

Genetic Mappings in Artificial Genomes

Maciej Komosinski

Szymon Ulatowski

Institute of Computing Science
Poznan University of Technology
Piotrowo 2, 60-965 Poznan, Poland
maciej.komosinski@cs.put.poznan.pl
fax: +48 61 8771525

Abstract

This paper concerns processing of genomes of artificial (computer-simulated) organisms. Of special interest is the process of translation of genotypes into phenotypes, and utilizing the mapping information obtained during such translation. If there exists more than one genetic encoding in a single artificial life model, then the translation may also occur between different encodings. The obtained mapping information allows to present genes-phenes relationships visually and interactively to a person, in order to increase understanding of the genotype-to-phenotype translation process and genetic encoding properties. As the mapping associates parts of the source sequence with the translated destination, it may be also used to trace genes, phenes, and their relationships during simulated evolution.

A mappings composition procedure is formally described, and a simple method of visual mapping presentation is established. Finally, advanced visualizations of gene-phenes relationships are demonstrated as practical examples of introduced techniques. These visualizations concern genotypes expressed in various encodings, including an encoding which exhibits polygenic and pleiotropic properties.

Keywords: genetic mapping, translation, simulation, visualization, genotype, phenotype

1 Introduction

Creating models of real organisms serves various purposes. Such artificial models may focus on individual organisms, genetics, environment, species, populations, fitness, etc. In the fields of evolutionary computation and optimization, an individual organism corresponds to a single solution, and a genotype is often equivalent to a phenotype. Thus it is very important to design efficient representations of individuals, and appropriate genetic operators which modify these representations (see for example [3]). The representation (called also *encoding*) is significant because it determines the space of all possible solutions, and influences the genotype-phenotype-fitness relationship. Therefore it immediately concerns the effectiveness of simulated evolutionary process, and the quality of obtained best solutions.

In the simplest case, the representation reflects directly all parts of a simulated individual. But even in this case, there are often a few possibilities to choose from. Apart from those, there is a wide range of choices of more complex representations, required particularly for complex tasks where solutions (phenotypes) cannot be directly encoded into genotypes. In

The final version of this paper appeared in *Theory in Biosciences* 123(2):125–137, 2004. <http://dx.doi.org/10.1016/j.thbio.2004.04.002>

nature, the situation is different – there is a single, very complex genetic representation: the DNA sequence (see [19] for a comparison of artificial and real organisms and their genetics).

In the field of Artificial Life, many evolutionary computation models are used in a more realistic (biologically plausible) setting. The freedom in designing and using various genetic encodings allows researchers to compare them and study their properties in a single simulation model [28, 6, 15, 20]. If many genetic encodings exist in a single simulation model, then each of them must be capable of expressing phenotypes in that model. This situation raises the problem of *mapping* genes into phenes (mapping of genotype parts responsible for creation of phenotype parts). Such a mapping provides answers to two complementary kinds of questions:

- which genes influence (or are responsible for creating/growing) a given set of phenes?
- what phenes were influenced (created) by a given set of genes?

Artificial Life offers yet another possibility: not only multiple genetic encodings can be designed, but they can form a *hierarchy*, so that a higher-level encoding has different properties than a lower-level one. The simplest case is when a genome in some encoding $f1$ is translated into a sequence in another encoding $f0$, which is then transformed into a phenotype. Thus we have two mappings, each of them providing information about a relation between parts of source and destination encodings. To answer questions about the $f1 \leftrightarrow$ phenotype relation, the two mappings must be appropriately composed to propagate the information they hold. In the general case, this hierarchy of encodings may be much more complex, for example it may be similar to the existence of many stages of the DNA-to-phenotype translation process. A similar mapping approach in biology is an alignment of proteins to nucleotide sequences (i.e. two “states” in the information processing chain), using for example BLAST tools [2].

The mapping information can be primarily used for two purposes:

- tracing genes in genotypes [27], phenes in individual organisms, and gene-phenes relationships during evolution. This is important for analyses of evolutionary dynamics and identifying changes in gene pools, species and populations in subsequent generations [4],
- providing (visual) information about the organism genes-phenes relationship for human researchers and users. This increases understanding of mechanisms behind a genetic encoding, reveals its properties, and supports humans in manually designing (editing) artificial genotypes, in improving them and in fixing problems on the genetic level.

