Electronic Supplementary Material (ESI) for Integrative Biology.
This journal is © The Royal Society of Chemistry 2016

Electronic Supplementary Information

Optimizing Micropattern Geometries for Cell Shape and Migration with Genetic Algorithms

Philipp J. Albert and Ulrich S. Schwarz

Institute for Theoretical Physics and BioQuant, Heidelberg University, Philosophenweg 19, 69120 Heidelberg, Germany

A. Exemplary Description of A Genetic Algorithm

An example is most instructive to explain the genetic algorithm (GA) in detail. In many problems a binary representation of individuals is used. A sample genome then looks like this:

$$\text{Genome} \qquad G = \boxed{1|0|0|1|1|0|1|0|1}, \tag{S1}$$

where each box represents a gene. The individual shown in (S1) has a genome consisting of nine genes. The first one being $g_1 = 1$, the second $g_2 = 0$ and so on. A very simple fitness function is

$$F(G) = \sum_{\text{Gene } i} g_i \tag{S2}$$

which has a maximum for all genes g_i being one. The algorithm now works as follows:

1. Start with a random population. E.g. with four individuals

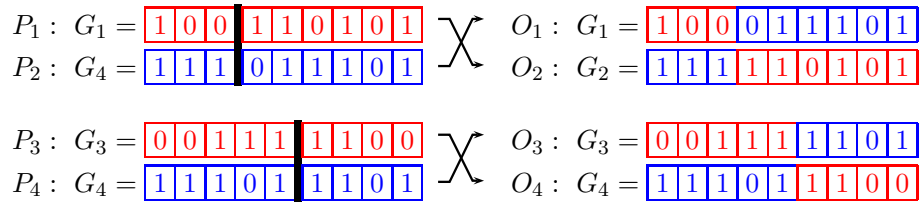
$$\begin{aligned} \text{Member 1 :} \quad G_1 &= \boxed{1|0|0|1|1|0|1|0|1} \\ \text{Member 2 :} \quad G_2 &= \boxed{1|1|0|0|1|0|1|1|1} \\ \text{Member 3 :} \quad G_3 &= \boxed{0|0|1|1|1|1|1|0|0} \\ \text{Member 4 :} \quad G_4 &= \boxed{1|1|1|0|1|1|1|0|1}. \end{aligned} \tag{S3}$$

2. Fitness evaluation of each individual. With the Fitness function Eq. S2 this is

$$\begin{aligned} F(G_1) &= F\left(\boxed{1|0|0|1|1|0|1|0|1}\right) = 5 \\ F(G_2) &= F\left(\boxed{1|1|0|0|1|0|1|1|1}\right) = 6 \\ F(G_3) &= F\left(\boxed{0|0|0|1|1|1|1|0|0}\right) = 4 \\ F(G_4) &= F\left(\boxed{1|1|1|0|1|1|1|0|1}\right) = 7 \end{aligned} \tag{S4}$$

3. Parent selection according to their Fitness. A possible method is fitness proportional selection with probability $p_i = F(G_i)/\sum_j F(G_j)$ for the i-th individual. The value of $p_i \in [0,1]$ reflects the relative fitness of an individual. Individuals are selected with the probability p_i for reproduction. For this example we assume that G_4 is select twice and G_1 and G_3 once. G_3 is not selected.

4. Recombination: The genomes of parents selected for recombination are crossed e.g. by one point crossover where genomes of couples are split at random location and then mixed.



(S5)

   The colors illustrate the crossover of the genomes. Pairing of parents is random from the ones selected in step three but self crossover (G_4 with G_4) is avoided. Parents are labeled with P_i and offspring with O_i. Selection and crossover have increased the average fitness of the population from 5.5 to 6 in this example.

5. Mutation of offspring e.g. by randomly flipping a bit of the offspring population with a certain probability.

6. The offspring become the new parent generation and the iteration starts again at step 2.

The algorithm is terminated when a certain fitness is reached or no significant changes occur over several iteration steps.

