BIOINFORMATICS ORIGINAL PAPER

Vol. 23 no. 24 2007, pages 3356–3363
doi:10.1093/bioinformatics/btm433

Systems biology

Efficient parameter estimation for spatio-temporal models of pattern formation: case study of *Drosophila melanogaster*

Yves Fomekong-Nanfack[1], Jaap A. Kaandorp[1,*] and Joke Blom[2]

[1]Section Computational Science, Faculty of Science, University of van Amsterdam, Kruislaan 403, 1098 SJ Amsterdam and [2]Center for Mathematics and Computer Science (CWI), Department MAS, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands

Received on April 22, 2007; revised on August 16, 2007; accepted on August 17, 2007

Advance Access publication September 24, 2007

Associate Editor: Chris Stoeckert

ABSTRACT

Motivation: Diffusable and non-diffusable gene products play a major role in body plan formation. A quantitative understanding of the spatio-temporal patterns formed in body plan formation, by using simulation models is an important addition to experimental observation. The inverse modelling approach consists of describing the body plan formation by a rule-based model, and fitting the model parameters to real observed data. In body plan formation, the data are usually obtained from fluorescent immunohistochemistry or *in situ* hybridizations. Inferring model parameters by comparing such data to those from simulation is a major computational bottleneck. An important aspect in this process is the choice of method used for parameter estimation. When no information on parameters is available, parameter estimation is mostly done by means of heuristic algorithms.

Results: We show that parameter estimation for pattern formation models can be efficiently performed using an evolution strategy (ES). As a case study we use a quantitative spatio-temporal model of the regulatory network for early development in *Drosophila melanogaster*. In order to estimate the parameters, the simulated results are compared to a time series of gene products involved in the network obtained with immunohistochemistry. We demonstrate that a (μ, λ)-ES can be used to find good quality solutions in the parameter estimation. We also show that an ES with multiple populations is 5–140 times as fast as parallel simulated annealing for this case study, and that combining ES with a local search results in an efficient parameter estimation method.

Supplementary information and availability: *Bioinformatics* online; software: http://www.science.uva.nl/research/scs/3D-RegNet/fly_ea

Contact: jaapk@science.uva.nl

1 INTRODUCTION

In many animals, morphogen gradients specify different structures starting from a single cell at early embryo development (Gilbert, 2006; Wolpert, 1969; Houchmandzadeh *et al.*, 2002). The morphogens provide spatial information by forming concentration gradients that subdivide the developing embryo in different regions. Distinct cell types and structures emerge as a consequence of the different combinations of morphogen gradients. This is a general mechanism by which cell type diversity and structures can be generated in body plan formation. Understanding the body plan formation also requires understanding the underlying biochemical process. This is the level at which genes influence the transcription of other genes. Genetic regulatory networks (GRNs) can be described in terms of a network of interactions between genes and proteins.

Several mathematical rule-based models (see de Jong, 2002 and references therein) have been proposed to describe GRNs. In modelling pattern formation, spatially coupled ordinary differential equations (ODEs) and partial differential equation (PDEs) have been used to describe the temporal and spatial behaviours of the genetic interaction in the system. The goal is to understand the GRNs by quantitative simulation of the model to reproduce a spatial temporal pattern obtained from experimental data. Quantitative models (Reeves *et al.*, 2006) are in general used to test the GRNs underlying the mechanisms behind the pattern formation and to explore some principles such as evolvability and robustness. The model-building process can be described in three main steps:

(1) *Extraction of spatio-temporal gene expression data in a quantitative way*
(2) *Modelling in terms of mathematical equations*
(3) *Parameter estimation: finding the optimal parameters that provide the best fit with respect to the model solution.*

When one provides a quantitative model to infer the mechanism behind pattern formation ruled by a GRN, analysis of the dynamics is necessary. Assuming that all parameters are known from literature or experimental measurements (i.e. all initial conditions, kinetic coefficients in the biochemical system, diffusion coefficients, transcription-binding factors and the spatial domain is specified), the inference problem consists in solving the equations and is called the *direct problem*. Then, by means of sensitivity analysis (Saltelli *et al.*, 2004), one can analyse the model robustness with respect to the parameters. Unfortunately, in practice many parameters are unknown and estimation of these parameters from experimental data is crucial for quantitative modelling of GRNs. This is called the

*To whom correspondence should be addressed.

inverse problem. In such a case, only the governing equations describing the system and possibly some of the parameters therein are given.

Inverse problems are typically ill-posed, and when the data concerned are spatial and temporal, the fitting procedure can be computationally very expensive. The parameter search space is usually unknown and grows exponentially with the number of unknown parameters. Therefore, the choice of an appropriate optimization technique is crucial. When prior knowledge about the parameter values is available, it is possible to use local search techniques (van Riel, 2006). In general, this is not the case and the fitness landscape of a quantitative model can only with difficulty be generated, or even worse, the search space is unrestricted. In such cases a global search procedure is required.