1.1 Genetics in Artificial Life systems

The genetic mapping in artificial life models is well known, but it is yet to be explored and understood in natural organisms, where genomes are much bigger and more complex, and many genes are epistatic. Another feature typical to most artificial life systems is the characteristics of genotype-phenotype mappings, which causes genomes to often vary greatly in size, depending on the phenotype size (number of phenes).

A large amount of work has been carried out to design good genetic representations, mainly in terms of fitness of evolved phenotypes. The range of these representations is wide, and includes approaches dedicated for optimization of control systems (brains) alone (which is traditionally the domain of Artificial Intelligence) [16, 22, 14, 17, 7, 10], designs (bodies, morphologies) alone [13, 5, 9, 12, 23, 25], and both brains and bodies at the same time [26, 11, 8, 1, 24, 20, 21]. However, comparing such representations is difficult because not only genetic encodings are different, but also the simulation models vary. When the encodings concern different phenetic models, comparing them is virtually impossible. Another aspect is that the comparison itself can be performed using many criteria, like fitness values, qualitative nature of solutions, their simplicity, understandability, etc.

Some researchers tried to systematize and compare artificial genetic encodings in a single phenetic model [28, 6, 20, 15]. This allows to test their performance in the same environment, and thus focus on the characteristics of the encodings (features like their complexity,

constraints, support for modularity, body symmetry, compression, redundancy, etc.). In this paper, many genetic encodings also concern a single phenetic model. The Framsticks model [18] is presented as an example, although ideas expressed here are general and can be applied to most artificial life genetics models.

If there exist many genetic encodings and translation procedures, then the corresponding graph may be constructed. Fig. 1 presents such a sample mapping graph, where genetic encodings constitute nodes, and directed arcs are translation (mapping) procedures. There is a difference between terms *translation* and *mapping*. Translation is the process of conversion of a source encoding into a destination encoding (the argument of the translation procedure is always a specific sequence). A mapping is the information about the relationships between parts of source and destination sequences, or it is the process of obtaining/providing this information during translation. In the sample graph shown in Fig. 1, f_0 is a base encoding which directly describes all parts of a phenotype (thus there is an explicit mapping between f_0 and its phenotype). f_1 and f_4 are two higher-level encodings. The dashed arc between f_4 and f_1 represents a possibility of approximate translation (for example, a translation which loses some information). f_{Test} is another encoding which can be translated into f_1 .

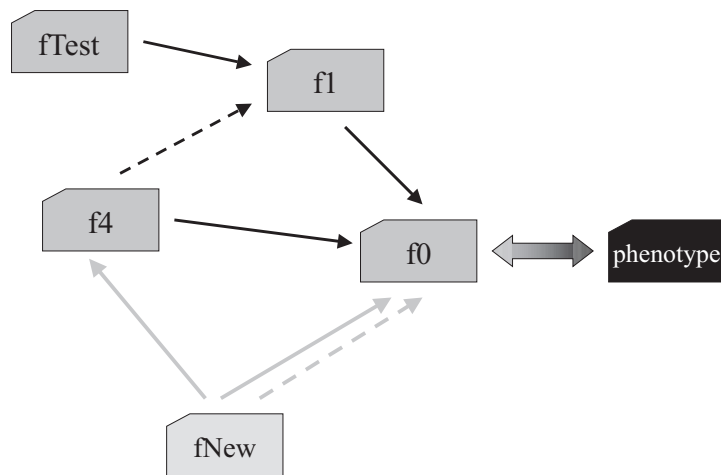


Figure 1: A sample genetic encoding translation (mapping) graph.

The f_{New} encoding illustrates a situation of adding a new encoding to the mapping graph. The f_{New} designer might for example provide not only two translation procedures (to f_4 and f_0), but also two alternative ways of translating f_{New} to f_0 , one of them approximate (shown as a dashed line). The f_{New} encoding might for example define genotypes as sequences composed of the ATGC letters.

Note that if there exist multiple paths from the source encoding to the destination, then the resulting destination sequences may differ for each path.