There exist a large variety of parent selection, crossover and mutation operators [1]. GA algorithm are also not restricted to using a binary genome and we use a floating point representation. Our pattern representation, crossover, mutation and parent selection operators are described in detail in the following.

B. Pattern Representation

To encode pattern shapes in a genome we use two different methods. The first one uses arc shapes and the second a Fourier series. While the genome (S1) uses a binary representation GA also work with floating point numbers as genes. For the arc representation we take six numbers, namely the position (x, y) of an arc, its length L, width d, radius R and orientation α as shown in Figure 2B of the main text. A single gene consists of these six numbers. The corresponding genome of a pattern consisting of three arcs (=genes) then looks like

$$G_{\text{Arc}} = \boxed{x_1, y_1, L_1, d_1, R_1, \alpha_1 \,|\, x_2, y_2, L_2, d_2, R_2, \alpha_2 \,|\, x_3, y_3, L_3, d_3, R_3, \alpha_3}. \tag{S6}$$

Examples are shown in Figure S2A. The single point crossover mechanism explained in (S5) still works with this genome. It exchanges whole arcs to form offspring.

The second approach we take to encode shapes in a genome is by Fourier descriptors (FD) [3]. Given a connected shape in the complex plane its contour can be expressed with a parameter curve $z(s) = x(s) + \mathrm{i}y(s)$ with parameter $s \in [0, 1)$ as illustrated in Fig. S1A. The periodicity

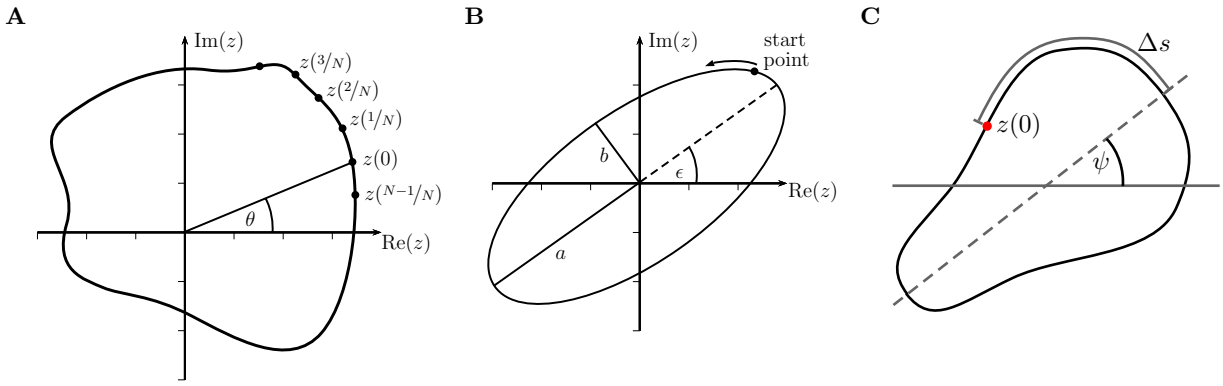


Figure S1 (A) Two dimensional shape $z(s)$ in the complex plane with equidistant sample points. The total number of points is N and the first and last point are identical ($z(0) = z(N)$). The polar angle is indicated by θ. In general $\theta(s)$ is not a monotonic function of the parameter s. (B) Ellipse with four degrees of freedom, the semi-axis a, b, tilt ϵ and phase of the start point. (C) Shape with mirror symmetry. The symmetry axis is indicated by the dashed line and rotated by ψ with respect to the horizontal axis. The start point of the contour curve $z(s)$ is indicated by a red dot. It is shifted by Δs from the symmetry axis.

