA based rotation for shift and rotation invariant morphology comparison. Upper and bottom row: two identical but shifted and rotated multicellular morphologies become aligned and rotated so they can be directly compared. The PCA-based algorithm operates on a vector of cell positions and ignores cell sizes.

differently, even though the difference between the shapes may be minimal. To avoid this type of effects, PCA was only used as a secondary mechanism for shape comparison. The shapes were compared two times, with and without PCA rotation. The minimum difference between the two comparisons was chosen as a distance between morphologies:

$$\text{dist}(A, B) = \min(d_m(A, B), d_m(\text{PCA\_transform}(A), \text{PCA\_transform}(B)) \quad (7.3)$$

A limitation of this distance measure is that PCA does not give directions for principal axes. Ideally, both the shapes as well as their mirrored versions should be compared. This was not implemented for the sake of simplicity and under expectation that if it ever becomes an issue, the archive may be populated with mirrored versions of similar morphologies. The full algorithm used to calculate novelty values for all individuals in a population is presented as a pseudo code in the Appendix (Listing 4, p. 181).

## 7.3 Results

The experimental setup assumes similar physical properties of the development to those used in the earlier experiments on the evolution of shape and morphology. Embryos are allowed to develop for 400 time steps, with a hard limit on the number of cells set to 100. The minimal viability criteria are: at least one cell and termination of divisions in 300 time steps, so that for the remaining 100 time steps the structure has the time to equilibrate. Genomes in initial populations are initialized with a single regulatory unit, consisting of a single promoter and a single product, so that complexification of the networks could be more easily observed (in evolutionary simulations described in chapters 4-6, genomes were initialized with 5 or more regulatory units). Furthermore, sexual crossover is not used. Although the

**Table 7.1:** Essential GA parameters used in the experiments with the open ended evolution of 3D morphologies. Additional parameters are provided in the Appendix (Table C.4, p. 186).

| Parameter | Value |
|---|---|
| Population size | 300 |
| Elite individuals | 0 |
| Asexually created individuals | 300 |
| Individuals through crossover | 0 |
| Initial population | randomized genomes, 1 regulatory units each |
| Termination condition | 5000 generations |
| Selection | binary tournament ($k = 2$, $p = 0.75$) |
| Cell limit during development | 100 |
| Developmental time steps | 400 |

**Table 7.2:** Types of products and promoters enabled in the experiments with the open ended evolution of 3D morphologies and the interpretation of subsequent input and output elements.

| Promoter types | Product types | External factors | Effectors |
|---|---|---|---|
| additive | transcription factor<br>morphogen | "1" (fixed high concentration)<br>4 maternal morphogen sources at (0,0,10), (9.43,0,-3.33), (-4.71,8.16,-3.33), (-4.71,-8.16,-3.33) | divide (threshold 0.8)<br>die (threshold 0.5)<br>rotation $R_H$<br>rotation $R_L$<br>rotation $R_U$ |

crossover was observed to improve evolvability in earlier experiments (Chapter 6), it is disabled here so that the full evolutionary history of an individual in the final generation can be traced backwards to a single ancestral individual in generation 0. The population size is set to 300 individuals, there is no elitism and GA runs for 5000 generations (see summary in Table 7.1). Four maternal morphogen gradients were enabled, with sources equidistant from the centre of coordinate system and from each other (Table 7.2).

One thing that should be emphasized about the approach used in this chapter, is that this is not a random search through a space of possible genotypes. Such search would result mostly in individuals incapable of division. Those, for which multicellular development occurs, would most often form either clumps of cells or some degenerate shapes (such as cells growing in a line). The novelty algorithm searches the space of possible phenotypes, i.e., of possible morphologies. Each novel morphology becomes favoured by the genetic algorithm and increases in frequency in the population. In result, this reduces the novelty of this particular morphology and so it will soon be replaced by other, previously unknown morphologies. This creates an artificial embryogenesis system in which there is a constant pressure to innovate.

### 7.3.1 Evolved morphologies

Figure 7.3 presents the novelty history of the evolving population over 5000 generations. The average level of novelty value in the population raises quickly during the first 500 generations and then remains relatively stable over time (although ex-

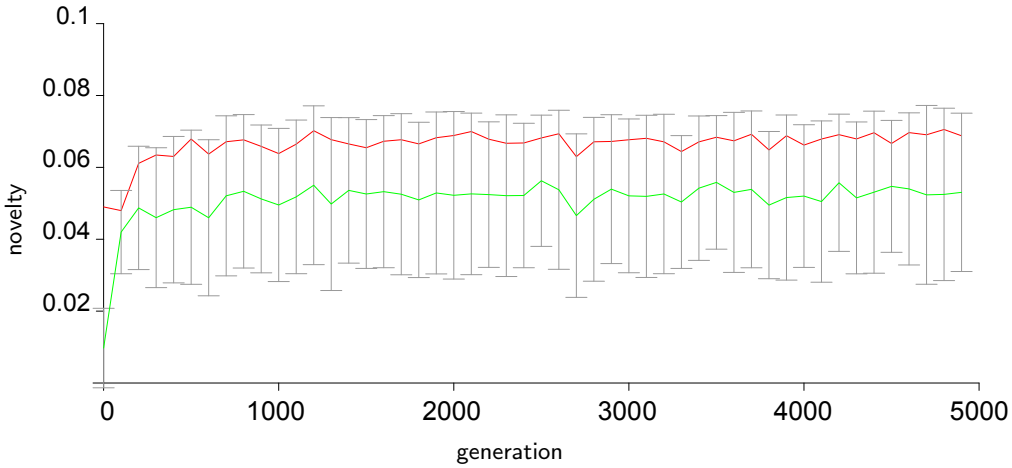



**Figure 7.3:** Maximum (red) and average (green) novelty value in the population during 5000 generations of novelty-driven genetic algorithm (measured every 100 generations). Bars indicate standard deviation for novelty in the population.

periments running over 30 000 generations suggest that the average level of novelty slowly increases). As opposed to the evolution directed by an objective fitness function, a stable level of novelty does not mean stagnation in the evolving population. New forms are continuously being discovered. If stagnation ever occurred, it would be observed as novelty level decreasing over time. The proposed approach creates an artificial evolution system in which all the genomes take part in the Lewis Carroll's Red Queen's Race, where "it takes all the running you can do to keep in the same place".

A visual inspection of the morphologies generated by genomes in the final population reveals the level of morphological diversity that exists at this time (Fig. 7.4). Multiple distinct forms exist together with their recognizable variations. Many individuals have appendages and display some level of symmetry. The range of apparent morphological complexity of these morphologies far outreaches the complexity of the shapes that were achievable using goal oriented evolution in experiments using the GA (sections 6.2 and 6.4).

### 7.3.2 Novelty search archive

The novelty search archive is an essential part of the novelty search algorithm and stores past individuals. Individuals in the current population are compared not only to themselves, but also to already "extinct" individuals. The archive consists of individuals that were added either because they were novel at the time or, at random. The probability of random addition was set to $p = 5 \cdot 10^{-4}$ and applied only to individuals meeting viability criteria. The reason for adding an individual to the archive was also stored, so that each type could be later easily filtered. This resulted in the archive with 1088 individuals at generation 5000. 432 of them were added because their novelty was above the threshold at that time.

The analysis of the novel individuals stored in the archive shows that evolution started with visually simple morphologies (spherical clumps of cells or mats,

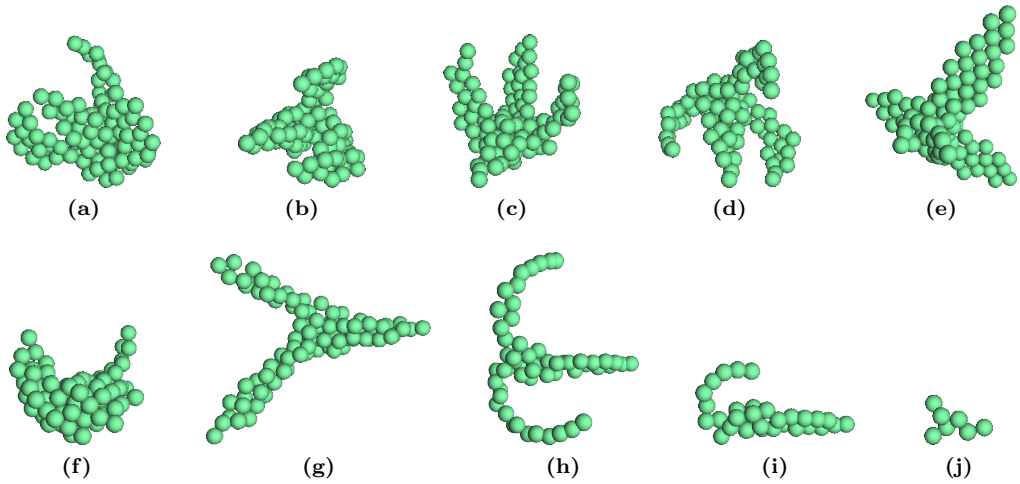



**Figure 7.4:** Morphological diversity in the population in generation 5000 of novelty-driven evolution (a sample of visually interesting individuals from a population of 300).

Fig. 7.5). At that time, genotypes were still small because the evolution started with random genomes consisting of single regulatory unit, basically allowing only for division. As the evolution progressed, more complex morphologies appeared, with protrusions in various directions, as well as more complex 2D shapes. The latter are likely degenerate structures in which a genetic element allowing to divide away from one plane was lost or damaged. Many of the morphologies look similar, suggesting that once appendages evolve, small adjustments to genes controlling their development result in a continuous stream of variation.

### 7.3.3 Evolutionary history

Since initial genomes consist of a single regulatory unit, morphological innovations would not be possible without genetic duplications, and the results show that, indeed, genomes grow over time (Fig. 7.6a). The number of non functional elements (such as TFs that do not bind to anything or promoters to which nothing can bind) remains at the level of 15-30% (Fig. 7.6b). Thus, the lengthening of the genome means also the growth of the number of vertices and the number of edges in the GRN (Fig. 7.7). The increase was not uniform over time. For example, between generation 3700 and 4600 the average size of the genome decreased twofold and also resulted in comparable decrease in the average number of connections in the networks. Then, the genomes would start growing again, and they regained most of their previous size in just 400 generations. However, this time the growth was mostly due to accumulation of disconnected ("junk") genetic elements: the size of the regulatory graph did not increase as much as the size of the genome did (compare Fig. 7.6a and Fig. 7.7).

Because the sexual crossover was disabled, a single individual can easily be traced backwards to the individual in the initial generation from which it evolved, together with all the mutations that it had accumulated over time. This allows to observe how mutations influenced morphology and also to trace all mutations in the lineage

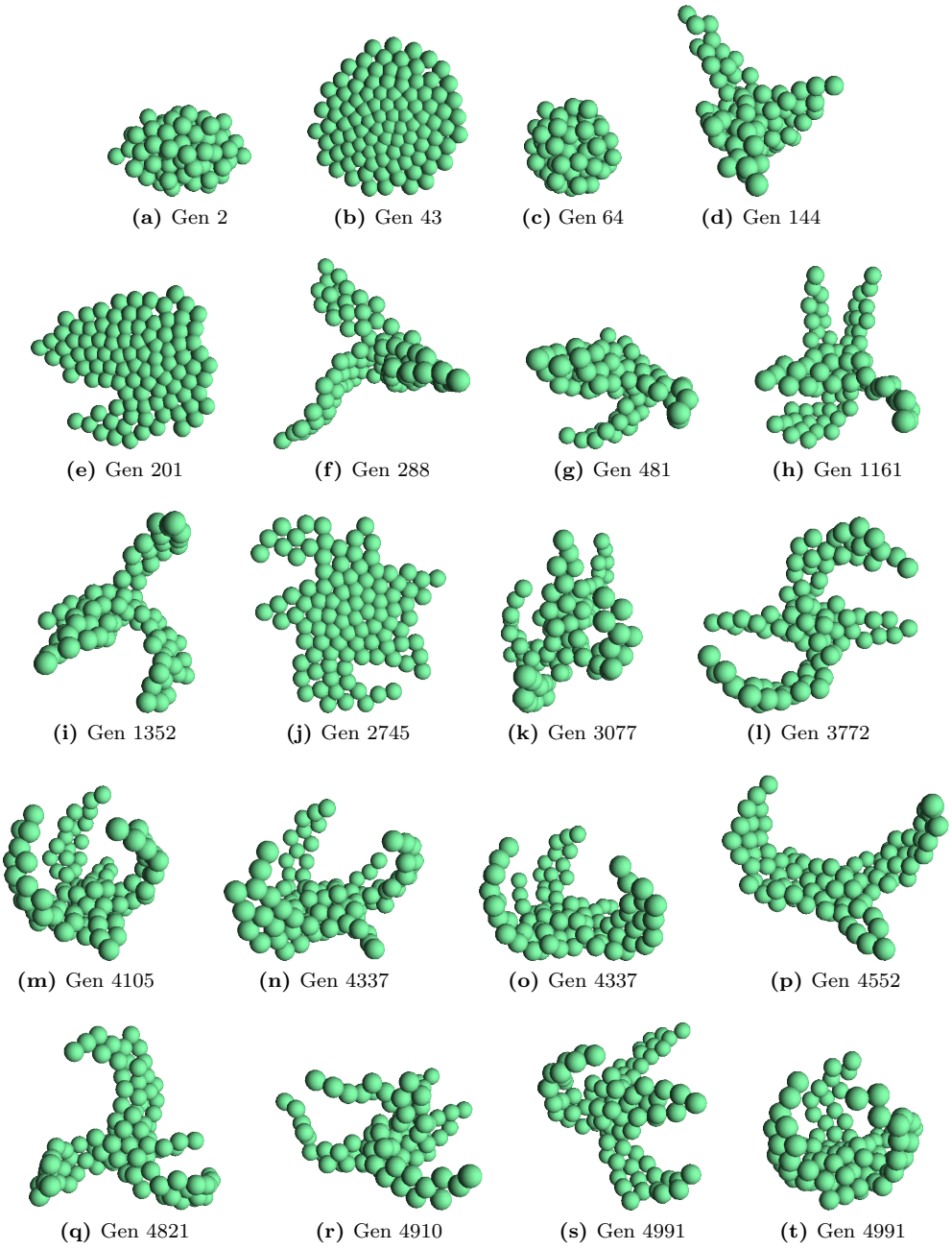



**Figure 7.5:** A sample of novel individuals stored in the archive after 5000 generations of novelty-driven evolution.

that contributed to the final state. To allow for such an analysis, all individuals that existed during the evolutionary run were stored, together with the identifiers of their parents. Many of the ancestors would not contain any mutations, so only individuals that actually contained some changes in the genome were considered (many had the same phenotype, because many changes to the genome are neutral).