2 Genetic mappings

A mapping describes dependencies between individual parts of the source and destination sequences¹. For clarity and simplicity of presentation, we assume that a basic part of a sequence is a single character. Thus the source sequence X and destination sequence Y are strings of numbered characters (signs). The string lengths are m and n , respectively. The

¹Some encodings may make the mapping information useless. This is the case when it is impossible to tell which part of the destination sequence originated from any given region in the source genome (e.g. because every character in the source influences the whole destination).

mapping is the relation R on sets $X = \{1, 2, \dots, m\}$ and $Y = \{1, 2, \dots, n\}$, so it is the subset of the Cartesian product $X \times Y$.

$$(x, y) \in R \Leftrightarrow \begin{cases} \text{the destination character at position } y \text{ originates from} \\ \text{(is influenced by) the source character at position } x \end{cases}$$

Consider a simple example: a translation procedure which replaces every digit-char pair in the source sequence (its encoding called $fTest$) into a number of identical characters in the $f1$ encoding (i.e. $a3LX$ into $aLLLX$). A more complex $fTest$ sequence $aX3fXr2X$ is translated into $f1$ sequence $aXfffXrXX$, and for this translation we have

$$\begin{aligned} m &= 8 \\ n &= 9 \\ X &= \{1, 2, 3, 4, 5, 6, 7, 8\} \\ Y &= \{1, 2, 3, 4, 5, 6, 7, 8, 9\} \\ R &= \{(1, 1), (2, 2), (3, 3), (3, 4), (3, 5), (4, 3), (4, 4), \dots\} \end{aligned}$$

or, using the shorter form which groups sequence characters when possible ([source chars]→ [corresponding destination chars]):

$$R = \{[1] \rightarrow [1], [2] \rightarrow [2], [3-4] \rightarrow [3-5], [5] \rightarrow [6], [6] \rightarrow [7], [7-8] \rightarrow [8-9]\}$$

The mapping information can always be used both ways, i.e. to find the subset of Y for the given subset of X , and vice versa. Fig. 2a illustrates the way of presentation of the input-output mapping, where rows correspond to input characters, and columns – to output characters. Fig. 2b depicts the relation R for the above example.

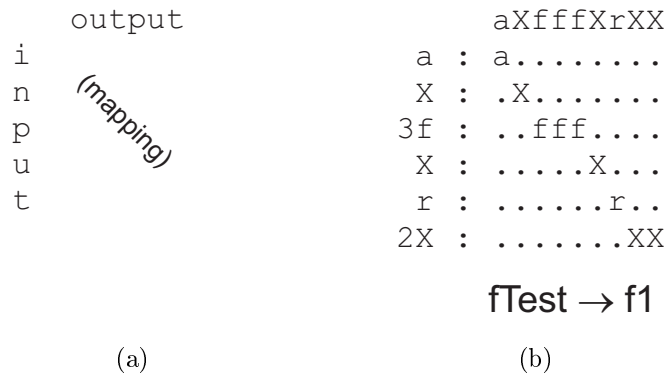


Figure 2: (a) Input-output mapping presentation. (b) Sample mapping.

The translation procedures and their mappings may differ strongly. The mapping $fTest \rightarrow f1$, as defined above, was extremely simple. However, the translation process may be more complex, as it is the case with the $f4 \rightarrow f0$ translation. The $f4$ encoding describes an organism by specifying its developmental process rather than its final form. The encoding models a set of interacting phenotypic parts which execute actions specified by the genotype. Phenotypes are built by passing through a developmental phase. A developing organism consists of a set of interconnected cells, which can be undifferentiated or differentiated. Cells execute genetic codes which alter their properties, or create new cells through division. After a division, the newly created cells execute different codes (they differentiate). Development of a phenotype starts out as a single undifferentiated cell. As new cells are created, they follow their instruction in parallel. Development halts when all cells mature. The $f4$ encoding includes also a repetition gene – see [20] for details.

2.1 Genotype to phenotype mapping

Although the notation introduced so far concerns mapping a sequence into a corresponding sequence expressed in another encoding, it can also be used to map a genotype sequence into its phenotype. The mapping will then hold the information about related genes and phenes – that is, how the organism is constructed.

The phenotype is usually made of a number of objects of different types. Thus the Y set can be treated as the set of these uniquely numbered (labeled) objects. For example, in Framsticks, there are four types of phenotype objects [18, 21]: *parts*, *joints*, *neurons*, and *neural connections* (neural connections are not considered here for simplicity). Fig. 3 presents a mapping of $f1$ genotype $cX[1:1]X[G]$ into the corresponding phenotype which contains 3 parts, 2 joints and 3 neurons.