$z(s+n) = z(s),\ n \in \mathbb{Z}$ can be used to expand the contour in a truncated Fourier series (equation (1) of the main text)

$$z(s) = \sum_{\nu=-N_{\max}}^{N_{\max}} z_\nu \exp\left(2\pi \mathrm{i} \nu s\right). \tag{S7}$$

The complex valued coefficients z_ν are given by

$$z_\nu = \frac{1}{N} \sum_{n=0}^{N-1} z\left(\frac{n}{N}\right)\ \mathrm{e}^{-2\pi \mathrm{i} \nu \frac{n}{N}}, \tag{S8}$$

with the equidistant sampling points $z(n/N)$ shown in Fig. S1A. A pair composed of a positive and corresponding negative coefficient $(z_\nu, z_{-\nu})$ encodes an ellipse as shown in Fig. S1B. The pair of complex valued coefficients has four degrees of freedom which correspond to the two semi-axis, the tilt and the start point of the parameterization along the contour.

A single gene now holds one coefficient z_ν and a genome becomes

$$G_{\mathrm{FD}} = \boxed{z_0}\boxed{z_1}\boxed{z_{-1}}\boxed{z_2}\boxed{z_{-2}}\boxed{z_3}\boxed{z_{-3}}\cdots\boxed{z_{N_{\max}}}\boxed{z_{-N_{\max}}}. \tag{S9}$$

Examples are shown in Fig. S2B. In contrast to radial FD (see [3] for definition) Cartesian FD are not limited to simple valued boundaries and shapes can have overhangs. Only connected regions can be represented, although the regions can have holes as shown in Fig. S2B. FD have several useful properties to describe cell and pattern shapes and to compare them. To name a few, translation of a shape is achieved by changing the coefficient z_0, shapes can be made scale invariant by dividing all coefficients by $|z_1|$ [2]. Higher order coefficients represent finer features or, when a cell shape is described, truncating the series Eq. S7 early cuts off noise in the shape. Mirror symmetry can be achieved if the coefficients fulfill

