

# An Algorithm to Create Phenotype-Fitness Maps

Jean-Baptiste Mouret<sup>1,2</sup>, Jeff Clune<sup>3</sup>

<sup>1</sup>Université Pierre et Marie Curie-Paris 6, UMR 7222, ISIR, France

<sup>2</sup>CNRS UMR 7222, ISIR, France

<sup>3</sup>Cornell University, Ithaca, USA

mouret@isir.upmc.fr

## Extended Abstract

Understanding the relationships between phenotypic characteristics and fitness is central to evolutionary biology and the design of new evolutionary algorithms (EAs). Whether in computational models of evolution (Kauffman, 1993; Adami, 1998; Lenski et al., 2003) or in evolutionary algorithms (Goldberg, 1989), the common approach is to perform selection based on fitness and study the phenotypes that evolve. Unfortunately, computational evolution tends to be highly convergent, meaning there is little diversity in the population and thus little variation along key phenotypic dimensions. Such a lack of diversity prevents an understanding of how fitness would change along those dimensions ‘had evolution searched there’. The problem is compounded by the fact that fitness landscapes often have many local optima that populations get stuck on, which makes it difficult to know if there are higher fitness peaks in other areas of the fitness landscape that evolution failed to discover. Both biologists and engineers often spend a lot of time asking that very question, and would benefit from tools that help them answer it.

Here we introduce an algorithm to compute *phenotype-fitness maps* as a way to understand the relationship between phenotypic dimensions and fitness. The central idea is to explicitly select for fit organisms in *all areas* of a phenotype landscape, where the axes of that landscape are defined by phenotypic dimensions of interest. To produce such maps, we introduce the Multi-Objective Landscape Exploration (MOLE) algorithm, which is a multi-objective evolutionary algorithm, specifically NSGA-II (Deb, 2001), with two objectives: (1) searching for new organisms that are far from solutions already generated, with distance measured in a Cartesian space defined by the key dimensions, and (2) generating highly fit organisms. With MOLE, scientists can see how fitness changes as a function of various phenotypic dimensions (Figure 1). This combination of a fitness objective and an archive-based exploration objective is similar to “novelty-based multi-objectivization” (Mouret, 2011; Lehman and Stanley, 2011).

We investigate phenotype-fitness maps produced via MOLE by evolving the topology and parameters of feed-forward neural networks to recognize binary patterns in an 8-pixel retina (Kashtan and Alon, 2005). Fitness is the normalized error for all 256 possible input patterns. Two encodings are investigated: a direct encoding that allows for arbitrary, non-recurrent topologies (DNN, see Mouret and Doncieux (2012)), which is similar to NEAT (Stanley and Miikkulainen, 2002), and a more constrained direct encoding that is feedforward and specifies the number of layers and maximum number of neurons per layer (KA, see Kashtan and Alon (2005)). These constraints reduce the search space to a region known to contain perfect solutions. Each phenotype landscape shows the highest-performing organism at each location found during 30 independent evolutionary runs. As a control, we also conducted 30 runs per encoding with a typical EA, represented by NSGA-II with a single fitness objective.

Preliminary results (Figures 1 and 2) show that generated phenotype-fitness maps can provide an informative window into how phenotypic dimensions relate to fitness. Moreover, a single MOLE run can find high-performing organisms with a variety of phenotypic traits instead of the homogenous set typically generated by a single EA run: The quality and diversity of solutions MOLE generates suggests that it could also represent a powerful alternative to classic, convergent EAs.

## References

- Adami, C. (1998). *Introduction to artificial life*, volume 1. Telos Pr.
- Claussen, J. C. (2007). Offdiagonal complexity: A computationally quick complexity measure for graphs and networks. *Physica A: Statistical Mechanics and its Applications*, 375(1):365–373.
- Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Wiley.
- Goldberg, D. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-wesley.
- Kashtan, N. and Alon, U. (2005). Spontaneous evolution of modularity and network motifs. *PNAS*, 102(39):13773–13778.
- Kauffman, S. A. (1993). *The origins of order: Self organization and selection in evolution*. Oxford University Press, USA.