This approach has been used in the parameter estimation of models of biochemical pathways using kinetic equations (Katare *et al.*, 2004; Mendes and Kell, 1998). There are relatively few studies in literature in which GRNs or metabolic pathways are inferred from spatio-temporal data. Reinitz and Sharp (1995) and Gursky *et al.* (2004) studied *eve* stripe formation in early development of *Drosophila melanogaster* embryos using a genetic circuit model. They were able to infer a GRN of five genes from spatio-temporal data obtained from immunofluorescently stained embryos. In Reinitz and Sharp (1995), each parameter estimation took approximately 1 week of CPU time on a Sparc 2 using simulated annealing. Using as starting point the parameters obtained by Reinitz and Sharp(1995) and Gursky *et al.* (2004) applied a steepest descent algorithm to find the optimal parameters for the model. Jaeger *et al.* (2004a, b) inferred a GRN model of six genes with 62 unknown parameters from quantitative spatio-temporal expression data (Poustelnikova *et al.*, 2004) (Fig. 1) for the gap genes. Although the model could be spatially reduced to one dimension, the fitting procedure was extremely computationally expensive. Using parallel simulated annealing (PLSA) (Chu *et al.*, 1999; Lam and Delosme, 1988a, b), it took between 8 and 160 h on ten 2.4 GHz Pentium P4 Xeon processors (Jaeger *et al.*, 2004a, b).

Using a more complex approach, Perkins *et al.* (2006) considerably reduced the computational time to 1 day on a serial platform. Their strategy makes use of specific characteristics of the experimental gap formation data, namely. that the production of the various proteins takes place in specific parts of the domain. This strategy has three different stages. In the first stage, these domains are defined, matching the observed data, and a linearized chemistry is used as a model that effectively decouples the system in the chemistry dimension. In the second stage, the boundaries of the domain are fitted, and in the last stage, the fully coupled system is solved with a local search strategy and with as initial parameter guesses the parameter values estimated in the second stage. However, this type of bottom-up approach is in many cases not feasible. Therefore, a brute-force global approach is still the most frequently used method in the parameter estimation problem.

In this article, we discuss an approach to estimate model parameters from spatio-temporal data with a global search strategy. We investigate the efficiency of an evolution strategy for the parameter estimation of GRN models capable of simulating spatio-temporal patterns. Our choice is inspired by Moles *et al.* (2003) and Mendes and Kell (1998) where the authors compared different global optimization strategies and suggested that the evolution strategy is the most competitive and the only one capable of finding the true minimum in the parameter estimation of biochemical networks. We combine this approach with a local search strategy. As a case study, we infer from the FlyEx data (Poustelnikova *et al.*, 2004) the connectionist model consisting of six genes presented by Jaeger *et al.* (2004a, b) that describes the regulatory interactions in the gap gene system of the blastoderm stage of *D.melanogaster*.

2 METHODS

2.1 Case study: regulatory interactions in the gap gene system of *D.melanogaster*

The biological case chosen is a model of a GRN capable of simulating the spatio-temporal pattern formation in the early development of a *Drosophila* embryo. Much work has been done (Reinitz and Sharp, 1995; Reinitz *et al.*, 1998; Jaeger *et al.*, 2004a; Gursky *et al.*, 2004; Janssens *et al.*, 2006) to understand the role of GRNs in the segment determination system. The early *Drosophila* blastoderm is a syncytium containing nuclei not surrounded by a membrane. The pattern formation in the *Drosophila* blastoderm results from the interactions among segmentation genes. To simulate this, we use the model given by Jaeger *et al.* (2004a) based on a connectionist model (Mjolsness *et al.*, 1991). It is a dynamical model consisting of a discrete representation of the nuclei in space and a continuous regulation of the genes in time. The developmental time of interest is between cycles 13 and 14A before gastrulation (cleavage cycle n is the time between the $n-1$th and the nth cell division, c.f. Foe and Alberts, 1983). Three different rules describe the phenomena that occur during this time: interphase, mitosis and division (see Supplementary Material for details). The resulting model is a system of 348 equations with 66 unknown parameters

$$\frac{\mathrm{d}g_i^a(t)}{\mathrm{d}t} = \begin{cases} R_a\Phi_a\left(\sum_{b=1}^{N_g} W_a^b g_i^b + m_a g_i^{bcd} + h_a\right) & \text{(regulation)} \\ -\lambda_a g_i^a & \text{(decay)} \\ +D_a\left(g_{i+1}^a - 2g_i^a + g_{i-1}^a\right) & \text{(diffusion)} \end{cases}$$

where N_g denotes the number of genes or gene products involved and Φ is a sigmoid function with range (0,1). The model simulates the spatio-temporal evolution for the concentration of the genes *caudal*(*cad*), *hunchback*(*hb*), *Krüppel*(*Kr*), *giant*(*gt*), *knirps*(*kni*) and *tailless*(*tll*). $g_i^a(t)$ represents the concentration level at time t of gene a in nucleus i with $1 \leq i \leq N$ and N the number of nuclei during a cleavage cycle. The concentration, g_i^{bcd}, of the maternal gene *bicoid* is taken from experimental observations and is kept constant in time during the simulation. The parameters are: the regulatory weight matrix W_a^b, describing the influence of gene b on gene a, the production rate R_a, the activation threshold h_a for Φ, the decay rate λ_a, the diffusion coefficient D_a and the regulatory influence m_a of maternal *bcd*. Initial gene expression levels are available from experiments. For the genes *Kr*, *gt*, *kni* and *tll*, these are very close to zero and set to 0 in the simulations. The data we have used to fit the model are the same as used by Jaeger *et al.* (2004a). These data are available from the FlyEx database http://flyex.ams.sunysb.edu/flyex/ (Poustelnikova *et al.*, 2004). The model and datasets used in the parameter estimation are discussed in detail in the Supplementary Material.