	Parts	Joints	Neurons
	012345678901234567890123456789	012345678901234567890123456789	0123456789
c	:		
X	: 01.....0.....		
[:0.....		
	:01.....		
1:1]	:0.....		
X	: ..2.....1.....		
[G]	:2.....		

f1 → phenotype

Figure 3: A sample genotype-to-phenotype mapping.

2.2 Composition of mappings

In cases when an encoding is not transformed directly into a phenotype, there is a need of composition of mappings. It is also needed in the previous example (Fig. 3), where we transformed an $f1$ genotype into the phenotype. The intermediate encoding was $f0$ (see also Fig. 1 on page 3). In this case, two mappings are used: $f1 \rightarrow f0$ and $f0 \rightarrow \text{phenotype}$. To obtain the $f1 \rightarrow \text{phenotype}$ mapping, these two mappings must be composed. In general, the composition process concerns all intermediate mappings on the source-destination sequence translation path.

To compose two mappings R_1 and R_2 , we need

- X – source encoding sequence
- Y – intermediate encoding sequence
- Z – destination encoding sequence
- $R_1 \subset X \times Y$
- $R_2 \subset Y \times Z$

The final mapping $R \subset X \times Z$, which is the composition of R_1 and R_2 , is computed as

$$(x, z) \in R \iff \exists_{y \in Y} ((x, y) \in R_1 \wedge (y, z) \in R_2)$$

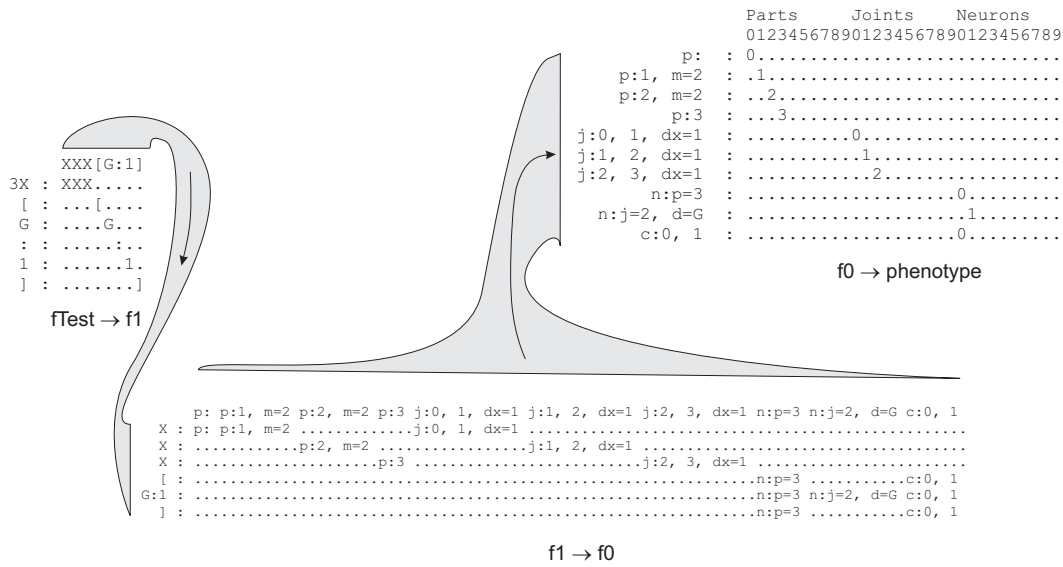


Figure 4: The three mapping steps: $fTest \rightarrow f1$, $f1 \rightarrow f0$, $f0 \rightarrow phenotype$.

3 Illustrative examples

3.1 Composition of mappings

Fig. 4 illustrates three successive steps during transformation of $fTest$ genotype $c3X[G:1]$ into the corresponding phenotype. Note how the $f1$ fragment $[G:1]$ is mapped into the $f0$ encoding.

The composition of these three mappings ($fTest \rightarrow f1$, $f1 \rightarrow f0$, $f0 \rightarrow phenotype$) produces the final result shown in Fig. 5, where the relation between individual $fTest$ genes and phenotype morphology is displayed.