An analysis of the lineage of the individual that had the highest value of novelty in generation 5000 (Fig. 7.4a, Fig. 7.8t) was performed. It was found to have 2083 genetically different ancestors. The evolution started from a spherical embryo, and complexification of the structure consisted of adding appendages and modifying

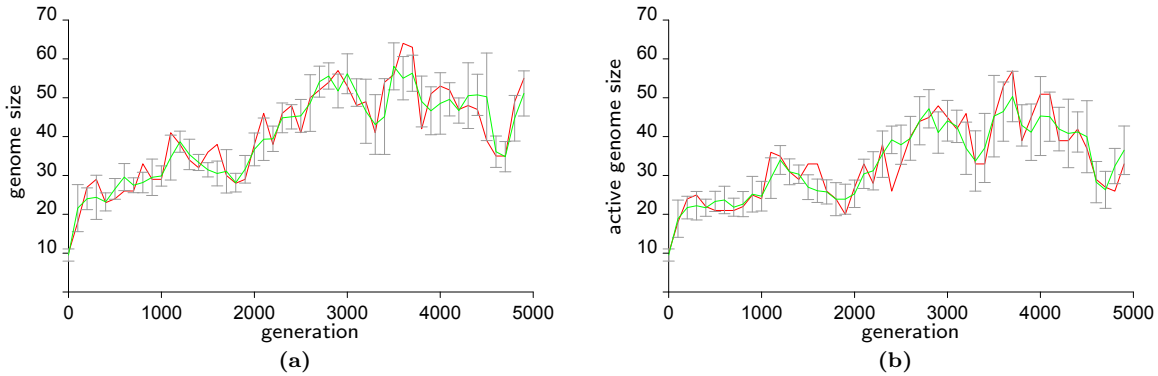



**Figure 7.6:** Genome and active genome size during the evolutionary run. Panel (a) shows the genomes size, (b) the size of the active part of the genome. The red line corresponds to the individual with the highest novelty in a given generation, the green line to the average, bars indicate standard deviation. The values were determined every 100 generations.

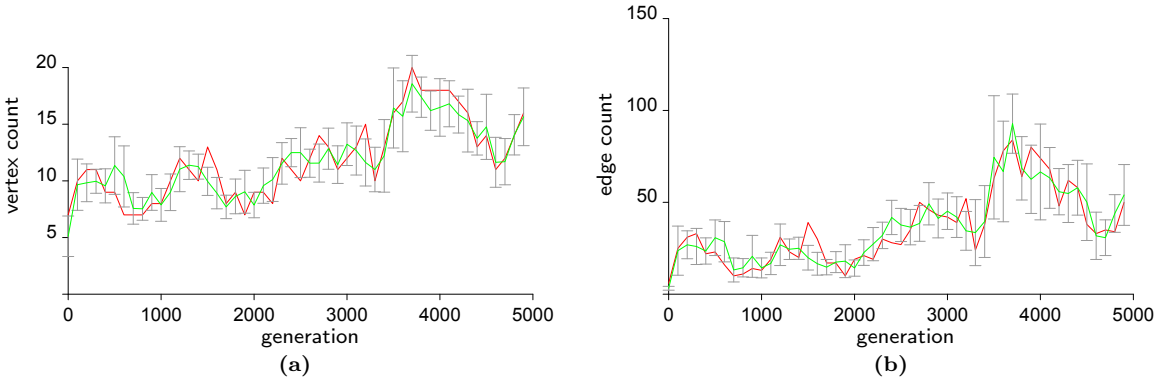
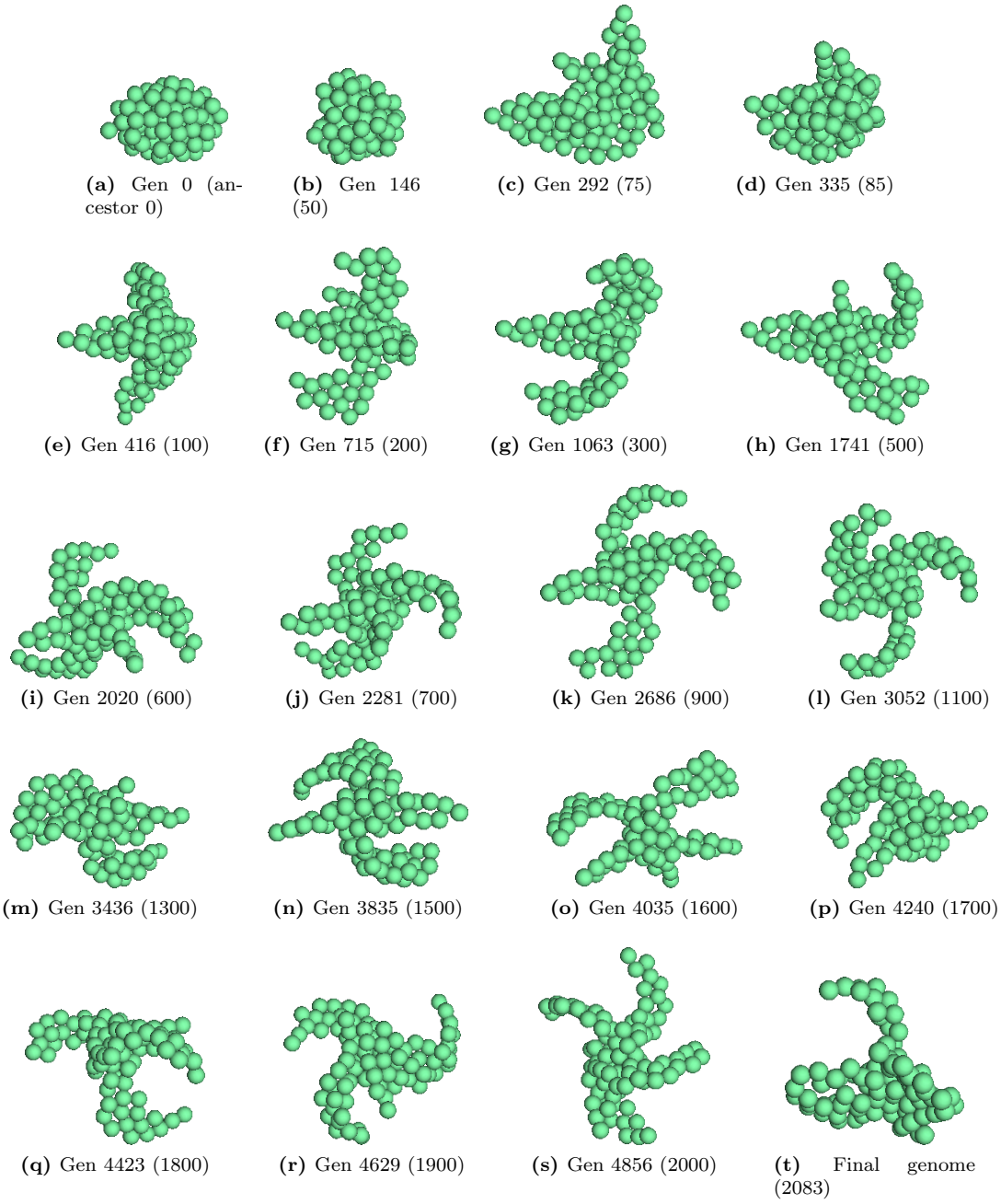



**Figure 7.7:** Vertex (a) and edge (b) count in the regulatory network during novelty-driven evolutionary run. Note that inputs and outputs also constitute vertices, thus the initial number of vertices is 1 (single regulatory unit) plus inputs and outputs. The red line corresponds to the individual with the highest novelty value in a given generation, the green line to the average, bars indicate standard deviation. The values were determined every 100 generations.

their directions (Fig. 7.8). Typically, morphological changes between parent and offspring were small. Individuals separated by a few hundred generations are still recognizable as variations of the same morphology and often share distinct structural features. This shows that the morphologies do not explore the phenotype space in large random jumps. Instead, evolution usually progresses by small variations to the phenotype.

Even though evolution progressed in small steps, the history of the chosen individual contains morphologies that are very different from its final morphology. If an attempt was made to evolve the same morphology using an objective function based on similarity to this final shape (as in Chapter 6), the GA would stand a poor chance of approximating this particular target. Even though the morphologies of ancestors were dissimilar from the final individual and would have very low fitnesses, they were important stages in the evolutionary history of this individual. An experiment conceptually related to this discussion was recently performed by Woolley and Stanley (2011) using an online, human driven image evolution system PicBreeder (Secretan



**(a)** Gen 0 (ancestor 0) **(b)** Gen 146 (50) **(c)** Gen 292 (75) **(d)** Gen 335 (85)

**(e)** Gen 416 (100) **(f)** Gen 715 (200) **(g)** Gen 1063 (300) **(h)** Gen 1741 (500)

**(i)** Gen 2020 (600) **(j)** Gen 2281 (700) **(k)** Gen 2686 (900) **(l)** Gen 3052 (1100)

**(m)** Gen 3436 (1300) **(n)** Gen 3835 (1500) **(o)** Gen 4035 (1600) **(p)** Gen 4240 (1700)

**(q)** Gen 4423 (1800) **(r)** Gen 4629 (1900) **(s)** Gen 4856 (2000) **(t)** Final genome (2083)

**Figure 7.8:** Select direct ancestors of a final individual with the highest value of novelty in generation 5000. Labels show generation number and a the subsequent number in a chain of mutated genomes. The individual shown in (t) is the same morphology as in Fig. 7.4a, but presented from a different point of view.

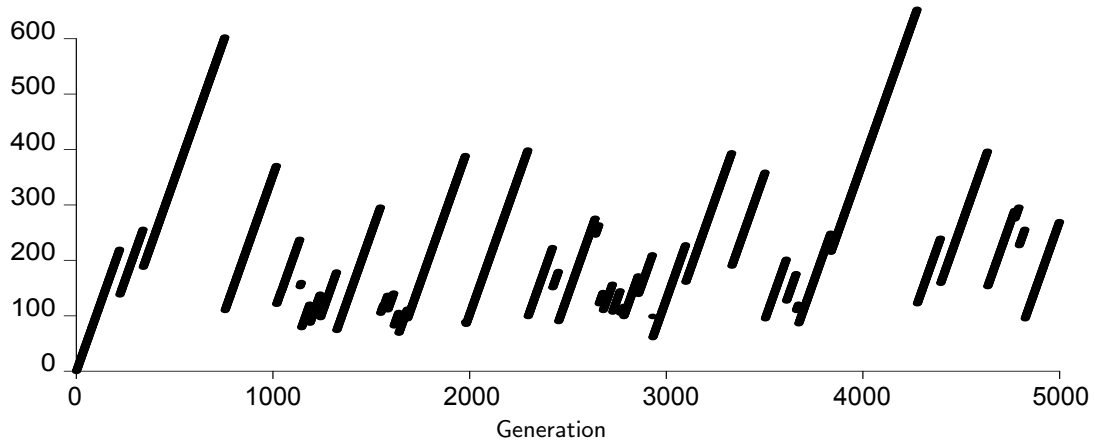



**Figure 7.9:** Time from the most recent common ancestor for the whole population in every generation in the novelty-driven evolutionary run.

and Beato, 2008). Patterns were evolved using human selection, but an attempt to evolve such human selected patterns *de novo* using a GA was unsuccessful.

### 7.3.4 Evolutionary time from the most recent common ancestor

The distance from the most recent common ancestor for the whole population in every generation was traced to provide an insight into the level of diversification of the evolving lineages (Fig. 7.9). For example, in the 230th generation, the distance has a value of $\sim 140$, which means that all individuals in this and all following generations derive from a single individual that existed around generation 90 and none of the remaining 299 individuals in that generation had left any surviving descendants. The longest evolutionary time from a common ancestor for the whole population exceeded 600 generations on two occasions (Fig. 7.9). The average time (233.9 generations) suggests that certain new individuals had considerable competitive advantage over the rest of the population and their descendants took over the whole population relatively quickly.

### 7.3.5 Visualization of the phenotype search space

Multidimensional scaling (MDS) was applied to the individuals stored in the archive during evolutionary run to visualize how the space of possible phenotypes is explored over time. MDS (Cox and Cox, 2000) is a statistical technique used in exploratory data analysis which allows to visualize dissimilarities between data samples. It can also be used to visualize the results of a cluster analysis. MDS takes a matrix of distances (dissimilarities) between $N$ data samples as an input and assigns each sample a point in $K$-dimensional space ($K < N$) so that the distances between associated points match the distances between samples in the original distance matrix with a minimum error. Thus, MDS is capable of generating a geometric interpretation of multivariate data, attempting to minimize the error introduced by the reduction of data dimensionality. The implementation available in the statistical package R

(R 2.13, 2011, function cmdscale) was used to perform classical multidimensional scaling of a data matrix, also known as principal coordinates analysis (Gower, 1966).