2.2 Optimization criteria

Given a model that simulates spatio-temporal data, the problem is to estimate the unknown parameters such that the simulation results fit some observed spatio-temporal (target) data. The parameter estimation is done by optimization techniques. The optimization goal is to find those values of the unknown parameters that minimize a scalar valued *cost-function*, by exploring the set of possible values in an allowed search space. As in the previously mentioned *Drosophila* studies, we have chosen to use as cost-function the least-squares of the difference of the simulated and the observed data:

$$E(\theta) = \sum_{i,t} (g_i^a(t,\theta)_{\text{model}} - g_i^a(t)_{\text{data}})^2 \tag{1}$$

with θ the parameter vector, to which a constraint or penalty function is added. An explicit search-space constraint is given for parameters R_a, λ_a and D_a. For the parameters W_a^b, m_a and h_a a collective penalty function is used (Reinitz and Sharp, 1995) to restrict the function value of Φ to the domain $[\Lambda, 1-\Lambda]$ with Λ a small parameter (in this study taken to be 0.001) (for details see Supplementary Material). We use the root mean square (RMS) (Reinitz and Sharp, 1995) as a measure of the quality of a model solution for a given set of parameters:

$$RMS = \sqrt{\frac{E(\theta)}{N_d}} \tag{2}$$

where $E(\theta)$ is given by Equation (1) and N_d is the number of data points.

2.3 Global search by evolution strategy

An evolution algorithm (EA) is a stochastic iterative algorithm that operates on some encoded individuals from an initial population. It consists of three main operators: crossover, mutation and selection. The first two are exploration operators of the search space, while the last one lets the population evolve towards the optima of a problem. Marnellos (1997) compared SA and a course-grained parallel island Genetic Algorithm on various biological problems (neurogenesis, curve-fitting and life history). The first two are continuous models and for these he reported that SA was the faster method, but GA had a faster initial convergence. Among all the existing EAs (see Bäck *et al.*, 1997; Spears *et al.*, 1993, for an exhaustive overview) such as Genetic Algorithms (Goldberg, 1989; Holland, 1992) or Evolutionary Programming (Fogel *et al.*, 1966), we have chosen an Evolution Strategy (ES) (Beyer, 1996). ES shows proven superiority, compared to other classical EAs for problems with a high number of unknown parameters (Runarsson and Yao, 2000; Moles *et al.*, 2003). In contrast with the original Genetic Algorithm, ES has initially been designed for a constrained continuous variable optimization problem. Like most EAs, it is a stochastic process that modifies an original population of individuals from iteration to iteration with the aim of minimizing an objective function. Evolution from generation to generation is based on a deterministic selection and a stochastic mutation. One of the main advantages of ES compared to standard EAs is the usage of strategic parameters such as on-the-fly adaptation of the mutation parameters. In this study, we use a modified (μ, λ)-ES, based on stochastic fitness ranking. This method is simple and has proven to be more efficient than most EAs and ESs for large parameter estimation problems (Runarsson and Yao, 2000, 2005). A pseudo-algorithm is given in Algorithm 1.

Algorithm 1 (μ, λ)-ES

```
1: INITIALIZATION: generate an initial population of $\lambda$ individuals according to an $n$-dimensional probability distribution over the search space.
2: while termination criteria not met do
3:     SCORE: evaluate each individual objective function.
4:     RANKING: sort individuals based on a stochastic ranking.
5:     SELECTION: select the $\mu$ best individuals out of $\lambda$ offspring as parents for the next generation.
6:     RECOMBINATION: apply recombination only to the best $\mu$ individuals (differential evolution).
7:     MUTATION: Gaussian mutation is applied to the other individuals in the population (with boundary control).
8: end while
```

The main part in ES is the creation of λ new offspring (Algorithm 1, steps 6 and 7) by recombining two parents and mutating the individuals. We use a global intermediate recombination described in Equation (3) and a non-isotropic mutative self-adaptation rule (Runarsson and Yao, 2000) described in Equations (4–6). The recombination is given by

$$\theta_c' = r(\theta_o, \theta_c, \theta_{c+1}) = \theta_c + \gamma(\theta_o - \theta_{c+1}) \tag{3}$$

where θ_i is the parameter vector of an individual i, individual o is the highest ranked individual (the fittest) and γ is the recombination factor. In this way, a number of μ new individuals are created from the λ offspring. The individuals c are chosen among the best μ offspring obtained after a stochastic ranking (Runarsson and Yao, 2000). The rest of the new population is filled with the (unchanged) μ best individuals (repeatedly). Mutation is applied to these $\lambda - \mu$ individuals according to the *non-isotropic* self-adaptation rule:

$$\sigma_{k,j}' = \sigma_{(i;\lambda),j} \exp \tau' N(0,1) + \tau N_j(0,1) \tag{4}$$