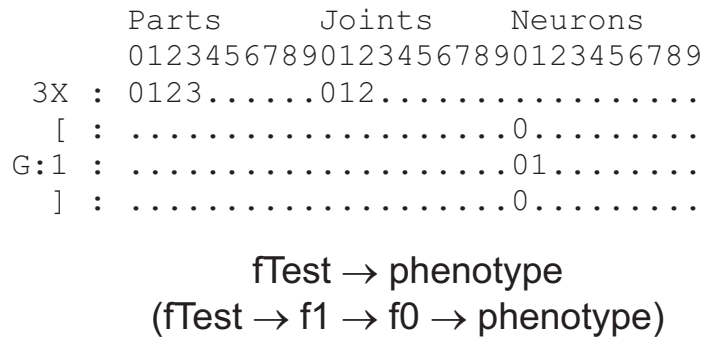


Figure 5: The final $fTest \rightarrow phenotype$ mapping.

3.2 Visualization of genes activity and the genes ↔ phenes relationship

The mechanisms described in previous sections can be applied to visually present the genes ↔ phenes relationship. Fig. 6 on the next page concerns a simple $f1$ genotype. The left picture shows highlighted parts of the phenotype which correspond to the user-selected fragment

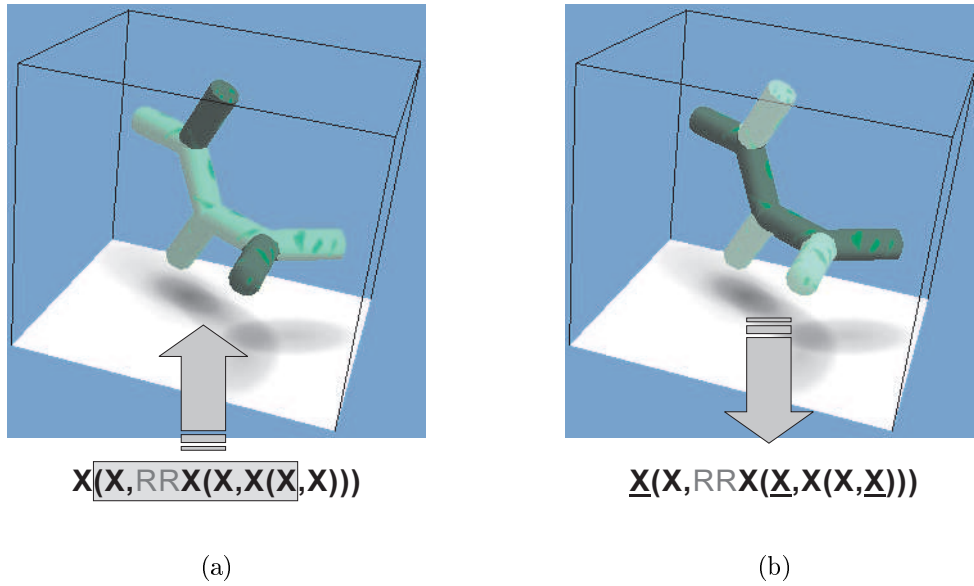


Figure 6: A simple mapping between a $f1$ genotype and the corresponding phenotype. (a) User selected a part of the genotype, corresponding phenes are highlighted. (b) User highlighted some parts of the body, corresponding genes are underlined.

```
(,LFMW(LLX[[ 1 :3.455,0 :1.453]][ 1 :-509.744,-1 :1.033][ @9 :-1] ,,rrX[| :2.834]a(rrX[G:0.827](RLLX[9:-1],
LLiqwX[@G :93.351,-1 :727.177]),rrfM[X[G:1](LLcMX[ @0 :2.095][8:-1], LLfX[7 :1]),rrX[G:1.151,
-1 :1.629]F(LLcX[8:-0.859], LLX[7:1.482]), rrX[G:3.254]F(LLX[9:-2.621], LLX[8 :0.783]),rrX[G:1]F(LLwX
[9:-1],LLX[[1 :1.221,-1 :-3.208]), rrwX[G:1.052](LLFiX[ @ 1 :1.601][ -20:-0.665],LLQX[! :1.000]),
rrX[G:0.757,-1 :-2.644]f(rLLMMX[[-21:-1.377], LLX[[-22 :0.984]), rrQX[G:0.887,-1 :0.750](LLX
[[G:-0.930],LLX[[-22 :2.345]]) ,,LLX[[-19:-4.840,-1 :-0.757]])
```