The function used as the distance measure between any two individuals for novelty search algorithm (Eq. 7.3) was also used to compute the distance matrices for MDS. Because of the number of individuals in evolutionary histories is very large (300x5000=1.5 million), only individuals found in the novelty search archive were selected for analysis. Because individuals added to the archive by chance alone represent an unbiased sample of viable individuals from the whole history of evolution, they allow to visualize the spectrum of morphologies existing in an evolutionary run. The MDS analysis of all 656 individuals of this type was performed (Fig. 7.10).

Reducing something as complex as morphology to a single point in 2D represents a huge loss of information. Nonetheless, strong patterns in the data are apparent: three major clusters of morphologies can be observed and visual comparison of random pairs of neighbouring points confirms that they have similar morphologies.

Unless the search through the space of possible phenotypes is very chaotic and small changes to the genotype result in radical changes to the phenotype, one can expect that individuals that appeared close in time are also similar phenotypically and thus located closer to each other in Fig. 7.10. This is indeed the case: individuals seem to be surrounded by individuals that appeared close in time (coloured similarly in Fig. 7.10). Still, many individuals from later generations were similar to those from earlier evolutionary periods. It is possible that mutations and the loss of genes in later individuals results in degenerate (e.g., spherical) morphologies, similar to genetically simpler ancestors. Such an appearance of new individuals revisiting areas of the search space with less complex morphologies is expected. In addition, many of the original forms ("body plans") emerge early on, and later generations build upon variations of them, not unlike what is thought to have happened during the evolution of life on Earth.

It seems that the most salient small cluster of individuals (Fig. 7.10, cluster A) consists predominantly of individuals formed earlier in evolutionary history (blue and green in Fig. 7.10). Inspection of morphologies in that cluster revealed that it groups flat morphologies (consisting of a single layer of cells). Most of them are circular, but some interesting variations are also present. Closeness of the points suggests that this is a degenerate lineage that does not allow for much variation between phenotypes. Some new individuals were added to this cluster at the late stages of evolution (red points), perhaps because a loss of genes allowing to reorient divisions in the 3rd dimension is an event that happens in many later lineages and will result in a novel flat shape.

Intermediate morphologies can be found between the three clusters (Fig. 7.10). Large part of the individuals from initial generations remain close to each other (Fig. 7.10, cluster B, dark blue). This cluster was found to group spherical individuals (also some from later generations) and the further from the centre of its weight, the more pronounced are the protrusions. The more dispersed cluster C groups a variety of morphologies with appendages, many of them from middle (greenish points) and late (reddish) stages of evolutionary history.

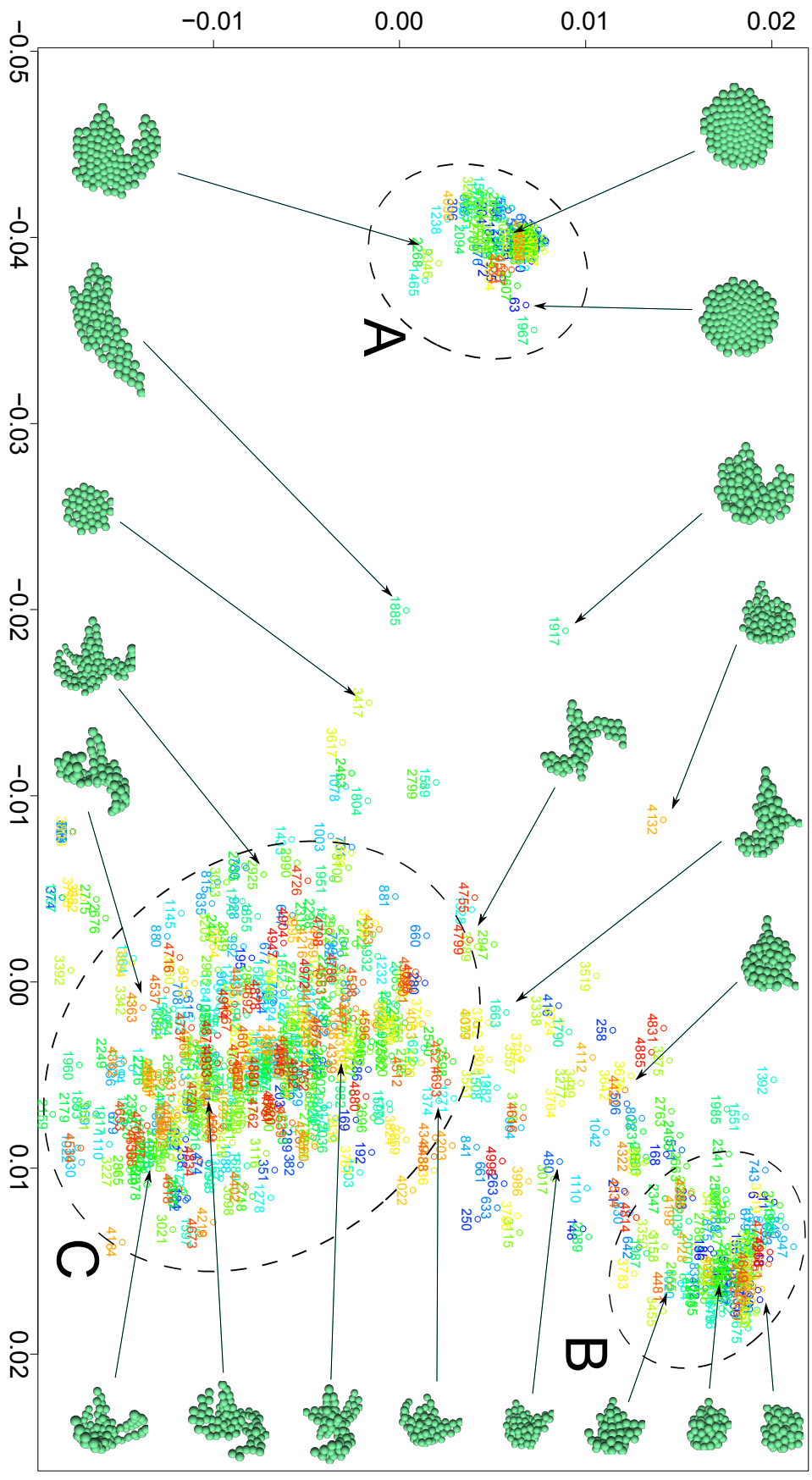



**Figure 7.10:** Geometric representation (obtained through MDS) of the similarity between 656 viable random individuals that existed during evolution. Each data point is labelled with generation number of the given individual and colour corresponding to the generation it comes from. Individuals sampled from initial generations are blue, those from the final generations are red and individuals that existed around 2000-3000th generation are green. Morphology of selected individuals is overlaid on the graph. Approximate clusters A,B and C were determined visually.

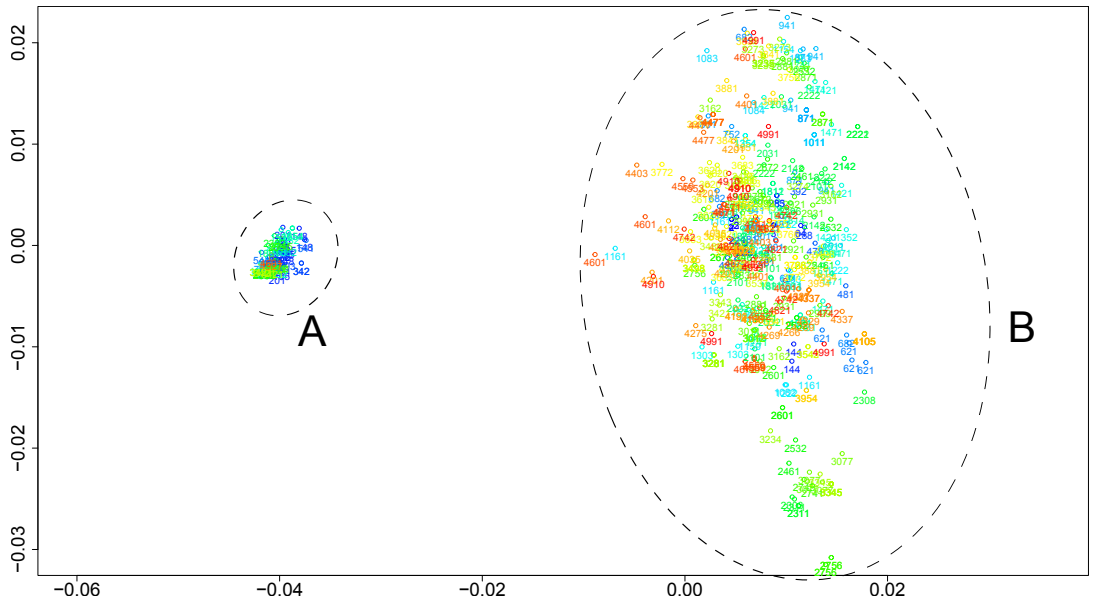



**Figure 7.11:** Geometric representation (obtained through MDS) of the similarity between 432 individuals added to the novelty archive because of their novelty level exceeding the threshold. Each data point is labelled with generation number of the given individual and colour corresponding to the generation it comes from. Approximate clusters A and B were determined visually.

To determine how the subset of novel individuals from the archive differs from the subset of individuals added by chance, an MDS on these individuals was performed in the same manner. Two salient clusters are evident (Fig. 7.11). Cluster A again groups the flat individuals. The second cluster (B) has many individuals from the initial generations in its centre (blue points) whereas its outer areas are populated with more recent morphologies (green and red). This may be interpreted as the visualization of evolution progressing by exploring new areas of the phenotype space over time.

### 7.3.6 Repeatability

Each experiment with novelty search using different random seed would result in a different evolutionary trajectories in which a different set of morphologies was discovered. Some similarities between the runs can however be observed: typically, evolution would start with spherical or flat embryos and then discover some forms of appendages (Fig. 7.12). Then, a variety of morphologies is generated, with appendages growing in different directions, bending and twisting. Such appendages were also observed to increase or decrease in number, while the overall morphology remained unchanged. This shows how evolution can act on larger morphological features. The crown-like morphologies with two, three or four appendages have been observed in many other runs and seem to be a recurring form in repeated experiments. In contrast to the earlier discussed run, evolution was observed to employ apoptosis as a way to generate new solutions, resulting in individuals that remove their centres at the end of development (Fig. 7.12ef).

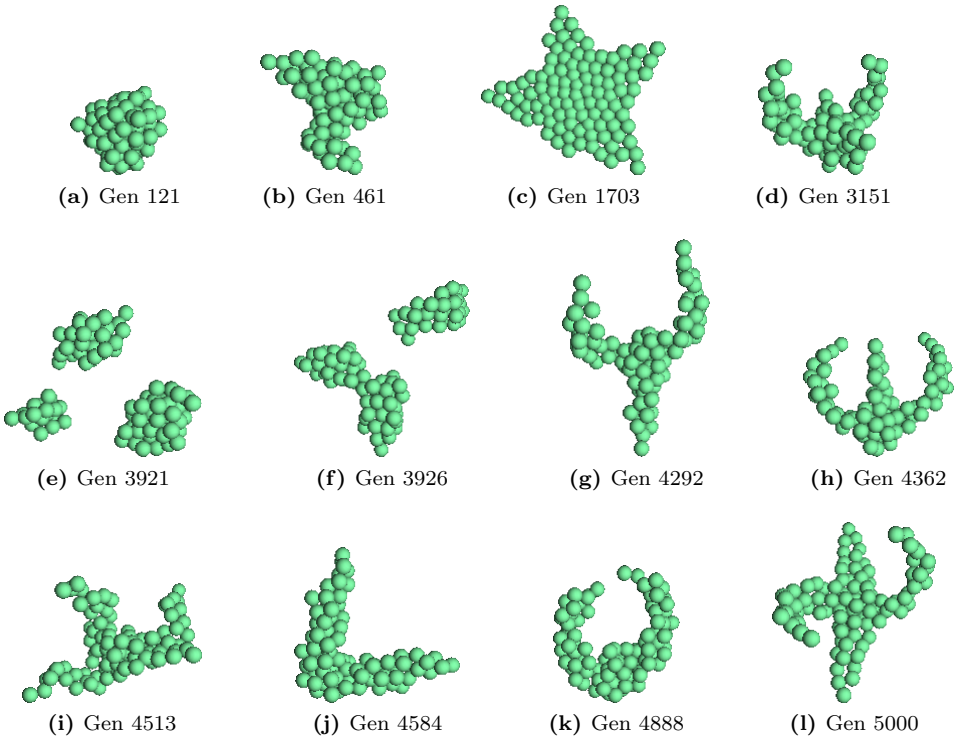



**Figure 7.12:** A sample of novel morphologies stored in the archive at the end of experiment repeated with different seed of random generator. (ef): in some of the individuals apoptosis would occur during development.

## 7.4 Summary

The apparent complexities of morphologies that could be obtained using open ended evolution were found to be clearly higher than that of morphologies obtainable using an objective fitness function in Chapter 6. The system was observed to explore the phenotype space in small steps, with mutations typically resulting in small changes to the phenotypes. However, by retracing the evolutionary history of a particular individual, it could be seen how its further ancestors could be very different from it. This is very similar to what is observed in biological evolution. The difference between complexities of morphologies in this chapter and in Chapter 6 illustrate the high deleterious effects of attempting to evolve complex morphologies using an objective function: trying to directly evolve a more complex morphology can mean that evolutionary milestones necessary to reach it may never be found.

Furthermore, the results obtained in this chapter demonstrate how the novelty search algorithm can be used to create an open ended evolution system for 3D morphologies, in which a constant evolutionary pressure for new shapes exists. This can be seen as a way to explore the space of achievable phenotypes for a given developmental model as well as to observe any inherent biases towards particular morphologies. Importantly, the presented approach could by applied to other existing developmental systems.

At the same time, the results show how a fitness driven developmental system, with minor variations to the code, can be transformed into an open ended evolu-

tion system. This allows to create a much more biologically plausible setting for performing biologically inspired experiments related to investigation of the relations between evolution of genomes, regulatory networks and morphologies.