$$\theta_{k,j}' = \theta_{(i;\lambda),j} + \sigma_{k,j}' N_j(0,1) \tag{5}$$

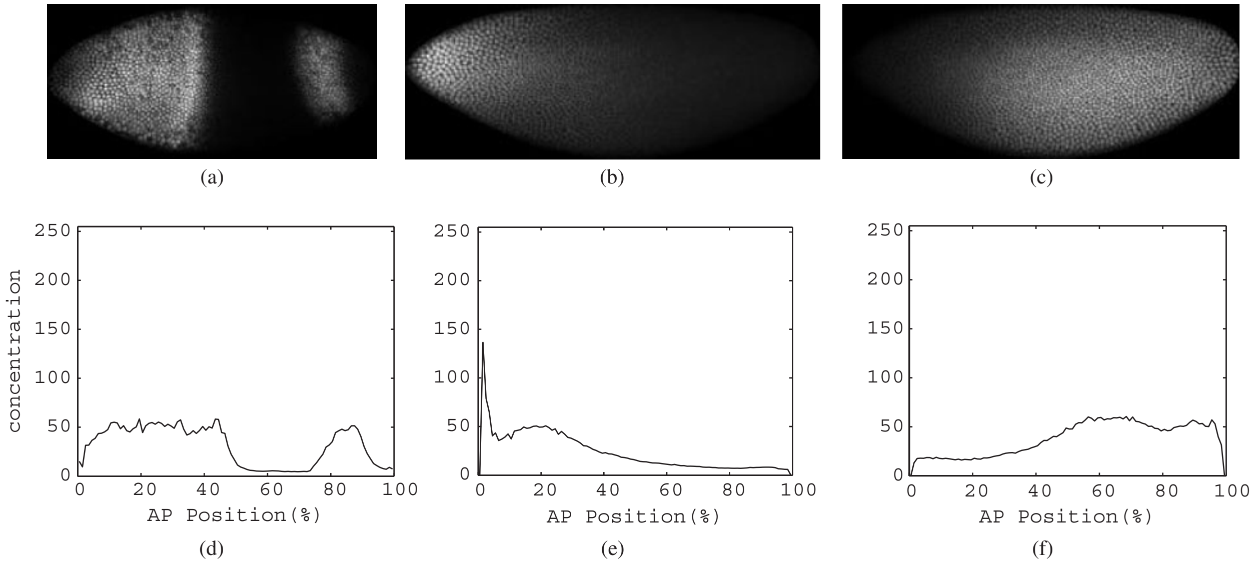
$$\sigma'_k = \sigma_i + \alpha(\sigma_k' - \sigma_i) \tag{6}$$

with $k = \mu + 1, \ldots, \lambda$, $i = k \bmod \mu$ and $j = 1, \ldots, n$. σ_k' is the step-size control per individual (parameter vector) k and $\sigma_{k,j}'$ an element of this vector. σ aims to tune the search distribution so that maximal progress is maintained (mutations become smaller as the search progresses). $\tau = \varphi^* / \sqrt{2\sqrt{n}}$ and $\tau' = \varphi^* / \sqrt{2n}$ are the learning rates for a parameter and for an individual, respectively, and $\varphi^* = 1$ is the expected rate of convergence. Finally, $N(0,1)$ is a normally distributed uniformly random variable and $N_j(0,1)$ a new random variable for each parameter j. Equation (6) implies an exponential smoothing of the σ mutation parameter for reducing random fluctuations in the self adaptation, with $\alpha = 0.2$ as the smoothing factor. For an explanation of this mutation strategy we refer to Runarsson and Yao (2005).

2.3.1 Island evolution strategy

We have developed an evolution strategy with multiple subpopulations (island-ES, also called regional model or island model). In this article, the focus is not on the performance in terms of computational time of a parallel version of ES, but on its effectiveness in terms of the quality of the solution. The island-ES used here is run on a single processor, working as a regional model. It has been shown (Cantú-Paz, 1995; Mühlenbein *et al.*, 1991) that an island evolution algorithm can qualitatively outperform a serial EA.

A number of subpopulations are defined to evolve, as described in Algorithm 1, independently of each other for a certain number of generations (isolation time or migration interval κ). After the isolation time, a number of individuals are distributed over the subpopulations by a procedure called migration. The number of exchanged individuals (migration rate π), the selection method of the individuals for migration and the scheme of migration determine the average genetic diversity in the subpopulations and the exchange of information between subpopulations. Multiple subpopulations initialized independently ensure a diverse set of individuals covering a large part of the optimization search space. The migration operation spreads the best individuals over subpopulations. In this study, we migrate the best one of the μ selected parents randomly every 500 generations to other subpopulations. This elitist migration ensures that the new individual inserted in a subpopulation can allow the population to escape local minima if trapped in one with a high value of the cost-function. We use a complete net structure (Lohmann, 1991) with random assignment.



(a) (b) (c)

(d) (e) (f)

Fig. 1. Gene expression data. **a**,**b** and **c** correspond to confocal images of stained *Drosophila* blastoderm embryos. Staining is done by fluorescent immunohistochemistry (Kosman *et al*., 1998). **d**,**e** and **f** are the average quantitative gene expression levels obtained by successive image-processing operations (Myasnikova *et al*., 1999, 2001). Images are from the late blastoderm stage cleavage cycle 14A; (a,d) time class 8 for *hunchback* (embryo ba3); (b,e) and (c,f) time class 1 for *bicoid* and *caudal*, respectively (embryo cb11). Images are from the FlyEx database http://flyex.ams.sunysb.edu/flyex. The y axis gives the relative protein concentration expression level normalized to a fluorescence intensity range from 0–255. The x axis corresponds to the anterior–posterior (A–P) axis of the embryo.

At each migration, an individual from a population can migrate to any other subpopulation.