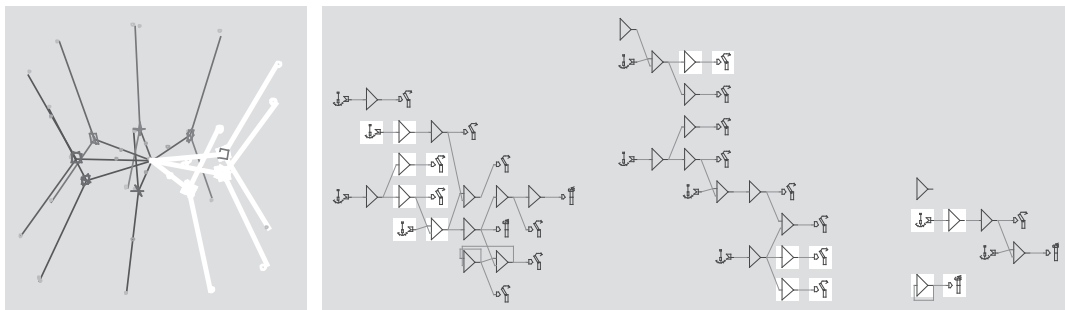


Figure 7: Highlighted body and brain parts for the user-selected genotype fragment.

of the genotype. The right picture shows underlined genes which correspond to the user-selected parts of the phenotype. To achieve such functionality, the two mappings: $f1 \rightarrow f0$ and $f0 \rightarrow$ phenotype must have been computed for the $f1$ genotype $X(X,RRX(X,X(X,X)))$ – and then composed.

Fig. 7 presents a more complex genotype of a creature whose body was designed by a human, and the neural network brain was evolved. Corresponding phenes are highlighted for the user-selected genes.

Fig. 8 on the next page is analogous to Fig. 6, but it concerns a more complex, developmental genetic encoding, $f4$ [20]. To achieve such functionality, the two mappings: $f4 \rightarrow f0$ and $f0 \rightarrow$ phenotype must have been computed for the presented $f4$ genotype, and then composed. Note how a continuous fragment of a genotype influences separate phenes (*pleiotropy*,



Figure 8: A mapping between an f_4 genotype and the corresponding phenotype. Left: user selected a part of the genotype. Right: user selected a part of the body.

on the left picture), and how a single part of a phenotype can be influenced by many distant genes (*polygeny*, on the right picture).

Finally, Fig. 9 on the next page illustrates some interesting situations for the f_4 genotypes, where a single, small genotype fragment influences many distinct parts of the phenotype (*pleiotropy*).

4 Conclusions

This article concerned the mechanisms of processing artificial genomes, including translation between multiple genetic encodings and mapping of related genes between pairs of encodings. The ideas presented in this paper can be applied to most genetics models. Genetic mappings were described in detail, including the formal view, possible applications, and some practical examples. The phenotype, which is the last stage of the translation chain, was also considered. In the presented approach, the phenotype is handled just like each of the genetic encodings, which simplifies both the formal considerations and their application.

A simple method for visual presentation of mappings was introduced, and advanced visualizations of gene-phenotype relationships were shown to illustrate practically translation, mapping, and mappings composition. In a computer software, these operations can be performed interactively. The visual examples shown in Sect. 3.2 are merely static snapshots, therefore the reader is encouraged to download the Framsticks GUI (Graphical User Interface) and experiment with these procedures [21].

In the computer program, it is possible to select parts of the phenotype and genotype to get an instant visual feedback and understand their relationship. The user can also move the cursor along the genotype to see which phenotype parts are influenced by the genotype character under cursor. Another option available is to modify the genotype by adding, deleting, or editing its parts while the corresponding phenotype is computed and displayed. The software can be used to illustrate the phenomena described in this paper; the genotypes used in Sect. 3.2 are available within the Framsticks distribution files². The program source code which implements the mechanisms described in Sect. 2 is available in the SDK package [21].

The ideas presented in this paper allow for direct experiments with artificial genetic encodings, increasing comprehension of the genotype-to-phenotype translation process and properties of genetic encodings – including modularity, compression, redundancy, and many

²The name of the genotype shown in Fig. 7 on the preceding page is “Roller” (file `walking.gen`). The name of the genotype shown in Fig. 8 is “hp_4*.gen”. The names of genotypes shown in Fig. 9 on the following page are “hp_5*.gen”, “hp_8*.gen”, and “h_3*.gen”. All genotypes except for the “Roller” are available in the file `encoding_f4_best.gen`, and they are all results of evolutionary optimization with fitness defined as body height.