The analysis performed in this chapter also demonstrated how the MDS algorithm can be used to visualize the exploration of new areas of the phenotype space over time, providing a new way of visualizing the progress of evolution in an open ended alife system.

# Chapter 8

# Summary and future work

The objective of this thesis was twofold. One was to investigate properties of the biologically inspired model of development and gene regulatory networks (GRNs) in the context of their potential practical applications. The limitations of direct genotype-phenotype mappings mean that only relatively simple structures/designs can be created using such an approach. Hence, the field of evolutionary computation is on a constant lookout for new, indirect encodings that will allow to automatically create more sophisticated designs or designs with certain desirable properties (such as failure tolerance). To achieve that, the field frequently looks towards biological inspirations. This thesis investigated the applicability of the GRN model to control problems and found it to exhibit a high degree of evolvability. The computational cost and overall evolvability of artificial GRNs and genotype-like encodings makes them still a less likely choice than better understood optimization methods such as neural networks (evolved or trained) for typical engineering problems, but we are only starting to understand the properties and potential of this type of approaches. It is a matter of time until they find their niches for which their advantages will outweigh higher computational costs. The proposed model of development in 3D controlled by GRN was found to be capable of evolving non trivial morphologies. Such morphologies are encoded in a very indirect way and emerge through an orchestrated effort of genes interacting with each other, cell-to-cell communication and laws of physics embedded in the environment. Although algorithmic generation of an incremental assembly for arbitrary structures consisting of uniform components would be relatively straightforward, it is the indirectness associated with biologically inspired development that makes such approach interesting. It involves compression of information and self organization through local processes and this, as was demonstrated in Chapter 6, can result in structures that exhibit features that were not selected for, such as high failure tolerance.

The other objective of this thesis was to design and create an evolvable computer model of GRN controlled multicellular development that can be used to investigate how genomes and morphologies evolve, thus allowing to perform experiments *in silico* that would be relevant to evolutionary developmental biology (*evo-devo*). Although multiple prior models of embryogenesis exist, this work attempted to create a model

that would be more biologically plausible and would lift some of the limitations found in earlier approaches. In the model introduced in this thesis, the development occurs in 3D. Cells do not occupy discrete positions on a uniform grid and can move freely in a continuous space and interact through a simulated physics. The fully grown 3D embryo is a result of processes that occur at multiple levels of abstraction. At the lowest level, TFs interact with regulatory regions on the artificial DNA. The network of their interactions is represented as a graph of regulation, i.e., GRN. GRN controls the action of each cell. Each cell takes an independent decision whether to divide, die or reorient its axis of division or whether to emit morphogens which can be sensed by other cells. Cells interact in a simulated physical environment and form a 3D structure thanks to the adhesive forces between them. Finally, at the highest level of evolution, genomes are mutated using biologically plausible mutation operators and their populations evolve guided by a fitness function.

The introductory part of this thesis consisted of an overview of the essential biological concepts this work attempted to model and abstract. It focused on the encoding of genetic information in living organisms and the mechanisms of gene regulation, captured by the concept of the GRN. It also provided a brief introduction to the recent discoveries of the evolutionary developmental biology that shed light on how genes control the multicellular development and how small changes to the genome can result in large, but organized changes in the morphology. Chapter 2 provided an overview of other existing models of GRNs and of multicellular development on which this work builds upon and extends. Chapter 3 introduced the model of the genome and the GRN. The essential feature of the virtual genome is that it is based on an abstraction of biological regulatory and product coding regions and it allows to encode an arbitrary topology of the GRN in a genome which does not have a fixed size and can grow and shrink during evolution.

The ability of the proposed model to evolve GRNs that perform specific tasks was investigated in Chapter 4. GRNs were evolved to produce specific patterns of gene expression over time and to react to stimuli provided as an externally enforced concentration of a product. Networks were evolved to process and respond to information encoded in the concentration of the input signal, its frequency and timing of the stimuli. Investigation of the degree of generalization found that most of the obtained networks generalized the presented problems and avoided overfitting to the training set, a property crucial for any method of automatic controller design. The quality of the solutions evolved using different versions of the fitness functions was investigated and it was demonstrated how the evolvability can be improved with additional terms in the fitness function that compensate for certain biases of the GRN. An attempt to investigate how the evolved solutions function by visualizing the topologies of their GRNs proved difficult, as the networks evolved to be very dense and highly interconnected. Nonetheless, an insight into how the solutions work could be gained by investigating how the networks react to different stimuli. Finally, the networks were shown to exhibit a degree of tolerance to the inherent noise of real gene expression, simulated as a Gaussian noise added to the product concentrations. Functional networks with increased tolerance to high levels of noise

were obtained by evolving them with lower noise levels.

Chapter 5 built on top of the initial attempt to evolve GRNs that can process signals and applied it to the evolution of foraging behaviours of unicellular organisms. The purpose of this chapter was to evaluate GRNs as real time controllers and to create a more biologically plausible scenario of their evolution: organisms trying to survive in a simulated environment. A more complex version of the problem was also investigated, in which networks would have to present different behaviours based on the same signals from sensors. The evolution of network topology was investigated and a reduction of network density over time was observed. These experiments served as a test bed for an open ended evolution system in which unicellular organisms compete for resources and evolve. Such a system is currently being developed and is based on the GRN model proposed in this thesis (Erdei et al., 2011).

Chapter 6 introduced the model of multicellular development, controlled by the model of GRN introduced earlier. During development, each cell would have the same genome and GRN, but would make independent decisions on its behaviour based upon its environment which included detection of maternal gradients and morphogens from other cells as well as based on its current state (concentrations of its TFs). It was demonstrated how, by using an objective fitness function comparing the embryo with an expected shape, embryos with desired morphologies can be evolved. The embryos were investigated using gene knock-out experiments to determine how the morphology is influenced by morphogens and correlations between the loss of a gene and a loss of a morphological feature were observed (also investigated in Joachimczak and Wróbel, 2008a). The embryos were investigated for their robustness to cellular damage during development and were found to present a very high degree of tolerance, capable of regaining most of their fitness even if 50% of the cells were being removed. It was also demonstrated how embryos capable of regrowing their fragments after lesions could be evolved. Section 6.4 demonstrated how the model can be used to evolve morphologies with differentiated cells, by evolving shapes with desired multi-colour patterning. Even without the presence of maternal gradients, the cells were able to self organize and develop patterns using only endogenous morphogens. Furthermore, it was shown how the morphologies are obtained thanks to cells exploiting physics of the system.

The final Chapter 7 proposed a novel method to create an open ended evolution environment from an existing artificial embryogenesis system, with only relatively small changes to the code. Open ended evolution was found to result in a range of morphologies that have much higher apparent complexities than the morphologies achievable in experiments with an objective fitness function (Chapter 6). Evolutionary histories of selected individuals were investigated by retracing their ancestors through evolutionary time. The exploration of the phenotype space was analysed with the use of multidimensional scaling (MDS). The evolution was observed to proceed through relatively small changes to the phenotypes and continuously discovering new areas of the phenotype space. Such properties are very desirable from the point of view of evolvability.

## 8.1 Summary of contributions

This thesis contributes to our understanding of evolvability of artificial GRNs and the range of problems it can be applied to and pushes the boundary of what can be done using highly biologically inspired approach to artificial embryogenesis. It shows how GRNs can be used to evolve circuits that process continuous signals and control simple robots and finds that they display very good generalization properties. It shows the inherent robustness associated with this method of control and demonstrates how evolvability is improved by incorporating the knowledge of some of the biases associated with this model of computation into the fitness function. The thesis introduces a new model of GRN controlled multicellular development that does not need to occur on a grid and happens in a continuous space with cells interacting through simulated physical forces. The model is a compromise between realism and computational tractability and allows to investigate the evolution of development *in silico*, giving full access to the history of evolution and changes that occur in genomes, GRNs and morphologies. It furthers our understanding of development and its inherent features that emerge even if they are not selected for, such as very high robustness to damage and self organization relying on intercellular communication. It shows how the fitness function can be modified to evolve multicellular structures that have desired properties such as self termination of growth or ability to regrow their parts.

Finally, the thesis proposes an open ended system for evolution of 3D morphologies which can be used to explore the space of possible phenotypes in this and other developmental models. It can also be used as a platform to study evolution of development, since it applies continuous evolutionary pressure to evolve new morphologies.

Revisiting the original thesis presented at the beginning of this work (p. 17), the evolvability of the proposed GRN model has been demonstrated on multiple types of problem in chapters 4, 5 and 6. The model of three dimensional development in a grid-less setting was found to display a high degree of evolvability and demonstrated very desirable emergent properties such as robustness to damage (Chapter 6). It was then extended into an open ended system that allows to investigate general properties of evolving genomes and morphologies (Chapter 7).

## 8.2 Future work

The experiments performed in this thesis provide only an introductory overview of what can be achieved using computation inspired by biological genomes, regulatory networks and development. Many directions of further research can be envisioned, both focusing on the practical applications and on relevance for understanding biology.

For example, the experiments in evolving signal processing GRNs (Chapter 4) focused on the evolvability of a single GRN interfacing the external world through special TFs. It would be interesting to investigate whether employing ensembles

of GRNs (multiple cells sharing the GRN or a developing embryo) would have an improved evolvability and would allow to solve more complex tasks. Would the division of labour and specialization among the cells emerge? It remains an open question whether allowing evolution more fine grained control on the properties of TFs, such as their degradation rates (by encoding additional values for every genetic element) is helpful (some GRN models indeed allow that, e.g., Knabe et al., 2006). Would it improve evolvability to explicitly implement higher levels of regulation that exist in cells, such as protein-protein interactions? Furthermore, the current model of TFs binding to the promoters does not involve any competition for the binding sites. Although high concentration of a TF will currently lead to the saturation of its effect owing to the use of a sigmoidal activation function, there is no effect of increased number of the binding sites (which would reduce the effect a given TF per site). This has implications on the effects of gene duplications and it should be investigated how this influences the evolvability of the system.

Hardware implementation of the GRN would allow to increase the speed of evolution by orders of magnitude. In fact, systems allowing for hardware accelerated evolution of neural networks solving problems similar to the ones discussed in Chapter 4 have been built in the past, one example being the CAM-Brain Machine (de Garis and Brain Builder Group, 1999; de Garis et al., 2000). Recent advancements in low powered and highly parallel computing only revived the interest in hardware implementations of neural networks (see, e.g., Indiveri et al., 2011, and references therein) and often the same or similar hardware could be used to simulate GRNs.

On the other hand, the function performed by each node in the regulatory graph could be modified, so that it behaves as a neuron and thus the genome would encode topology of a neural network. Although not biologically plausible, this would allow to compare the evolvability of this method of encoding network topologies with other types of indirect encodings. It was recently demonstrated that networks of adaptive exponential integrate-and-fire neurons (Adex, Brette and Gerstner, 2005) and leaky integrate-and-fire with fixed threshold neurons (Dayan and Abbott, 2001) can be evolved using the discussed genetic encoding and can generate desired output spike trains in response to a specific spike train stimuli (Wróbel et al., 2012a).

A completely opposite direction would be to focus on increasing the biological realism of the GRN model, so that it could be used to automatically design synthetic ("wet") GRNs having certain properties. The interest in creating bacterial cells performing desired functions is currently on the rise (see, e.g., Elowitz and Leibler, 2000; Friedland et al., 2009) and potential therapeutic and bioengineering applications could be revolutionary.

The developmental model introduced in chapter Chapter 6, although one of the very few (if not the only) that is capable of evolving morphologies in 3D without the use of a grid, has many limitations which should be addressed in a later work. For one, the grid-less diffusion model does not conserve mass, and thus provides only simple cell-cell communication, without being biologically plausible. It could be improved by assuming that morphogens diffuse on a grid, but the grid does not have to be uniform and can adapt its resolution depending on the local gradient steepness

(e.g., using octrees). Alternatively, cells can be assumed to transfer morphogens only to their adjacent neighbours determined either by the distance or using some other neighbourhood criteria (e.g., Voronoi neighbours). Furthermore, lifting the current assumption of spherical cell representation would pave a way for a whole new range of phenomena essential for biological development that relies on elastic properties of cells. Simulation of a more complex physics would have higher computational cost but the experience with the current version of the system shows that physics simulation amounts to only a small fraction of the computational costs: it is the need to update the state of the GRN with hundreds of connections and do so for every cell in the system that is the most computationally expensive.

Yet another interesting research direction for the current developmental model would be to improve the capability of the embryos to regrow their parts after lesions. Results obtained in this work are promising and suggest that incorporating more explicit signals of cellular damage would be helpful to improve this ability. Furthermore, the networks evolved with expression noise in Chapter 4 were found to exhibit much higher robustness to noise than their counterparts evolved without noise. Would the developmental process evolved with expression noise result in embryos that are more robust both to noise and cellular damage as a side effect? The first results suggest that it is indeed so (Joachimczak and Wróbel, 2012a).

If the cells were allowed to differentiate into neurons of various properties, the system would allow for evolution of developmental neural networks. Although many methods exist to evolve the structure of a neural net, the common problem of all evolutionary design methods is the issue of scalability. None of the existing methods allows to evolve structures of complexity comparable even with that of a neural structures of a common fly. But we know that structures like the brain are created through a developmental process and we know that the encoding of their structure is very indirect. Nowhere in the genome there is a list of connections between neural cells, for the simple reason of the genome being orders of magnitude smaller in information content than the brains are. Hence, artificial development is looked upon as one of the approaches which, thanks to the efficiency of indirect encoding and modularity, will allow us to automatically design artificial brains of much higher complexities than are currently possible. Thus, an important area of further research would be to investigate the cells' ability to differentiate into neurons and evaluate various possible methods of organizing the connectivity between differentiated neurons.