2.4 Local search

Global search is often used for parameter estimation problems where no information on parameters is available. Although it has proven to be efficient in many problems to identify promising regions, a slow convergence when reaching the global minima is always observed. Combining a global search with a local optimizer to identify the minimum speeds up convergence. The hybrid approach used was inspired by Katare *et al*. (2004) where the authors have successfully used a hybrid genetic algorithm to estimate parameters of small (5 parameters) and large (31 parameters) kinetic model of propane aromatization on a zeolite. Also, Gursky *et al*. (2004) used a local search to refine the parameters obtained after a global search by simulated annealing.

There is a large variety of local search techniques. Most local optimizer techniques such as Powell's method, the quasi-Newton methods or Levenberg–Marquardt are based on the gradient descent approach and thus require the derivative of the objective function $f(\theta)$. If analytic expressions are not available for the derivative, a finite-difference approximation of the gradient of $f(\theta)$ can be used. In many situations, computing the objective function $f(\theta)$ can be expensive and numerical approximation of the gradient of $f(\theta)$ is thus too costly. Furthermore, biological data can be noisy, making the use of the gradient difficult if not impossible. In these cases, Newton-like local optimizers become inappropriate. A good alternative is a direct search method. Direct search such as generating set search (Kolda *et al*., 2004; Lewis *et al*., 2005), pattern search (Hooke and Jeeves, 1961) or downhill simplex(DS) (Nelder and Mead, 1965) are suitable to solve a variety of optimization problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, non-differentiable, stochastic or highly non-linear. In this study, we use the DS as local search strategy. DS assumes that the initial starting point (simplex) is around a local minimum. Simplex-based direct search methods are based on a comparison of the cost-function values at the vertices of a simplex that is updated by the algorithm steps [a simplex is the geometrical figure consisting, in N dimensions, of $N+1$ points (or vertices) and all their interconnecting line segments, polygonal faces, etc. giving in 2D a triangle and in 3D a tetrahedron.]

2.5 Validation

To validate our optimization method, we have reverse-engineered 'artificial GRNs data'. Results of these validation tests can be found in the Supplementary Material.

3 RESULTS

The purpose of the model presented in Section 2.1 is to simulate the pattern formation of the early *Drosophila* embryo, as described in Section 1 and shown in Figure 1. The aim of the optimization is to find suitable model parameters that can simulate realistic patterns, in comparison with real quantified gene expression patterns. The search space is based on Jaeger *et al*. (2004a), but it is slightly enlarged for some parameters (see Supplementary Material). Different settings for (μ, λ)-ES are used followed by DS local search. The population size λ is varied, in ES $\lambda = \{200, 350, 500\}$ and in the island ES with four subpopulations $\lambda = 500/4 = 125$. The other method parameters are in all cases $\mu = \lambda/5$, $\gamma = 0.85$ and $\alpha = 0.2$ (Runarsson and Yao, 2005). In all settings 20 optimization runs have been performed. To facilitate comparison the initial populations in the different settings are generated using the same 20 random seeds and the number of generations for different λ is such that the (sequential) computational time is comparable in all runs. The DS is applied to each resulting gap gene circuit and runs for 130 000 iterations. All simulations are

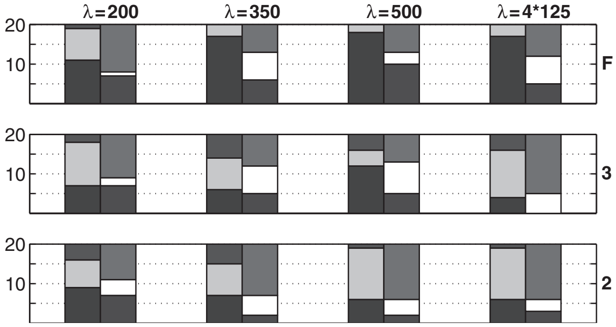


Fig. 2. Comparison of the different optimization runs for (F) full search, (3) reduced search with activation thresholds set at −3.5 and (2) reduced search with activation thresholds set at −2.5. Each bar-column represents 20 runs of a setting. Duo bar-columns are read as follows: left: after ES, right: after DS; bottom bar: RMS $>$ 14, middle(light): RMS $\in$ (12,14), top: RMS $\leq$ 12.

performed on a serial 3.4 GHz 'Intel Xeon' processor and took 8–11 CPU hours for the complete ES + DS search.

3.1 Full search

The first setting assumes that no a priori knowledge is available regarding any of the 66 parameters other than the search space. After the global search only one gap gene circuit has an RMS <12 and did not show any specific defect. In Figure 2 we have visualized the results. A table with the exact numbers is given in the Supplementary Material (Table 2). The parameters for the acceptable gap gene circuits are of similar quality as Jaeger *et al.* (2004a, b) but show a somewhat larger diversity due to the full search and the larger search space.

3.2 Reduced search

In this setting the 20 optimizations are first run with the activation thresholds h_{hb}, h_{Kr}, h_{gt} and h_{kni} at a nominal value of −3.5, as suggested by Jaeger *et al.* (2004a). For the other parameters we have set the parameter search space as in the previous 'Full Search' setting. The problem is now 62-dimensional. The fixation of the four activation thresholds results in a much easier optimization problem as can be judged from the fact that 16 out of the 80 runs result in an RMS <12 after the ES. Also the advantage of using the island search can be seen more clearly: 16 out of 20 runs result in an RMS <14, in contrast to the 8 in the (100 500)-ES runs.