$$y_\nu = x_\nu \tan(2\pi\nu\Delta s + \psi), \tag{S10}$$

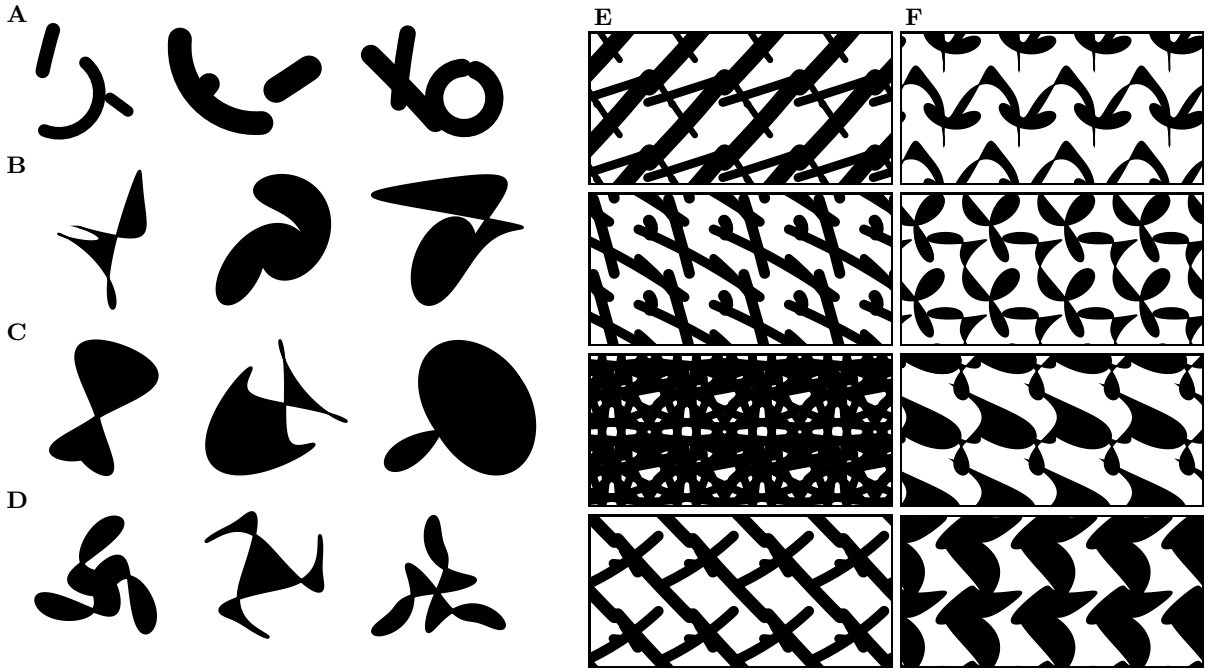


Figure S2 Examples of patterns generated from genomes with random entries (A) Arc representation of patterns with genome size restricted to three. (B) Patterns generated from Fourier descriptors (FD) as defined in (1) of the main text with $N_{\max} = 9$. (C) Patterns generated from FD with the symmetry condition Eq. S10. (D) Patterns generated from FD with three-fold symmetry achieved by only using coefficients $z_{1\pm3\nu}$ ($z_1, z_4, z_{-2}, z_7, z_{-5} \dots$). All other coefficients are zero. (E) Ratchet patterns generated from a random genome representing arcs. A unit cells containing four arcs is arranged into a 4 x 2 square lattice with periodic boundary conditions. (F) Same as previous figure with patterns represented by Fourier descriptors.

where ψ denotes the orientation of the symmetry axis and Δs is the offset of the start point of the parameter curve $z(s)$ from this axis as illustrated in Fig. S1C. Shapes that fulfill equation Eq. S10 are shown in Fig. S2C. This condition can also be used to define a measure for how symmetric a shape is. Rotational symmetry is achieved by setting certain coefficients to zero. For a m-fold rotational symmetry only the coefficients with $z_{1\pm\nu m}$ are different from zero [3]. E.g. the three fold symmetrical shapes in Fig. S2D have only the non-zero coefficients $z_1, z_4, z_{-2}, z_7, z_{-5}$.

Both pattern representation methods can also be used to generate periodic lattices. By putting a shape represented by one of the two methods into a unit cell and repeating this unit cell several times a lattice is generated. Examples are shown in Fig. S2E for a arc parameterization and Fig. S2F for a Fourier parameterization.

C. Parent Selection, Mutation, Crossover and Elitism

C.1. Parent Selection

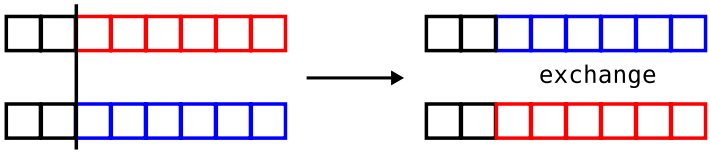
The example in section A uses fitness proportional selection in step three to select tow parent individuals for recombination. Each parent is selected according to its fitness F_i with probability

$$p_i = \frac{F_i}{\sum_j F_j}, \tag{S11}$$

where the sum extends over all members of a population. Other variants exists [1], e.g. by adding an offset to the sum to make the selection of less fit members more likely. Fitness based parent selection is closely coupled to the definition of the fitness function. A strongly nonlinear fitness measure could make selection of the fittest member much more likely. To avoid this dependence a ranking based selection can be used. Here, members are sorted according to their fitness and selected with a probability proportional to their rank. This avoids problems with large differences in fitness. However, we use fitness proportional selection as defined in equation (S11) because it was found to perform well.

C.2. Recombination

As with Parent Selection a wide range of recombination methods exists. We focus on recombination of two parents. Crossover operations between multiple parents exist [1] but are not considered here. Parents can be selected multiple times to generate offspring with different partners, but each pair of parents generates just two offspring keeping the total population size fixed. One point crossover (S5) was already mentioned when the algorithm was introduced. It can be applied to genomes with a binary or floating point representation. Genes between the parents are exchanged which mimics the chromosome exchange in sexual reproduction



(S12)

Each square represents a gene. A crossover point is selected indicated by the vertical line at a random position (selected with uniform probability). Genes behind the crossover point are exchanged between the two parents. E.g. if patterns are represented by several arcs (each arc is then a square), offspring are a combination of the arcs present in both parents. E.g. for individuals mad of three arcs an offspring is formed by taking the first and second arc from the first parent the third arc of the second parent. This crossover method conserves the order of the genes which is important if patterns are represented by FD where the coefficient scale with their order. One point crossover as defined in (S12) was found to perform better than arithmetic crossover operations where the floating point values of certain genes are averaged.