One of the goals of the current system is to ultimately add another level of realism and allow multicellular structures to move in a simulated environment, as was evaluated for single cells in Chapter 5. The early implementation of such a system in 2D, based on the developmental model presented in Chapter 6, yielded very exciting results in which morphologies and controllers for multicellular soft-bodied robots were successfully evolved. The system was able to discover a wide range of modes of locomotion and morphologies, such as undulating elongated individuals and, most interestingly, individuals that grow primitive, fin-like appendages (Joachimczak et al., 2012; Joachimczak and Wróbel, 2012).

Finally, the open ended system proposed in Chapter 7 was designed to create a more biologically plausible platform for simulating the evolution of artificial genomes and morphologies. It would be interesting to investigate how statistical properties of regulatory networks change during evolution, for example, their degree distributions, modularity or clustering coefficients. Are some network motifs overrepresented? (see, e.g., Alon, 2007) It would be interesting to pursue some of the main *evo-devo* questions, such as whether the changes in morphologies that happened during evolutionary histories occurred mostly due to changes to gene regulation or due to addition of new genetic elements. Are mutations to older genes more detrimental than those to more recent ones? Does the phenomenon of terminal addition occur during evolution of development (better known as the haeckelian principle of ontogeny recapitulating phylogeny)? Finally, an even more realistic system based on the current one can be envisioned to simulate evolution of development in a more alife setting, in which morphologies compete for some limited resource. For example, virtual, multicellular plants could be grown to collect sunlight shining from the top, thus igniting the competition for height, but at the cost of stability and robustness to side forces, such as simulated wind.

Although the speed growth for a single CPU core have abruptly slowed down in recent years, the speed of CPUs continues its exponential growth, though now mostly thanks to symmetric multiprocessing (SMP) performed by the increasing number of cores on a single chip. This trend will increase the relevance of highly parallelisable methods of optimization such as evolutionary computation, as well as the importance of highly parallel methods of computing that are robust to interference and failures of its components. This means an increased demand for more evolvable indirect genotype-phenotype encodings that can scale to larger structures, such as the methods inspired by biological development and regulatory networks described in this work.

# Appendix A

# Software implementation

The software platform used to obtain the results described in this thesis has been developed in C++ with Standard Template Library (STL). The code was written in a cross-platform manner and the platform relies on wxWidgets cross platform toolkit (wxWidgets 2.9.3, 2011). OpenGL was used for rendering of 3D embryos and virtual 2D worlds in which foraging animats evolve.

The main application allows to configure and execute the GA and view its progress. It also allows to investigate any individual, its genome, regulatory network, and developmental process. It also allows to observe gene expressions changing in real time, as well as to visualise expressions in all cells and gradients of morphogens in 3D (Fig. A.1).

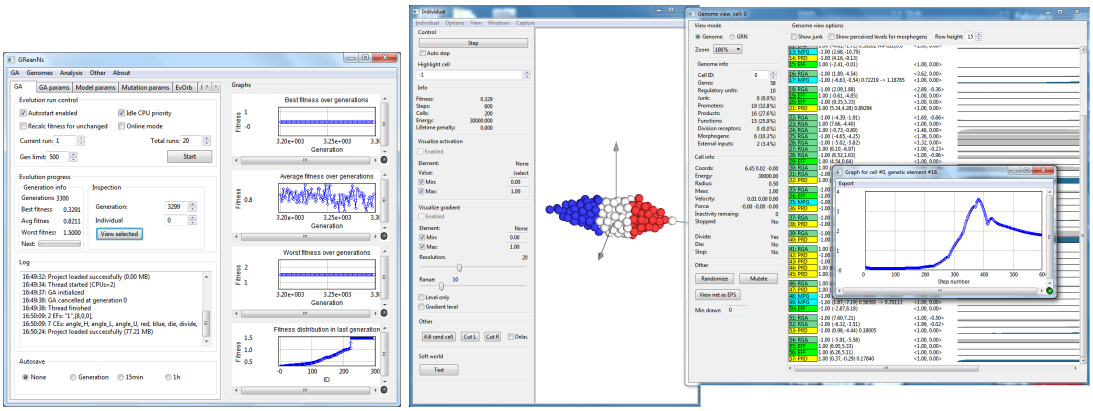


**Figure A.1:** GUI of the software platform designed to perform the experiments discussed in this thesis. Main window is visible, as well 3D view of a developed embryo, genome view and history of activation of a selected regulatory element.

## A.1 Parallelisation

The most CPU intensive part of the simulation (evaluation of the newly created population of genomes) is parallelised to fully exploit the SMP paradigm that is nowadays increasingly common in the form of multi core CPUs. The main thread spawns worker threads in a number equal to detected CPUs. Each thread evaluates

a single genome (computes the fitness). Worker threads that have completed their work request genomes from the pool of genomes that have not been evaluated yet. Furthermore, some of the CPU intensive loops (such as generation of an initial population) rely on OpenMP, a lightweight and convenient thread model that became increasingly available in compilers during the existence of the project.

A console version of the main application (sharing its code, but lacking the GUI) has been prepared for use on computer clusters. It employs MPI to distribute the workload using a master-slave model. The master process stores the current state of the GA and the configuration of the experiment. Whenever a new generation is created, the master distributes the genomes for evaluation among all available slave processes. Because evaluation times can easily vary by at least an order of magnitude (especially for simulated embryogenesis, which can not occur at all or stop prematurely), slave processes request additional work as soon as they have completed their initial assignment. The cost of communication remains very low, because each genome is sent in a binary format in messages that are typically less than 5KB in size. The results described in this thesis have been obtained on the two largest Polish clusters, Galera (Tri-city Academic Computer Centre, TASK) and Halo2 (Interdisciplinary Centre for Molecular and Mathematical Modelling, University of Warsaw).

The main process stores the full history of the evolution in files which can be later analysed. Files generated by the GUI and cluster version of the software are fully interchangeable.

Some of the experiments described in this thesis relied heavily on random number generation. This applied especially to the experiments with noisy gene expression (section 4.6). In these experiments, random number generation becomes a considerable cost. Furthermore, the default random number generator present in the C++ standard library is not thread safe. Synchronizing access to it using locks would be very expensive. Because of these concerns, a much faster algorithm, Mersenne twister (Matsumoto and Nishimura, 1998) was used to generate random numbers with the use of SFMT library (2007). Since this library is also not thread safe, a wrapper class was created. Each thread requests access to random numbers through this class and the class synchronizes their access. To reduce the cost of locking, each thread has a local cache of hundreds of random numbers, and new batch of random number is generated once this cache is exhausted.

## A.2 Analysis

Files storing the history of all individuals existing during an evolutionary run were parsed using scripts written in R, an open source software package for statistical analysis (R 2.13, 2011). Most of the graphs presented in this thesis were prepared in R. iGraph library for R (Csárdi and Nepusz, 2010) was used for drawing regulatory networks and analysing their properties.

# Appendix B

# Algorithms

In this section, the pseudo code for decoding a genome into GRN, simulation of GRN and genome recombination discussed in Chapter 3 is presented.

**Algorithm 1:** The algorithm for decoding a genome into a GRN graph.

**Input**: $N_{in}$,$N_{out}$ the numbers of defined external factors and effectors, respectively
$L$ a list of genetic elements
**Output**: graph $G$ representing the GRN and the array with information about the types of promoters and products associated with every vertex

```
/* STEP 1: locate regulatory units in L                                    */
```
move first $N_{in}$ elements found in $L$ with type *external factor* to a list $L_{inputs}$;
move first $N_{out}$ elements found in $L$ with type *effector* to a list $L_{outputs}$;
remove remaining elements with type *external factor* or *effector* from $L$;
```
/* L contains now only elements belonging to the classes of product or promoter  */
```
remove from $L$ all elements from class product that are before the first element of class promoter;
remove from $L$ all elements from class promoter that are after the last element of class product;
```
/* L consists now only of a series of regulatory blocks                    */
```
**foreach** *group of promoters followed by products in* $L$ **do**
    add a vertex $v$ to graph $G$; associate genetic elements from this group with the vertex $v$;

```
/* STEP 2: determine the connectivity in G                                 */
```
**foreach** vertex $v_{dst}$ in $G$ **do**
    **foreach** promoter $r$ associated with $v_{dst}$ **do**
        **foreach** vertex $v_{src}$ in $G$ **do**
            **foreach** product $p$ associated with $v_{src}$ **do**
                $w$=compute_affinity(r,p);  /* uses Eq. 3.1, p. 66 */
                **if** *w!=0* **then**
                    add an edge to graph $G$ from vertex $v_{src}$ with weight $w$ to vertex $v_{dst}$;
                    store that it connects through promoter $r$;  /* this information is only needed when multiplicative promoters are enabled */

```
/* STEP 3: connect inputs and outputs of G                                 */
```
**foreach** element $i$ in $L_{inputs}$ **do**
    add a vertex $v_{in}$ to graph $G$;  /* create input node */
    mark n-th input vertex as n-th external factor;/* each subsequent input node receives the signal from the subsequent type of external factor defined in the experiment */
    **foreach** vertex $v_{dst}$ in $G$ **do**
        **foreach** promoter $r$ associated with $v_{dst}$ **do**
            $w$=compute_affinity(r,i);
            **if** *w!=0* **then**
                add an edge to graph $G$ from vertex $v_{in}$ with weight $w$ to vertex $v_{dst}$;
                store that it connects through promoter $r$;
**foreach** element $o$ in $L_{outputs}$ **do**
    add a vertex $v_{out}$ to graph; $G$  /* create output node */
    mark n-th output vertex as n-th cellular effector;  /* each subsequent output node corresponds to subsequent type of cellular effector */
    **foreach** vertex $v_{src}$ in $G$ **do**
        **foreach** product $p$ associated with $v_{src}$ **do**
            $w$=compute_affinity(p,o);
            **if** *w!=0* **then**
                add an edge to graph $G$ from vertex $v_{src}$ with weight $w$ to vertex $v_{out}$;

**Algorithm 2:** Pseudo code for updating the state (concentrations) in a GRN with every time step.

**Input**: regulatory graph $G$, information about genetic elements associated with its vertices, edge weights $W[\,]$, concentrations $C[\,]$ associated with each node at the time $t$, time step $dt$
**Output**: updated concentrations $NC[\,]$ for time step $t + 1$

```
/* STEP 1: update the state of the external factors (input nodes)                    */
```
**foreach** input vertex $v_{in}$ in $G$ **do**
    assign the concentration of subsequent external factor to each subsequent vertex $v_{in}$; `/* there are at most as many input nodes as external factor types defined */`

```
/* STEP 2: update the state of all normal nodes in the graph                         */
```
**foreach** non i/o vertex $v_{dst}$ in $G$ **do**
    $a_{add} = 0$; `/* total activation of additive promoters for this vertex */`
    $a_{mul} = 1$; `/* total activation of multiplicative promoters */`
    **foreach** promoter $r$ that belongs to $v_{dst}$ **do**
        $r_a = 0$; `/* activation of the promoter` $r$ `*/`
        **foreach** vertex $v_{src}$ in $G$ that connects to promoter $r$ with the edge $e$ **do**
            $r_a = r_a + C[v_{src}] * W[e]$; `/* add to activation of promoter` $r$`, Eq. 3.3, p. 68 */`
        `/* handle activations of each promoter type separately for Eq. 3.4, p. 68  */`
        **if** is_multiplicative($r$) **then** $a_{mul} = a_{mul} * r_a$ **else** $a_{add} = a_{add} + r_a$;
    $a_{total} = a_{add} * a_{mul}$; `/* total activation of the vertex` $v_{dst}$`, Eq. 3.4, p. 68 */`
    `/* compute the new concentration of` $v_{dst}$ `using Eq. 3.2, p. 68          */`
    $NC[v_{dst}] = C[v_{dst}] + (\tanh(a_{total}/2) - C[v_{dst}]) * dt$;

```
/* STEP 3: update the state of cellular effectors (output nodes)                     */
```
**foreach** output vertex $v_{out}$ in $G$ **do**
    $a = 0$; `/* total activation of this vertex */`
    **foreach** vertex $v_{src}$ in $G$ that connects to $v_{out}$ through edge $e$ **do**
        $a = a + C[v_{src}] * W[e]$;
    `/* compute the new concentration for output vertex using Eq. 3.2, p. 68      */`
    $NC[v_{out}] = C[v_{out}] + (\tanh(a/2) - C[v_{out}]) * dt$

**Algorithm 3:** Pseudo code of the algorithm used to perform a multi point cross over.

**Input**: $P_1$,$P_2$ - arrays of genetic elements in parent 1 and parent 2
function *randomizeAction*() draws a random integer in [0..4] with relative probabilities equal to 5,5,1,1,30, respectively

**Output**: $R$ - a list of genetic elements of the recombined genome

```
c1=0;c2=0;          /* IDs of currently selected genetic elements in P1 and P2 */
a=0;                                          /* current action ID, a ∈ [0..3] */
finished=FALSE;                                                /* termination flag */
repeat
    ra=randomizeAction();          /* randomize current action, 4 means no change */
    if ra<4 then a=ra;
    /* make sure we are not trying to read beyond the genome                     */
    if a in {0,2} and c1 == size(P1) then finished=TRUE;
    if a in {1,3} and c2 == size(P2) then finished=TRUE;
    /* perform currently selected action                                         */
    if finished == FALSE then
        switch a do
            case 0
                R ← P1[c1];          /* insert a single genetic element from P1 to R */
                c1=c1 + 1; c2=c2 + 1;      /* increment current IDs for both parents */
                break;
            case 1
                R ← P2[c2];          /* insert a single genetic element from P2 to R */
                c1=c1 + 1; c2=c2 + 1;      /* increment current IDs for both parents */
                break;
            case 2
                R ← P1[c1];          /* insert a single genetic element from P1 to R */
                c1=c1 + 1;                        /* increment current IDs for P1 only */
                break;
            case 3
                R ← P2[c2];          /* insert a single genetic element from P2 to R */
                c2=c2 + 1;                        /* increment current IDs for P2 only */
until finished == TRUE;
```

**Algorithm 4:** Pseudo code for the calculation of novelty in a population of individuals during open ended evolution of 3D morphology (Chapter 7).