A second series has been done with activation thresholds h_{hb}, h_{Kr}, h_{gt} and h_{kni} having as nominal value −2.5. As can be seen in Figure 2 ((3) and (2)) the results are comparable with the −3.5 setting.

In all cases where an RMS $<$ 12 was obtained the simulated patterns match nicely the real spatio-temporal data (see Fig. 3 for an example). As in Jaeger *et al.* (2004a), in some other cases there is a small defect, especially for the late and posterior *tll* concentration.

The parameters obtained are in most cases comparable for different optimization runs and with the ones obtained in Jaeger *et al.* (2004a). In some cases it is clear that the model results are not sensitive to significant changes in the parameter values, as can be seen for W_{hb}^{kni} in the left scatterplot in Figure 4. Incidentally our regulatory weight matrix entries differ from those in Jaeger *et al.* (2004a), like W_{tll}^{cad} and W_{tll}^{hb} in the right scatterplot in Figure 4. More scatterplots and a qualitative summary of the obtained weight matrices are given in the Supplementary Material—Figures 9 and 10, Table 6.

4 DISCUSSION

Modelling pattern formation in terms of their GRN implies a description of the interactions between the different genes. Although some network structure is known, in most cases very little quantitative information is known about these interactions. Therefore, given a network of m genes, inferring the regulatory network consists of estimating $m \times m + c$ parameters where c is the number of other parameters (decay-rate, diffusion, etc.). It is essential to have computational methods that allow to estimate these unknown parameters in a reasonable time.

4.1 Convergence ES

In Figure 5, we illustrate the convergence behaviour of the evolution strategy. In the left plot, the average fitness evolution is given for the 20 optimization runs with $N = 62$ and $h = -2.5$. In all cases a fast initial convergence is followed by a slow decrease of the fitness. Note that the lines represent an equal amount of computational work, so the runs with $\lambda = 200$ are allowed many more generations resulting in a slightly better RMS than the $\lambda = 500$ case. Comparing the latter with the island-based ES with four subpopulations of each 125 individuals, it is obvious that the island-ES gives a significantly better RMS. The reason is that the fittest individual within one subpopulation is migrated to another subpopulation that might be stagnating, hence the staircase behaviour of the fitness curves (Fig. 5, right plot). This feature makes an island-based ES also much more reliable (see also under reliability).

4.2 Combining global and local search

Following the idea that heuristic search cannot easily find true minima, coupling (μ, λ)-ES with a local search can considerably increase the quality of the solution and speed up the convergence. This works only if the output solution of the ES is already in the neighbourhood of a solution corresponding to a minimum. Simple (μ, λ)-ES could almost always find gap circuits with an RMS between 11.00 and 16.00 in an average of 8–11 CPU hours. As shown in Figure 5, a quick convergence of the objective function is always observed after a few generations of ES. These first steps are the main strength of ES. Changing to a local search strategy if the ES stagnates results in an efficient and reliable parameter estimation method (see also the Supplementary Material).

4.3 Reliability of the method

The stochastic nature of ES implies that one has to run many simulations in order to obtain 'possible' solutions. Approximately 50% of the ES + DS runs produced gap gene