C.3. Mutation

Besides crossover mutation is the second main search operator. For binary genomes a random bit flip with a small probability is often used. In a similar fashion the number in a floating point representation can be randomized. However, this was found to destroy progress made by the algorithm quite often due to the total randomization of genes. A more subtle method [1] for floating point genomes is a gradual change by addition of a random number

$$g = \mathbf{x} \longrightarrow g = \mathbf{x} + \mathbf{N}(0, \sigma_m). \tag{S13}$$

Here $\mathbf{x}$ stands here for all values a single gene encodes and $\mathbf{N}(0, \sigma_m)$ are random numbers drawn from a Gaussian distribution with zero mean and variance σ_m. This mutation is applied to every gene and a small variance ensures that most changes are small. Large changes are still possible due to the Gaussian distribution but occur with low probability.

A population can loose its diversity if the mutation is too weak in a process called premature

convergence [1]. Through crossover all members become very similar and no more progress is made. This situation can be avoided by adapting the mutation strength (S13) to the diversity of the population

$$g = \mathbf{x} \longrightarrow g = \mathbf{x} + \mathbf{N}(0, \sigma_m/\sigma_p), \tag{S14}$$

where σ_p is now the variance of the fitness of the whole population. In case of an almost homogeneous population with low σ_p the mutation strength is increased while for very diverse populations it is lowered. The mutation strength is mainly chosen empirical and set to 0.001.

In our approach one gene encodes different properties. In an arc representation, a gene encodes position, radius, length and so on as defined in (S6). We constrain each of these values by boundaries to avoid situations where arcs become spaced far apart from each other, very thin/thick or too short/long. Written down in a general way a gene with its components labeled by j becomes

$$g = \mathbf{x} = (x_1, x_2, \dots, x_j, \dots) \qquad x_j \in [x_{j,\min}, x_{j,\max}]. \tag{S15}$$

E.g. one of the x_i denotes the length of an arc. The scale of the boundaries $x_{j,\min}$ and $x_{j,\max}$ is set by the scale of the cells the patterns are made for. For pattern represented by FD descriptors the size is controlled mainly by the first order coefficient and all subsequent coefficients are expected to be smaller. The mutation strength σ_m is scaled by the constraints to account for different magnitudes.

C.4. Elitism

Elitism is a mechanism which avoids loss of the fittest members by mutation or crossover. It keeps track of the fittest members as the algorithm progresses and places them directly into the offspring generation. The members with the smallest fitness are replaced by the best. We apply elitism and carry over for 10% of the fittest members to the next generation.

Supplementary References

[1] A. E. Eiben and J. E. Smith. *Introduction to evolutionary computing*. Natural computing series. Springer, Berlin ; Heidelberg [u.a.], corr. 2. p edition, 2007. ISBN 978-3-540-40184-1. URL `https://katalog.ub.uni-heidelberg.de/cgi-bin/titel.cgi?sess=bdf1892ce1f8d3d3babff25841136830&katkey=66434877&konto=a`.

[2] A. Folkers and H. Samet. Content-based image retrieval using Fourier descriptors on a logo database. *Pattern Recognition, 2002. Proceedings. ...*, pages 521–524, 2002. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1047991`.

[3] B. Jähne. *Digital image processing*. Engineering online library. Springer, Berlin ; Heidelberg [u.a.], 5., rev. a edition, 2002. ISBN 3-540-67754-2 ; 978-3-540-67754-3. URL `http://amazon.com/o/ASIN/B0088PZENC/`.