**Input**: population $P$, current state of the archive $A$, novelty threshold of adding individual to the archive $a_t$, the number of individuals $n_r$ added in past 10 generations with novelty above $a_t$

**Output**: novelty value $f_n$ for each individual in $P$, updated archive $A$, $a_t$, $n_r$

```
/* PCA_transform(p) rotates and shifts phenotype p using PCA                    */
/* cmpShapes(p1,p2) calculates distance d_m between phenotypes p1,p2 using Eq. 7.2,
   p. 152                                                                       */
/* findNearest(p,S,k) returns k nearest individuals in set S from individual p  */
/* avgDistance(p,S) returns the average distance of individual p from individuals in
   set S                                                                        */

/* STEP 1: calculate the distance between every phenotype in P and in P ∪ A     */
for i = 0 to size(P) − 1 do /* for each individual in P                        */
    for j = i to size(P) − 1 do /* calculate distance from every other phenotype in P */
        d_m1 = cmpShapes(P[i], P[j]);
        d_m2 = cmpShapes(PCA_transform(P[i]), PCA_transform(P[j]));
        d_population[i][j] = d_population[j][i] = min(d_m1, d_m2);
    for k = 0 to size(A) − 1 do /* calculate distance from every individual in A */
        d_m1 = cmpShapes(P[i], A[k]);
        d_m2 = cmpShapes(PCA_transform(P[i]), PCA_transform(A[k]));
        d_archive[i][k] = min(d_m1, d_m2);

/* STEP 2: calculate novelty using the computed distances, functions findNearest and
   avgDistance rely on distances stored in d_population and d_archive           */
for i = 0 to size(P) − 1 do /* calculate novelty f_n for each individual in P   */
    N = findNearest(P[i], P ∪ A \ P[i], 15);        /* insert 15 nearest into setN */
    f_n[i] = avgDistance(P[i], N);                  /* calculate average distance from N */
    if f_n[i] > a_t then A ← P[i];   /* insert a highly novel individual to the archive A */
    else
        if rand01() > a_t then A ← P[i];  /* insert a random individual to the archive A */
    /* update the count of recently added novel individuals and threshold a_t   */
    n_r = countNovel(A, 10);    /* counts novel inviduals in A from past 10 generations */
    if n_r > 10 then a_t = a_t * 1.2;                /* raise or lower the thrershold */
    else if n_r < 1 then a_t = a_t * 0.95;
```

# Appendix C

# GA settings

This appendix lists detailed settings used to control the GA, which were omitted in the main text.

**Table C.1:** Detailed GA parameters used in the signal processing experiments described in Chapter 4.

| **Settings for genome level mutations (event probabilities per genome)** | |
|---|---|
| Probability of duplication | 0.1 |
| Probability of deletion | 0.2 |
| $p$ for the geometric distribution of length of duplication/deletion | 0.1 |
| ($p$ is the probability of length not being extending by yet another element) | |
| **Settings for mutations at the level of a genetic element (probabilities per element)** | |
| Probability of element type change (see table below for probabilities) | 0.005 |
| Probability of element modifier sign change | 0.005 |
| Probability of element insertion of randomized element | 0.005 |
| Probability of element position change | 0.05 |
| Standard deviation of distance by which genetic element position is changed | 1 |
| **Properties of randomized genomes in the initial population** | |
| Number of regulatory units | 5 |
| Initial locations of element positions | located at the distance drawn from $N(0, 10)$ in random direction from $(0, 0)$ |
| Number of promoters in a randomized regulatory unit | drawn from $N(3, 3)$, with minimum of 1 |
| Number of products in a randomized regulatory unit | drawn from $N(3, 3)$, with minimum of 1 |
| **Relative probabilities for each type of element when the type field is randomized** | |
| Additive promoter | 1 |
| Multiplicative promoter | 1 |
| Transcription factor | 1 |
| External factor | 0.1 |
| Effector | 0.1 |

**Table C.2:** Detailed GA parameters used for evolution of chemotaxis in the experiments described in Chapter 5.

| **Settings for genome level mutations (probabilities per element, unlike in Tab.C.1)** | |
|---|---|
| Probability of duplication | $10^{-4}$ |
| Probability of deletion | $10^{-4}$ |
| $p$ for the geometric distribution of length of duplication/deletion | 0.1 |
| ($p$ is the probability of length not being extending by yet another element) | |
| **Settings for mutations at the level of a genetic element (probabilities per element)** | |
| Probability of element type change (see table below for probabilities) | $5 \cdot 10^{-4}$ |
| Probability of element modifier sign change | $5 \cdot 10^{-4}$ |
| Probability of element insertion of randomized element | 0 |
| Probability of element position change | 0.005 |
| Standard deviation of a distance by which genetic element position is changed | 1 |
| **Properties of randomized genomes in the initial population** | |
| Number of regulatory units | 5 |
| Initial locations of element positions | located at the distance drawn from $N(0, 10)$ in random direction from $(0, 0)$ |
| Number of promoters in a randomized regulatory unit | drawn from $N(3, 3)$, with minimum of 1 |
| Number of products in a randomized regulatory unit | drawn from $N(3, 3)$, with minimum of 1 |
| **Relative probabilities for each type of element when the type field is randomized** | |
| Additive promoter | 1 |
| Multiplicative promoter | 0 |
| Transcription factor | 1 |
| External factor | 0.1 |
| Effector | 0.1 |

**Table C.3:** Parameters of the simulated physics used to simulate multicellular development described in Chapter 6. Description of the physics is provided in section 6.1.2.

| Parameter | Value |
|---|---|
| Default cell diameter | 0.8 |
| Cell mass | 1 |
| Cell repulsion strength coefficient $c_r$ | 5 |
| Cell adhesion strength coefficient $c_a$ | 1 |
| Position of the initial cell | (0,0,0) |
| Fluid drag force coefficient $c_k$ | 5 |
| Physics time per simulation step | 0.05 |

**Table C.4:** Detailed GA parameters used in the experiments on evolving GRNs to control 3D development described in Sections 6.2 and 6.3.

| **Settings for genome level mutations (event probabilities per genome)** | |
|---|---|
| Probability of duplication | 0.02 |
| Probability of deletion | 0.04 |
| $p$ for the geometric distribution of length of duplication/deletion ($p$ is the probability of length not being extending by yet another element) | 0.1 |
| **Settings for mutations at the level of a genetic element (probabilities per element)** | |
| Probability of element type change (see table below for probabilities) | 0.005 |
| Probability of element modifier sign change | 0.005 |
| Probability of element insertion of randomized element | 0 |
| Probability of element position change | 0.005 |
| Standard deviation of a distance by which genetic element position is changed | 1 |
| **Properties of randomized genomes in the initial population** | |
| Number of regulatory units | 5 |
| Initial locations of element positions | uniform distribution over a square area with upper left corner at (-3.5,3.5) and bottom right at (3.5,-3.5) |
| Number of promoters in a randomized regulatory unit | 1 |
| Number of products in a randomized regulatory unit | 1 |
| **Relative probabilities for each type of element when the type field is randomized** | |
| Additive promoter | 1 |
| Multiplicative promoter | 1 |
| Transcription factor | 1 |
| Morphogen | 0.2 |
| External factor | 0.1 |
| Effector | 0.1 |

**Table C.5:** Detailed GA parameters used in the experiments on evolving GRNs to control patterning of 3D embryos described in section 6.4.

| **Settings for genome level mutations (event probabilities per genome)** | |
|---|---|
| Probability of duplication | 0.05 |
| Probability of deletion | 0.05 |
| Duplication/deletion length | position of first and last element chosen randomly |
| **Settings for mutations at the level of a genetic element (probabilities per element)** | |
| Probability of element type change (see table below for probabilities) | 0.005 |
| Probability of element modifier sign change | 0.005 |
| Probability of element insertion of randomized element | 0.005 |
| Probability of element position change | 0.01 |
| Standard deviation of a distance by which genetic element position is changed | 1 |
| Probability of element being repeated during creation of mutated genome | 0.005 |
| Probability of element being lost during creation of mutated genome | 0.01 |
| **Properties of randomized genomes in the initial population** | |
| Number of regulatory units | 5 |
| Initial locations of element positions | located at the distance drawn from $N(0, 10)$ in random direction from $(0, 0)$ |
| Number of promoters in a randomized regulatory unit | drawn from $N(3, 3)$, with minimum of 1 |
| Number of products in a randomized regulatory unit | drawn from $N(3, 3)$, with minimum of 1 |
| **Relative probabilities for each type of element when the type field is randomized** | |
| Additive promoter | 1 |
| Multiplicative promoter | 1 |
| Transcription factor | 1 |
| Morphogen | 0.2 |
| External factor | 0.1 |
| Effector | 0.1 |

**Table C.6:** Detailed GA parameters used in the experiments with open ended evolution of 3D morphologies described in Chapter 7.

| **Settings for genome level mutations (event probabilities per genome)** | |
|---|---|
| Probability of duplication | 0.05 |
| Probability of deletion | 0.05 |
| $p$ for the geometric distribution of length of duplication/deletion ($p$ is the probability of length not being extending by yet another element) | 0.5 |
| **Settings for mutations at the level of a genetic element (probabilities per element)** | |
| Probability of element type change (see table below for probabilities) | 0.005 |
| Probability of element modifier sign change | 0.005 |
| Probability of element insertion of randomized element | 0 |
| Probability of element position change | 0.01 |
| Standard deviation of a distance by which genetic element position is changed | 1 |
| **Properties of randomized genomes in the initial population** | |
| Number of regulatory units | 5 |
| Initial locations of element positions | uniform distribution over a square area with upper left corner at (-3.5,3.5) and bottom right at (3.5,-3.5) |
| Number of promoters in a randomized regulatory unit | 1 |
| Number of products in a randomized regulatory unit | 1 |
| **Relative probabilities for each type of element when the type field is randomized** | |
| Additive promoter | 1 |
| Multiplicative promoter | 1 |
| Transcription factor | 1 |
| Morphogen | 0.1 |
| External factor | 0.1 |
| Effector | 0.1 |

# Bibliography


Adami, C., Brown, C. T., and Kellogg, W. K. (1994). Evolutionary learning in the 2D artificial life system "Avida". In Brooks, R. and Maes, P., editors, *Proceedings of Artificial Life IV: Proceedings of the 4th International Workshop on the Synthesis and Simulation of Living Systems*, pages 377–381, Cambridge, MA. MIT Press.

Alon, U. (2006). *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Chapman and Hall/CRC.

Alon, U. (2007). Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 8(6):450–461.

Andersen, T., Newman, R., and Otter, T. (2009). Shape homeostasis in virtual embryos. *Artificial Life*, 15(2):161–183.

Arkin, A., Ross, J., and McAdams, H. H. (1998). Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected Escherichia coli cells. *Genetics*, 149(4):1633–1648.

Azevedo, R. B. R., Lohaus, R., Srinivasan, S., Dang, K. K., and Burch, C. L. (2006). Sexual reproduction selects for robustness and negative epistasis in artificial gene networks. *Nature*, 440(7080):87–90.

Banzhaf, W. (2003). On the dynamics of an artificial regulatory network. In Goos, G., Hartmanis, J., Leeuwen, J., Banzhaf, W., Ziegler, J., Christaller, T., Dittrich, P., and Kim, J. T., editors, *Advances in Artificial Life: Proceedings of the 7th European Conference on Artificial Life (ECAL 2003)*, volume 2801 of *Lecture Notes in Artificial Intelligence*, pages 217–227, Berlin - Heidelberg. Springer.

Barabási, A.-L. (2009). Scale-free networks: A decade and beyond. *Science*, 325(5939):412–413.

Beckmann, B. E., McKinley, P. K., and Ofria, C. (2007). Evolution of an adaptive sleep response in digital organisms. In *Advances in Artificial Life: Proceedings of the 9th European Conference on Artificial Life (ECAL 2007)*, ECAL'07, pages 233–242, Berlin - Heidelberg. Springer-Verlag.

Beer, R. D. (1995). On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior*, 3(4):469–509.

Bentley, P. J. (2003). Evolving fractal proteins. In Tyrrell, A. M., Haddow, P. C., and Torresen, J., editors, *Proceedings of the 5th International Conference on Evolvable Systems: From Biology to Hardware (ICES 2003)*, volume 2606 of *Lecture Notes in Computer Science*, pages 81–92, Berlin - Heidelberg. Springer.

Bentley, P. J. (2004a). Adaptive fractal gene regulatory networks for robot control. In Miller, J., editor, *Workshop on Regeneration and Learning in Developmental Systems in the Genetic and Evolutionary Computation Conference (GECCO 2004)*.

Bentley, P. J. (2004b). Fractal proteins. *Genetic Programming and Evolvable Machines*, 5(1):71–101.

Berg, J. M., Tymoczko, J. L., and Stryer, L. (2002). *Biochemistry*. W. H. Freeman, New York, NY, USA.

Beurier, G., Michel, F., and Ferber, J. (2006). A morphogenesis model for multiagent embryogeny. In Rocha, L. M., Yaeger, L. S., Bedau, M. A., Floreano, D., Goldstone, R. L., and Vespignani, A., editors, *Artificial Life X: Proceedings of the 10th International Conference on the Simulation and Synthesis of Living Systems*, pages 84–90, Cambridge, MA. MIT Press.

Beyer, H. G. and Schwefel, H. P. (2002). Evolution strategies – comprehensive introduction. *Natural Computing*, 1(1):3–52.

Blake, W. J., Kærn, M., Cantor, C. R., and Collins, J. J. (2003). Noise in eukaryotic gene expression. *Nature*, 422(6932):633–637.

Bolouri, H. (2008). *Computational Modelling Of Gene Regulatory Networks – A Primer*. Imperial College Press.

Bongard, J. (2002). Evolving modular genetic regulatory networks. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2002)*, volume 2, pages 1872–1877. IEEE Press.