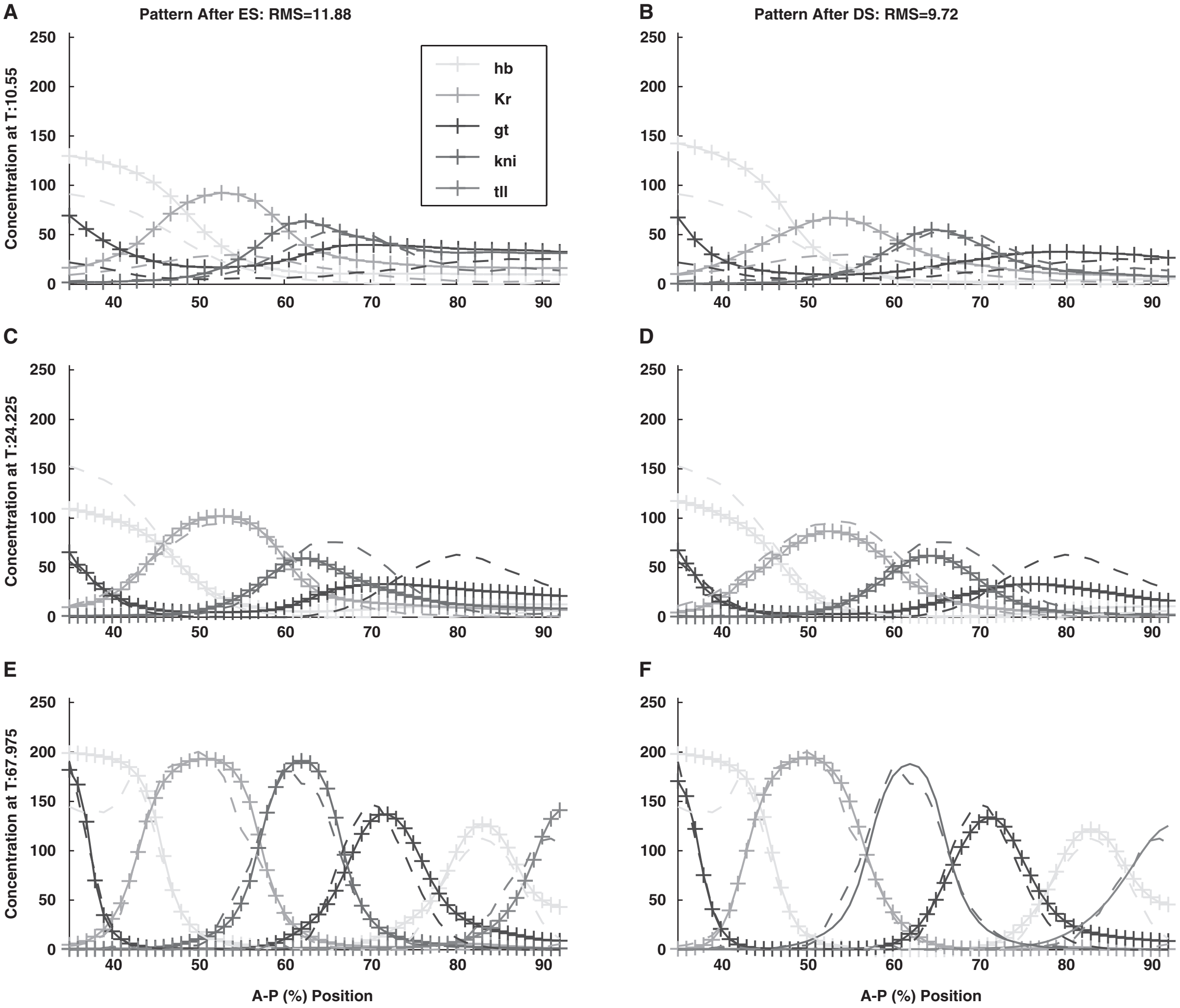


Fig. 3. Solution of the gap gene circuit gn52c13_200_62_25_14 at time points $T = 10.550$ and after division, $T1 = 24.225$ and $T8 = 67.975$ obtained after parameter estimation using (40 200)-ES (left) followed by Downhill simplex local search (right). Experimental (target) data is indicated with dashed lines.

circuits with a good RMS (≤12). This percentage is better than obtained by simulated annealing, as discussed in Jaeger *et al.* (2004a, b) where only 25% good solutions is reported.

Results obtained with the island-based (μ, λ)-ES show that in the reduced search setting (fixed *h*-values) 75% of the runs return gap gene circuits with an acceptable RMS (≤14), and if followed by a DS local search, 62% of the runs result in gap gene circuits with an RMS <12. The quality of the solutions obtained by the island version is comparable with the one obtained by the simple ES, but the number of solutions with an acceptable RMS is larger (75% versus 60%, c.f. Fig. 2 and Table 2 and 5 in the Supplementary Material). The higher reliability can be explained by the fact that each subpopulation evolves independently like a normal ES. When no improvement can be obtained in one of the subpopulations, or if the subpopulation is too homogeneous, a fully connected network migration is applied (in the current implementation this is done after a fixed number of generations, but it is possible to develop an adaptive strategy for this). Inserting new individuals in a subpopulation from another subpopulation allows each subpopulation to create diversity, and thus to escape from a local minimum.

4.4 Improvement of previous results

Jaeger *et al.* (2004a, b) presented 10 gap gene circuits including *bcd*, *cad*, *hb*, *Kr*, *gt*, *kni* and *tll* gene expression and covering a range of 35–92% of the *A–P* axis. These 10 gap gene circuits were selected among 40 results according to their RMS (≤12). Their results were obtained using a parallel simulated annealing method, and the computational time needed was between 8 and 160 h using ten 2.4 GHz processors for each simulation.

We have demonstrated that our method, (μ, λ)-ES followed by a DS search, gives solutions comparable to their solutions in terms of the RMS and in simulation results. In most cases, we find similar values for the parameters and a similar gap gene