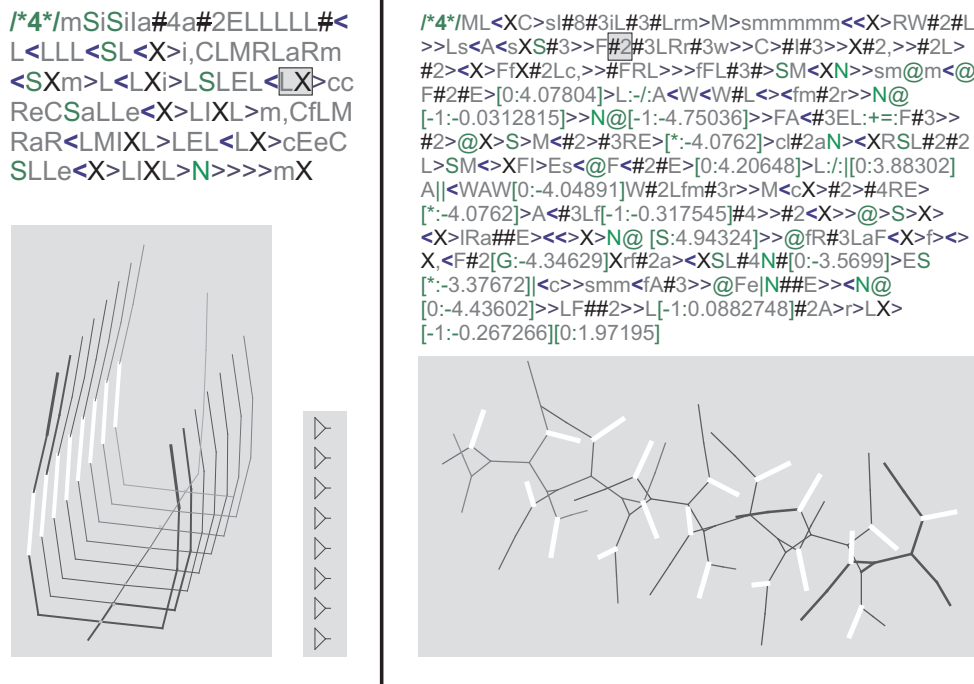


Figure 9: Distinct phenes (highlighted) for a single gene (user-selected): sample pleiotropic effects for the f_4 encoding genotypes.

more. Genetic mappings can also be used together with mechanisms of tracing genes during evolution [27, 4] in order to pursue the evolutionary history of phenes by investigating the genetic history of the corresponding genes.

Acknowledgements

This work has been supported by the State Committee for Scientific Research, from KBN research grant no. 3 T11C 050 26, and by the Foundation for Polish Science, from subsidy no. 38/2004.

References

- [1] John Albert. Computational modeling of an early evolutionary stage of the nervous system. *BioSystems*, 54:77–90, 1999.