Bongard, J. C. and Pfeifer, R. (2003). Evolving complete agents using artificial ontogeny. In Hara, F. and Pfeifer, R., editors, *Morpho-functional Machines: The New Species*, pages 237–258. Springer Japan, Tokyo.

Bonner, J. T. (1998). The origins of multicellularity. *Integrative Biology: Issues, News, and Reviews*, 1(1):27–36.

Bourg, D. M. (2001). *Physics for Game Developers*. O'Reilly Media.

Braitenberg, V. (1986). *Vehicles: Experiments in Synthetic Psychology*. A Bradford Book.

Brette, R. and Gerstner, W. (2005). Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *Journal of Neurophysiology*, 94(5):3637–3642.

Carroll, S., Grenier, J., and Weatherbee, S. (2004). *From DNA to Diversity: Molecular Genetics and the Evolution of Animal Design*. Wiley-Blackwell.

Channon, A. D. and Damper, R. I. (2000). Towards the evolutionary emergence of increasingly complex advantageous behaviours. *International Journal of Systems Science*, 31(7):843–860.

Chavoya, A., Andalon-Garcia, I. R., Lopez-Martin, C., and Meda-Campaña, M. E. (2010). Use of evolved artificial regulatory networks to simulate 3D cell differentiation. *Biosystems*, 102(1):41–48.

Chavoya, A. and Duthen, Y. (2008). A cell pattern generation model based on an extended artificial regulatory network. *Biosystems*, 94(1-2):95–101.

Chou, H.-H. and Reggia, J. A. (1997). Emergence of self-replicating structures in a cellular automata space. *Physica D: Nonlinear Phenomena*, 110(3-4):252–276.

Clune, J., Goldsby, H. J., Ofria, C., and Pennock, R. T. (2010). Selective pressures for accurate altruism targeting: evidence from digital evolution for difficult-to-test aspects of inclusive fitness theory. *Proceedings of the Royal Society B: Biological Sciences*, 278(1706):666–674.

Clune, J., Misevic, D., Ofria, C., Lenski, R. E., Elena, S. F., and Sanjuán, R. (2008). Natural selection fails to optimize mutation rates for long-term adaptation on rugged fitness landscapes. *PLoS Computational Biology*, 4(9):e1000187+.

Cox, T. F. and Cox, M. A. A. (2000). *Multidimensional Scaling*. Chapman and Hall/CRC.

Csárdi, G. and Nepusz, T. (2010). iGraph library 0.5.4. http://igraph.sourceforge.net/.

Davidson, E. H., McClay, D. R., and Hood, L. (2003). Regulatory gene networks and the properties of the developmental process. *Proceedings of the National Academy of Sciences of the United States of America*, 100(4):1475–1480.

Dayan, P. and Abbott, L. F. (2001). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. The MIT Press.

de Garis, H. and Brain Builder Group (1999). Artificial embryology and cellular differentiation. In *Evolutionary Design by Computers*, pages 281–295.

de Garis, H., Korkin, M., Gers, F., Nawa, E., and Hough, M. (2000). Building an artificial brain using an FPGA based CAM-Brain Machine. *Applied Mathematics and Computation*, 111(2-3):163–192.

Eggenberger Hotz, P. (1997). Evolving morphologies of simulated 3D organisms based on differential gene expression. In Husbands, P. and Harvey, I., editors, *Proceedings of the 4th European Conference on Artificial Life (ECAL 1997)*, pages 205–213, Cambridge, MA. MIT Press.

Eggenberger Hotz, P. (2003a). Exploring regenerative mechanisms found in flatworms by artificial evolutionary techniques using genetic regulatory networks. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2003)*, volume 3, pages 2026–2033. IEEE Press.

Eggenberger Hotz, P. (2003b). Genome-physics interaction as a new concept to reduce the number of genetic parameters in artificial evolution. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2003)*, volume 1, pages 191–198. IEEE Press.

Eggenberger Hotz, P. (2004). Asymmetric cell division and its integration with other developmental processes for artificial evolutionary systems. In Pollack, J., Bedau, M. A., Husbands, P., Ikegami, T., and Watson, R. A., editors, *Artificial Life IX: Proceedings of the 9th International Conference on the Simulation and Synthesis of Living Systems*, pages 387–393, Cambridge, MA. MIT Press.

Elowitz, M. B. and Leibler, S. (2000). A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335–338.

Erdei, J., Joachimczak, M., and Wróbel, B. (2011). Ewolucja chemotaksji organizmów jednokomórkowych w dwuwymiarowym środowisku. In Obolewicz, P., Kujawa, K., and Sacharuk, P., editors, *ICT Young 1*, Zeszyty Naukowe Wydziału ETI Politechniki Gdańskiej, pages 173–178. (in Polish).

Evert, R. F. and Eichhorn, S. E. (2004). *Biology of Plants*. W. H. Freeman.

Federici, D. and Ziemke, T. (2006). Why are evolved developing organisms also fault-tolerant? In Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F., Mitchell, J. C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M. Y., Weikum, G., Nolfi, S., Baldassarre, G., Calabretta, R., Hallam, J. C. T., Marocco, D., Meyer, J.-A., Miglino, O., and Parisi, D., editors, *From Animals to Animats 9: Proceedings of the 9th International Conference on Simulation of Adaptive Behaviour (SAB 2006)*, volume 4095 of *Lecture Notes in Computer Science*, pages 449–460, Berlin - Heidelberg. Springer.

Flamm, C., Endler, L., Müller, S., Widder, S., and Schuster, P. (2007). A minimal and self-consistent in silico cell model based on macromolecular interactions. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1486):1831–1839.

Fontana, A. (2008). Epigenetic tracking, a method to generate arbitrary shapes by using evo-devo techniques. In Schlesinger, M., Berthouze, L., and Balkenius, C., editors, *Proceedings of the 8th International Conference on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems (EpiRob 2008)*.

Freeland, S. J., Wu, T., and Keulmann, N. (2003). The case for an error minimizing standard genetic code. *Origins of Life and Evolution of Biospheres*, 33(4):457–477.

Friedland, A. E., Lu, T. K., Wang, X., Shi, D., Church, G., and Collins, J. J. (2009). Synthetic gene networks that count. *Science*, 324(5931):1199–1202.

Galván-López, E. and Poli, R. (2006). An empirical investigation of how and why neutrality affects evolutionary search. In *GECCO '06: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, pages 1149–1156, New York, NY, USA. ACM.

Gershenson, C. (2003). Classification of random boolean networks. In Standish, R. K., Bedau, M. A., and Abbass, H. A., editors, *Artificial Life VIII:Proceedings of the 8th International Conference on Artificial Life*, pages 1–8, Cambridge, MA, USA. MIT Press.

Gerstein, M. B., Bruce, C., Rozowsky, J. S., Zheng, D., Du, J., Korbel, J. O., Emanuelsson, O., Zhang, Z. D., Weissman, S., and Snyder, M. (2007). What is a gene, post-ENCODE? History and updated definition. *Genome Research*, 17(6):669–681.

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361.

Glazier, J. and Graner, F. (1993). Simulation of the differential adhesion driven rearrangement of biological cells. *Physical Review E*, 47(3):2128–2154.

Gower, J. C. (1966). Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53(3/4):325–338.

Gregory, T. R. (2001). Coincidence, coevolution, or causation? DNA content, cell size, and the C-value enigma. *Biological reviews of the Cambridge Philosophical Society*, 76(1):65–101.

Harvey, I. and Bossomaier, T. (1997). Time out of joint: Attractors in asynchronous random boolean networks. In Husbands, P. and Harvey, I., editors, *Proceedings of the 4th European Conference on Artificial Life (ECAL 1997)*, pages 67–75.

Haygood, R., Fedrigo, O., Hanson, B., Yokoyama, K.-D., and Wray, G. A. (2007). Promoter regions of many neural- and nutrition-related genes have experienced positive selection during human evolution. *Nature Genetics*, 39(9):1140–1144.

Hoekstra, H. E. and Coyne, J. A. (2007). The locus of evolution: evo devo and the genetics of adaptation. *Evolution*, 61(5):995–1016.

Hogeweg, P. (1999). Shapes in the shadow: evolutionary dynamics of morphogenesis. *Artificial Life*, 6(1):85–101.

Hogeweg, P. (2000). Evolving mechanisms of morphogenesis: on the interplay between differential adhesion and cell differentiation. *Journal of Theoretical Biology*, 203(4):317–333.

Hornby, G., Lohn, J. D., and Linden, D. S. (2010). Computer-automated evolution of an X-Band antenna for NASA's Space Technology 5 mission. *Evolutionary Computation*, 19(1):1–23.

Hutton, T. J. (2007). Evolvable self-reproducing cells in a two-dimensional artificial chemistry. *Artificial Life*, 13(1):11–30.

Indiveri, G., Linares-Barranco, B., Julia, T., van Schaik, A., Etienne-Cummings, R., Delbruck, T., Liu, S.-C. C., Dudek, P., Häfliger, P., Renaud, S., Schemmel, J., Cauwenberghs, G., Arthur, J., Hynna, K., Folowosele, F., Saighi, S., Serrano-Gotarredona, T., Wijekoon, J., Wang, Y., and Boahen, K. (2011). Neuromorphic silicon neuron circuits. *Frontiers in Neuroscience*, 5.

Jakobi, N. (1995). Harnessing morphogenesis. In Holcombe, M. and Paton, R., editors, *Proceedings of Information Processing in Cells and Tissues*, pages 29–41.

Jędruch, W. T. and Barski, M. (1990). Experiments with a universe for molecular modelling of biological processes. *Biosystems*, 24(2):99–117.

Joachimczak, M., Kowaliw, T., Doursat, R., and Wróbel, B. (2012). Brainless bodies: Controlling the development and behavior of multicellular animats by gene regulation and diffusive signals. In *Artificial Life XIII: Proceedings of the 13th International Conference on the Simulation and Synthesis of Living Systems*, Cambridge, MA. MIT Press. (in press).

Joachimczak, M. and Wróbel, B. (2008a). Evo-devo *in silico*: a model of a gene network regulating multicellular development in 3D space with artificial physics. In Bullock, S., Noble, J., Watson, R., and Bedau, M. A., editors, *Artificial Life XI: Proceedings of the 11th International Conference on the Simulation and Synthesis of Living Systems*, pages 297–304, Cambridge, MA. MIT Press.

Joachimczak, M. and Wróbel, B. (2008b). Evolution of 3D development controlled by a gene regulatory network: The complexity of the search space and evolvability. In Klemm, K., Merkle, D., and Olbrich, E., editors, *8th German Workshop on Artificial Life: Proceedings of the GWAL-8, Leipzig, Germany*, pages 11–22, US. IOS Press.

Joachimczak, M. and Wróbel, B. (2009). Complexity of the search space in a model of artificial evolution of gene regulatory networks controlling 3D multicellular morphogenesis. *Advances in Complex Systems*, 12(3):347–369.

Joachimczak, M. and Wróbel, B. (2010a). Evolving gene regulatory networks for real time control of foraging behaviours. In Fellermann, H., Dörr, M., Hanczyc, M. M., Laursen, L. L., Maurer, S., Merkle, D., Monnard, P.-A., Stoy, K., and Rasmussen, S., editors, *Artificial Life XII: Proceedings of the 12th International Conference on the Simulation and Synthesis of Living Systems*, pages 348–355, Cambridge, MA. MIT Press.

Joachimczak, M. and Wróbel, B. (2010b). Processing signals with evolving artificial gene regulatory networks. In Fellermann, H., Dörr, M., Hanczyc, M. M., Laursen, L. L., Maurer, S., Merkle, D., Monnard, P.-A., Stoy, K., and Rasmussen, S., editors, *Artificial Life XII: Proceedings of the 12th International Conference on the Simulation and Synthesis of Living Systems*, pages 203–210, Cambridge, MA. MIT Press.

Joachimczak, M. and Wróbel, B. (2011a). Evolution of the morphology and patterning of artificial embryos: Scaling the tricolour problem to the third dimension. In Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F., Mitchell, J. C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M. Y., Weikum, G., Kampis, G., Karsai, I., and Szathmáry, E., editors, *Advances in Artificial Life. Darwin Meets von Neumann: Proceedings of the 10th European Conference on Artificial Life (ECAL 2009)*, volume 5777 of *Lecture Notes in Computer Science*, pages 35–43, Berlin - Heidelberg. Springer.

Joachimczak, M. and Wróbel, B. (2011b). Ewolucja sieci genowych kontrolujących wirtualne organizmy jedno- oraz wielokomórkowe. In Obolewicz, P., Kujawa, K., and Sacharuk, P., editors, *ICT Young 1*, Zeszyty Naukowe Wydziału ETI Politechniki Gdańskiej, pages 179–184. (in Polish).

Joachimczak, M. and Wróbel, B. (2012). Co-evolution of morphology and control of soft-bodied multicellular animats. In *GECCO '12: Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*. ACM. (in press).

Joachimczak, M. and Wróbel, B. (2012a). Evolution of robustness to damage in artificial 3-dimensional development. *Biosystems*. (in press).

Joachimczak, M. and Wróbel, B. (2012b). Open ended evolution of 3D multicellular development controlled by gene regulatory networks. In *Artificial Life XIII: Proceedings of the 13th International Conference on the Simulation and Synthesis of Living Systems*, Cambridge, MA. MIT Press. (in press).

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22(3):437–467.

Kauffman, S. A. (1993). *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, USA.

Knabe, J. F., Nehaniv, C. L., and Schilstra, M. J. (2008a). Do motifs reflect evolved function?—no convergent evolution of genetic regulatory network subgraph topologies. *Biosystems*, 94(1-2):68–74.

Knabe, J. F., Nehaniv, C. L., and Schilstra, M. J. (2008b). Evolution and morphogenesis of differentiated multicellular organisms: autonomously generated diffusion gradients for positional information. In Bullock, S., Noble, J., Watson, R., and Bedau, M. A., editors, *Artificial Life XI: Proceedings of the 11th International*

*Conference on the Simulation and Synthesis of Living Systems*, pages 321–328, Cambridge, MA. MIT Press.