- Kim, J. and Wilhelm, T. (2008). What is a complex graph? *Physica A: Statistical Mechanics and its Applications*, 387(11):2637–2652.
- Lehman, J. and Stanley, K. (2011). Evolving a diversity of creatures through novelty search and local competition. In *Proc. of GECCO*.
- Lenski, R. E., Ofria, C., Pennock, R. T., and Adami, C. (2003). The evolutionary origin of complex features. *Nature*, 423(6936):139–144.
- Li, M. and Vitányi, P. (2008). *An introduction to Kolmogorov complexity and its applications*. Springer-Verlag New York Inc.
- Mouret, J.-B. (2011). Novelty-based multiobjectivization. In *New Horizons in Evolutionary Robotics: Extended Contributions from the 2009 EvoDeRob Workshop*, pages 139–154. Springer.
- Mouret, J.-B. and Doncieux, S. (2012). Encouraging behavioral diversity in evolutionary robotics: an empirical study. *Evolutionary Computation*, 20(1):91–133.
- Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127.

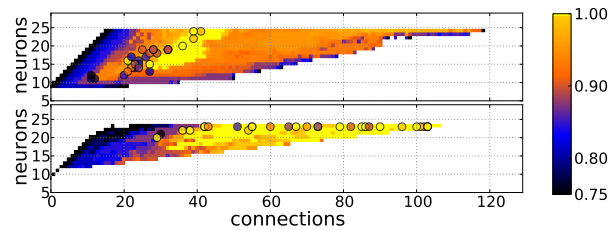


Figure 1: **Experiment:** A straightforward set of dimensions for evolving neural networks: number of nodes vs. number of connections. Fitness is colored. Circles indicate the best solution from each of 30 standard EA runs (some overlap). **Results:** (Top) Phenotype-fitness map obtained with the DNN encoding. The MOLE algorithm found 98 distinct perfect solutions (bright yellow areas) whereas 30 runs of a standard EA found only 6 perfect solutions (bright yellow circles). (Bottom) Phenotype-fitness map obtained with the KA encoding. The MOLE algorithm found 221 perfect solutions whereas a standard EA found 15. **Comments:** These maps reveal relationships between the dimensions and fitness, such as the minimum number of neurons and connections needed to solve the problem. The maps provide more insight into these relationships than the EA does alone. The maps also reveal the impact of different encodings: while all KA networks are expressible in the DNN encoding, evolution with DNN is less likely to evolve fit solutions with many connections, shedding light on why the constraints in the KA encoding improve its performance.

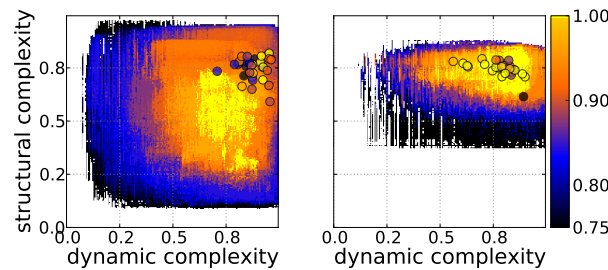


Figure 2: **Experiment:** We tested a second set of phenotypic dimensions on the same problem to understand the relation between fitness and *network complexity*. Of the many ways to evaluate structural complexity in networks (Kim and Wilhelm, 2008), we chose “off diagonal complexity” (Claussen, 2007) because it is fast to compute. With this measure, the complexity of fully connected networks and completely regular networks is zero, the complexity of a random graph is intermediate, and the highest complexity scores correspond to scale-free networks and hierarchical trees. In addition to structural complexity, neural networks can vary in the dynamics of their activity. This complexity can be captured by computing the Kolmogorov complexity of the sequence of outputs of each neuron for each input pattern. Here we approximate Kolmogorov complexity using the gzip2 compressor (Li and Vitányi, 2008). **Results:** (axes are normalized) (left) DNN phenotype-fitness map: MOLE found 2536 distinct perfect solutions whereas a standard EA found 6 (bright yellow circles) (right) KA encoding phenotype-fitness map: MOLE found 1690 perfect solutions whereas a standard EA found 15. The maps illuminate relationships between the dimensions, although the illumination is not flawless, as some perfect solutions found by the standard EA were not located by MOLE.