- [2] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*, 25(17):3389–3402, September 1 1997.
- [3] K. Balakrishnan and V. Honavar. Evolutionary design of neural architectures: A preliminary taxonomy and guide to literature. Technical report, Department of Computer Science, Iowa State University, Ames, Iowa, 1995.
- [4] Michael Barlow, John Galloway, and Hussein A. Abbass. Mining evolution through visualization. In *ALife VIII: Workshop Proceedings*, pages 103–110, 2002.
- [5] Peter Bentley. *Evolutionary design by computers*. Morgan Kaufmann, 1999.
- [6] Peter Bentley and Sanjeev Kumar. Three ways to grow designs: A comparison of embryogenies for an evolutionary design problem. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 35–43, Orlando, Florida, USA, 1999.
- [7] Alastair Channon. *The evolutionary emergence route to artificial intelligence*. PhD thesis, University of Sussex, School of Cognitive and Computing Sciences, 1996.
- [8] Dave Cliff and Geoffrey F. Miller. Co-evolution of pursuit and evasion II: Simulation methods and results. In P. Maes, editor, *From Animals to Animats IV*, Third European Conference on Artificial Life, pages 506–515. MIT Press, 1996.
- [9] Hugo de Garis. *Genetic programming. GenNets, artificial nervous systems, artificial embryos*. PhD thesis, Brussels University / ATR Human Information Processing Research Laboratories, Brain Builder Group, Evolutionary Systems Department, Kyoto, Japan, 1992.
- [10] Hugo de Garis and Michael Korkin. The CAM-Brain Machine (CBM) : An FPGA based tool for evolving a 75 million neuron artificial brain to control a lifesized kitten robot. *Autonomous Robots Journal, Special Issue on Personal Robotics*, 2000.
- [11] Frank Dellaert and Randall D. Beer. A developmental model for the evolution of complete autonomous agents. In P. Maes, M.J. Mataric, Jean-Arcady Meyer, Jordan B. Pollack, and Stewart W. Wilson, editors, *From Animals to Animats. Proc. of the 4th Int. Conf. on Simulation of Adaptive Behavior SAB '96*, pages 393–401, Cambridge, Mass., 1996. MIT Press.
- [12] Peter Eggenberger. Evolving morphologies of simulated 3D organisms based on differential gene expression. In Phil Husbands and Inman Harvey, editors, *Proceedings of the 4th European Conference on Artificial Life ECAL97, Complex Adaptive Systems Series*. MIT Press, 1997.
- [13] Pablo Funes and Jordan B. Pollack. Evolutionary body building: adaptive physical designs for robots. *Artificial Life*, 4(4):337–357, Autumn 1998.
- [14] Frederic Gruau, Darrell Whitley, and Larry Pyeatt. A comparison between cellular encoding and direct encoding for genetic neural networks. In John R. Koza, David E. Goldberg, David B. Fogel, and Rick R. Riolo, editors, *Proceedings of the First Annual Conference*, Genetic Programming 1996, pages 81–89, Cambridge, MA, 1996. MIT Press.
- [15] Gregory S. Hornby. *Generative Representations for Evolutionary Design Automation*. PhD thesis, Brandeis University Dept. of Computer Science, 2003.
- [16] Nick Jakobi and Matthew Quinn. Some problems (and a few solutions) for open-ended evolutionary robotics. In Phil Husbands and Jean-Arcady Meyer, editors, *Proceedings of the First European Workshop on Evolutionary Robotics: EvoRobot98*. Springer-Verlag, 1998.

- [17] H. Kitano. Designing neural networks using genetic algorithms with graph generation system. *Complex Systems*, 4:461–476, 1990.
- [18] Maciej Komosinski. The Framsticks system: versatile simulator of 3D agents and their evolution. *Kybernetes: The International Journal of Systems & Cybernetics*, 32(1/2; Special Issue on Artificial Life Software):156–173, 2003.
- [19] Maciej Komosinski, Grzegorz Koczyk, and Marek Kubiak. On estimating similarity of artificial and real organisms. *Theory in Biosciences*, 120(3-4):271–286, December 2001.
- [20] Maciej Komosinski and Adam Rotaru-Varga. Comparison of different genotype encodings for simulated 3D agents. *Artificial Life Journal*, 7(4):395–418, Fall 2001.
- [21] Maciej Komosinski and Szymon Ulatowski. Framsticks web site. <http://www.framsticks.com>.
- [22] Ibrahim Kuscu and Chris Thornton. Design of artificial neural networks using genetic algorithms: review and prospect. Technical report, University of Sussex, School of Cognitive and Computing Science, <ftp://ftp.cogs.susx.ac.uk/pub/reports/csrp/csrp319.ps.Z>, 1994.
- [23] Lukas Lichtensteiger and Peter Eggenberger. Evolving the morphology of a compound eye on a robot. In *Proceedings of the Third Workshop on Advanced Mobile Robots (Eurobot '99)*, 6-8 September 1999, 1999.
- [24] Hod Lipson and Jordan B. Pollack. Automatic design and manufacture of robotic lifeforms. *Nature*, 406(6799):974–978, 2000.
- [25] Stephane Sikora, David Steinberg, and Claude Lattaud. Integration of simulation tools in on-line virtual worlds. In Jean-Claude Heudin, editor, *Proceedings of 2nd International Conference on Virtual Worlds (VW2000)*, LNAI 1834, Paris, France, July 2000, pages 32–43. Springer-Verlag, 2000.
- [26] Karl Sims. Evolving 3D morphology and behavior by competition. In Rodney A. Brooks and Pattie Maes, editors, *Proceedings of the 4th International Conference on Artificial Life*, pages 28–39, Boston, MA, 1994. MIT Press.
- [27] Kenneth Stanley and Risto Miikkulainen. Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*, 21:63–100, 2004.
- [28] Kenneth O. Stanley and Risto Miikkulainen. A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130, Spring 2003.