Knabe, J. F., Nehaniv, C. L., Schilstra, M. J., and Quick, T. (2006). Evolving biological clocks using genetic regulatory networks. In Rocha, L. M., Yaeger, L. S., Bedau, M. A., Floreano, D., Goldstone, R. L., and Vespignani, A., editors, *Artificial Life X: Proceedings of the 10th International Conference on the Simulation and Synthesis of Living Systems*, pages 15–21. MIT Press/Bradford Books.

Komosinski, M. and Ulatowski, S. (1999). Framsticks: towards a simulation of a nature-like world, creatures and evolution. In Nicoud, J.-D., Floreano, D., and Mondada, F., editors, *Proceedings of 5th European Conference on Artificial Life (ECAL 1999)*, volume 1674 of *Lecture Notes in Artificial Intelligence*, pages 261–265. Springer-Verlag.

Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. A Bradford Book.

Kumar, S. and Bentley, P. J. (2003). Biologically inspired evolutionary development. In Tyrrell, A. M., Haddow, P. C., and Torresen, J., editors, *Proceedings of the 5th International Conference on Evolvable Systems: From Biology to Hardware (ICES 2003)*, volume 2606 of *Lecture Notes in Computer Science*, pages 57–68, Berlin - Heidelberg. Springer.

Kuo, P. D., Leier, A., and Banzhaf, W. (2004). Evolving dynamics in an artificial regulatory network model. In Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F., Mitchell, J. C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M. Y., Weikum, G., Yao, X., Burke, E. K., Lozano, J. A., Smith, J., Merelo-Guervós, J. J., Bullinaria, J. A., Rowe, J. E., Tiňo, P., Kabán, A., and Schwefel, H.-P., editors, *Parallel Problem Solving from Nature - PPSN VIII*, volume 3242 of *Lecture Notes in Computer Science*, pages 571–580, Berlin - Heidelberg. Springer.

Leclerc, R. D. (2008). Survival of the sparsest: robust gene networks are parsimonious. *Molecular Systems Biology*, 4(1).

Lehman, J. and Stanley, K. O. (2008). Exploiting open-endedness to solve problems through the search for novelty. In Bullock, S., Noble, J., Watson, R., and Bedau, M. A., editors, *ALIFE XI: Proceedings of the 11th International Conference on Artificial Life*, pages 329–336, Cambridge, MA. MIT Press.

Lehman, J. and Stanley, K. O. (2011). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223.

Lindenmayer, A. (1968). Mathematical models for cellular interaction in development: Parts I and II. *Journal of Theoretical Biology*, 18:280–315.

Lutz, B., Lu, H. C., Eichele, G., Miller, D., and Kaufman, T. C. (1996). Rescue of Drosophila labial null mutant by the chicken ortholog Hoxb-1 demonstrates that

the function of Hox genes is phylogenetically conserved. *Genes & Development*, 10(2):176–184.

Maheshri, N. and O'Shea, E. K. (2007). Living with noisy genes: how cells function reliably with inherent variability in gene expression. *Annual review of biophysics and biomolecular structure*, 36(1):413–434.

Mahfoud, S. W. (1995). *Niching Methods for Genetic Algorithms*. PhD thesis, University of Illinois at Urbana-Champaign.

Matsumoto, M. and Nishimura, T. (1998). Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, 8(1):3–30.

McClelland, J. L. and Rumelhart, D. E. (1988). *Explorations in parallel distributed processing: a handbook of models, programs, and exercises*. MIT Press.

Miller, J. F. (2004). Evolving a self-repairing, self-regulating, french flag organism. In Deb, K., editor, *GECCO '04: Proceedings of the 6th Annual Conference on Genetic and Evolutionary Computation*, volume 3102 of *LNCS*, pages 129–139, Berlin - Heidelberg. Springer.

Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2002). Network motifs: Simple building blocks of complex networks. *Science*, 298(5594):824–827.

Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. A Bradford Book.

Mondada, F., Franzi, E., and Guignard, A. (1999). The Development of Khepera. In *Experiments with the Mini-Robot Khepera, Proceedings of the First International Khepera Workshop*, HNI-Verlagsschriftenreihe, Heinz Nixdorf Institut, pages 7–14.

Munsky, B., Neuert, G., and van Oudenaarden, A. (2012). Using gene expression noise to understand gene regulation. *Science*, 336(6078):183–187.

Mutsuo Saito, M. M. and University, H. (2007). SIMD-oriented Fast Mersenne Twister. http://www.math.sci.hiroshima-u.ac.jp/˜m-mat/MT/SFMT/.

Nicolau, M. and Schoenauer, M. (2009). On the evolution of scale-free topologies with a gene regulatory network model. *Biosystems*, 98(3):137–148.

Nicolau, M., Schoenauer, M., and Banzhaf, W. (2010). Evolving genes to balance a pole. In Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F., Mitchell, J. C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M. Y., Weikum, G., Esparcia-Alcázar, A. I., Ekárt, A., Silva, S., Dignum, S., and Şima Uyar, A., editors, *EuroGP: 13th European Conference on Genetic Programming*, volume 6021 of *Lecture Notes in Computer Science*, pages 196–207, Berlin - Heidelberg. Springer.

Pellicer, J., Fay, M. F., and Leitch, I. J. (2010). The largest eukaryotic genome of them all? *Botanical Journal of the Linnean Society*, 164(1):10–15.

Prusinkiewicz, P. and Lindenmayer, A. (1996). *The algorithmic beauty of plants*. Springer-Verlag New York, Inc., New York, NY, USA.

Quayle, A. P. and Bullock, S. (2006). Modelling the evolution of genetic regulatory networks. *Journal of Theoretical Biology*, 238(4):737–753.

Quick, T., Nehaniv, C. L., Dautenhahn, K., and Roberts, G. (2003). Evolving embodied genetic regulatory network-driven control systems. In Banzhaf, W., Christaller, T., Dittrich, P., Kim, J. T., and Ziegler, J., editors, *Advances in Artificial Life: Proceedings of the 7th European Conference on Artificial Life (ECAL 2003)*, pages 266–277.

R 2.13 (2011). http://www.r-project.org/.

Ratcliff, W. C., Denison, R. F., Borrello, M., and Travisano, M. (2012). Experimental evolution of multicellularity. *Proceedings of the National Academy of Sciences of the United States of America*, 109(5):1595–1600.

Ray, T. S. (1992). Evolution and optimization of digital organisms. In Billingsley, K. R., Brown, H. U., and Derohanes, E., editors, *Scientific Excellence in Supercomputing: The 1990 IBM Contest Prize Papers*, pages 489–531, The University of Georgia. The Baldwin Press.

Reil, T. (1999). Dynamics of gene expression in an artificial genome - implications for biological and artificial ontogeny. In Floreano, D., Nicoud, J.-D., and Mondada, F., editors, *Advances in Artificial Life: Proceedings of the 5th European Conference on Artificial Life (ECAL 1999)*, volume 1674 of *Lecture Notes In Computer Science*, pages 457–466, London, UK. Springer-Verlag.

Sareni, B. and Krahenbuhl, L. (1998). Fitness sharing and niching methods revisited. *IEEE Transactions on Evolutionary Computation*, 2(3):97–106.

Schramm, L., Jin, Y., and Sendhoff, B. (2011). Emerged coupling of motor control and morphological development in evolution of multi-cellular animats. In Kampis, G., Karsai, I., and Szathmáry, E., editors, *Advances in Artificial Life. Darwin Meets von Neumann: Proceedings of the 10th European Conference on Artificial Life (ECAL 2009)*, volume 5777 of *Lecture Notes in Computer Science*, pages 27–34, Berlin - Heidelberg. Springer.

Schramm, L., Martins, V. V., Jin, Y., and Sendhoff, B. (2010). Analysis of gene regulatory network motifs in evolutionary development of multicellular organisms. In Fellermann, H., Dörr, M., Hanczyc, M. M., Laursen, L. L., Maurer, S., Merkle, D., Monnard, P.-A., Stoy, K., and Rasmussen, S., editors, *Artificial Life XII: Proceedings of the 12th International Conference on the Simulation and Synthesis of Living Systems*, pages 133–140, Cambridge, MA. MIT Press.

Schramm, L. and Sendhoff, B. (2011). An animat's cell doctrine. In Lenaerts, T., Giacobini, M., Bersini, H., Bourgine, P., Dorigo, M., and Doursat, R., editors, *ECAL 2011: Proceedings of the 11th European Conference on the Synthesis and Simulation of Living Systems*, pages 739–746, Cambridge, MA. MIT Press.

Secretan, J. and Beato, N. (2008). Picbreeder: evolving pictures collaboratively online. In *CHI '08: Proceedings of the 26th Annual SIGCHI Conference on Human Factors in Computing Systems*, pages 1759–1768, New York, NY, USA. ACM.

Serra, R., Villani, M., and Agostini, L. (2004). On the dynamics of random boolean networks with scale-free outgoing connections. *Physica A: Statistical Mechanics and its Applications*, 339(3-4):665–673.

Shipman, R., Shackleton, M., and Harvey, I. (2000). The use of neutral genotype-phenotype mappings for improved evolutionary search. *BT Technology Journal*, 18(4):103–111.

Shmulevich, I., Kauffman, S. A., and Aldana, M. (2005). Eukaryotic cells are dynamically ordered or critical but not chaotic. *Proceedings of the National Academy of Sciences of the United States of America*, 102(38):13439–13444.

Sienkiewicz, R. and Jędruch, W. (2011). The universal constructor in the DigiHive environment. In Kampis, G., Karsai, I., and Szathmáry, E., editors, *Advances in Artificial Life. Darwin Meets von Neumann: Proceedings of the 10th European Conference on Artificial Life (ECAL 2009)*, volume 5778 of *Lecture Notes in Computer Science*, pages 183–190, Berlin - Heidelberg. Springer.

Sims, K. (1994). Evolving virtual creatures. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '94, pages 15–22, New York, NY, USA. ACM Press.

Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127.

Stanley, K. O. and Miikkulainen, R. (2003). A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130.

Streichert, F., Spieth, C., Ulmer, H., and Zell, A. (2003). Evolving the ability of limited growth and self-repair for artificial embryos. In Banzhaf, W., Christaller, T., Dittrich, P., Kim, J. T., and Ziegler, J., editors, *Advances in Artificial Life: Proceedings of the 7th European Conference on Artificial Life (ECAL 2003)*, volume 2801, pages 289–298.

Stumpf, M. P. H., Wiuf, C., and May, R. M. (2005). Subnets of scale-free networks are not scale-free: Sampling properties of networks. *Proceedings of the National Academy of Sciences of the United States of America*, 102(12):4221–4224.

Taylor, T. (2004). A genetic regulatory network-inspired real-time controller for a group of underwater robots. In Groen, F., Amato, N., Bonarini, A., Yoshida, E., and Kröse, B., editors, *Proceedings of the 8th Conference on Intelligent Autonomous Systems (IAS-8)*, pages 403–412. IOS Press.

Tjian, R. (1995). Molecular machines that control genes. *Scientific American*, 272(2):54–61.

Tlusty, T. (2007). A model for the emergence of the genetic code as a transition in a noisy information channel. *Journal of Theoretical Biology*, 249(2):331–342.

Trefzer, M. A., Kuyucu, T., Miller, J. F., and Tyrrell, A. M. (2010). Image compression of natural images using artificial gene regulatory networks. In *GECCO '10: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, GECCO '10, pages 595–602, New York, NY, USA. ACM.

Tufte, G. (2008). Phenotypic, developmental and computational resources: scaling in artificial development. In *GECCO '08: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, pages 859–866, New York, NY, USA. ACM.

Turing, A. M. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 237(641):37–72.

van Noort, V., Snel, B., and Huynen, M. A. (2004). The yeast coexpression network has a small-world, scale-free architecture and can be explained by a simple model. *EMBO reports*, 5(3):280–284.

Wagner, A. (2005). *Robustness and Evolvability in Living Systems (Princeton Studies in Complexity)*. Princeton University Press.

West-Eberhard, M. J. (2003). *Developmental Plasticity and Evolution*. Oxford University Press, USA.

Wolpert, D. and Macready, W. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82.

Wolpert, L. (1968). The French Flag problem: A contribution to the discussion on pattern development and regulation. In Waddington, C. H., editor, *The Origin of Life: Toward a Theoretical Biology*, pages 125–133.

Woolley, B. G. and Stanley, K. O. (2011). On the deleterious effects of a priori objectives on evolution and representation. In *GECCO '11: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, pages 957–964, New York, NY, USA. ACM.

Wróbel, B., Abdelmotaleb, A., and Joachimczak, M. (2012a). Evolving spiking neural networks in the GReaNs (Gene Regulatory evolving artificial Networks) plaftorm. In *EvoNet2012: Evolving Networks, from Systems/Synthetic Biology to Computational Neuroscience (Artificial Life XIII Workshop)*. (in press).

Wróbel, B., Joachimczak, M., Montebelli, A., and Lowe, R. (2012b). The search for beauty: Evolution of minimal cognition in an animat controlled by a gene regulatory network and powered by a metabolic system. In *Proceedings of the 12th International Conference on Simulation of Adaptive Behaviour, From Animals to Animats 12 (SAB'12)*, Lecture Notes in Artificial Intelligence. Springer-Verlag. (in press).

Wuchty, S. (2001). Scale-free behavior in protein domain networks. *Molecular Biology and Evolution*, 18(9):1694–1702.

wxWidgets 2.9.3 (2011). http://www.wxwidgets.org/.

Yaeger, L. (1993). Computational genetics, physiology, metabolism, neural systems, learning, vision, and behavior or polyworld: Life in a new context. In Langton, C. G., editor, *Artificial Life III, Vol. XVII of SFI Studies in the Sciences of Complexity, Santa Fe Institute*, pages 263–298. Addison-Wesley.