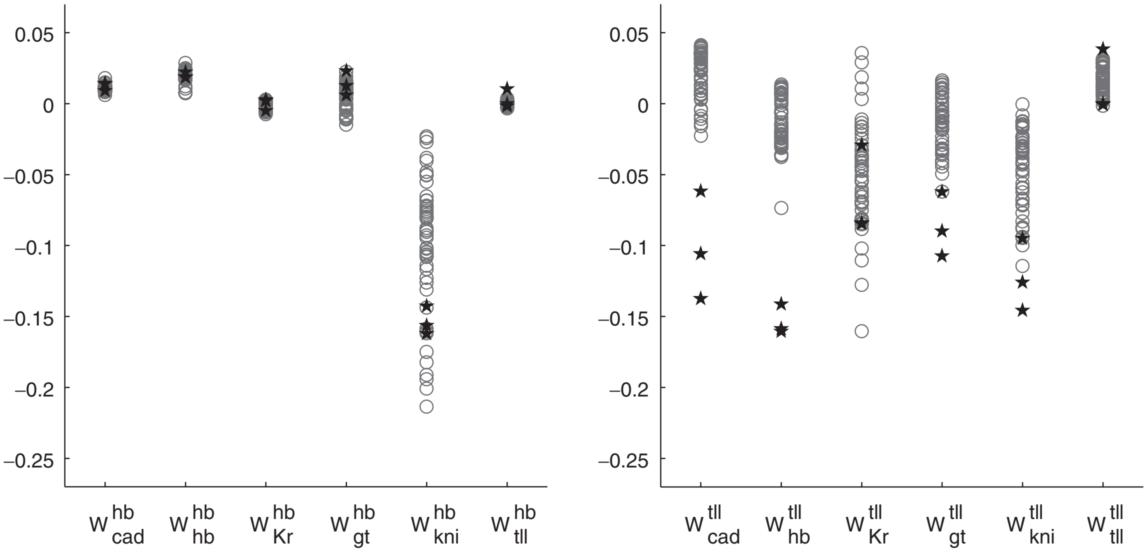


Fig. 4. Scatterplots for the regulatory weight parameters that describe the influence by *hb* and *tll*. Reduced search, activation threshold −2.5. For scatterplots of the other parameters see the Supplementary Material. The asterisks indicate the results by Jaeger *et al*. (2004a).

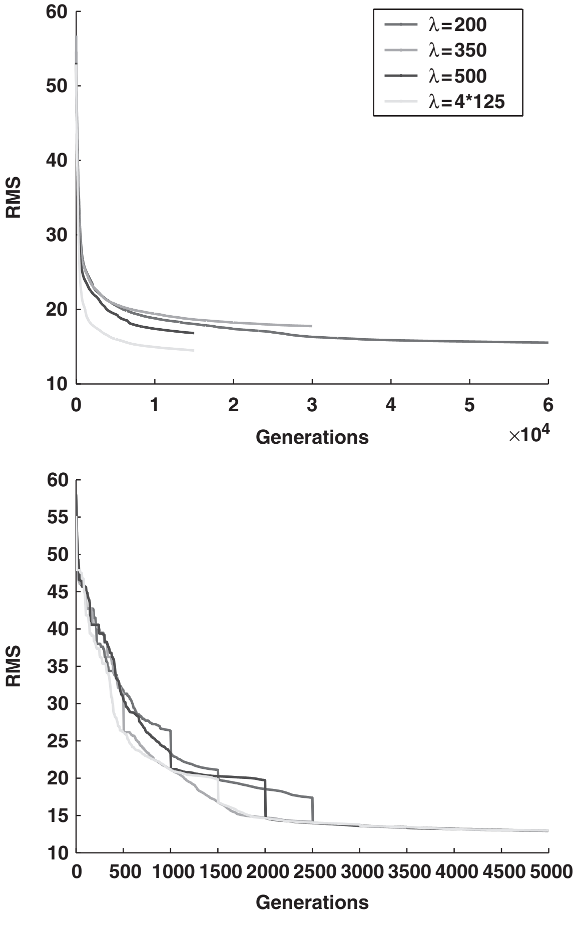


Fig. 5. Convergence behaviour of the fitness of (left) the average of 20 experiments (with $N=62$ and $h=-2.5$) for three different (μ, λ)-ES and the island (μ, λ)-ES and (right) the evolution of the fitness of the four subpopulations in the initial phase of a typical island-based (μ, λ)-ES run.

network (see also Supplementary Material, Figs 9 and 10 and Table 6).

One advantage of our method is that it is more reliable, i.e. the percentage of good solutions is larger than obtained by parallel simulated annealing: around 50% of the runs have a good solution quality compared to the 25% in Jaeger *et al.* (2004a, b). The island-based (μ, λ)-ES approach followed by DS even increases the ratio ‘good solutions’ to 62% using the same amount of work.

The most significant result of this work is the relatively small computational effort needed to reach a ‘good guess’ as starting point for the local search. Our method, (μ, λ)-ES followed by a local search, requires less computational time (8–11 CPU hours), and less resources (one 3.4 GHz processor) to achieve solutions as good as the one obtained with PLSA (between 8–160 CPU hours using 10 parallel 2.4 GHz processors), making our method 5–140 times as fast.

The test case in this study was a one-dimensional reaction-diffusion system with a large number (66) of parameters to estimate. In future work, we plan to infer GRN models for pattern formation in organisms where moving cells and deformable shape are essential features. Three-dimensional models will then be necessary and the number of parameters will increase substantially. Therefore, an efficient parameter estimation method will be mandatory.

ACKNOWLEDGEMENTS

We want to thank the anonymous referees for their many valuable suggestions. This work was supported by The Netherlands Organization for Scientific Research, project NWO-CLS 635.100.010 (‘3d-RegNet: simulation of developmental regulatory networks’). The work of Blom is in addition supported by the Dutch Bsik/BRICKS project. Website: http://www.science.uva.nl/research/scs/3D-RegNet/. We used data from the FlyEx database http://flyex.ams.sunysb.edu.

We modified the original gene circuit software available at http://flyex.ams.sunysb.edu/lab/download.html and also modified the C++ Direct Search software available at http://www.cs.wm.edu/~va/software/DirectSearch/direct_code/ which contains the downhill simplex used.

Conflict of Interest: none declared.

REFERENCES

Bäck,T. *et al*. (1997) *Handbook of Evolutionary Computation*. Institute of Physics Publishing and Oxford University Press, Bristol, UK.

Beyer,H.-G. (1996) Toward a theory of evolution strategies: self-adaptation. *Evol. Comput*., **3**, 311–347.

cantú-Paz,E. (1995) A summary of research on parallel genetic algorithms. *Technical Report IlliGAL No. 95007*. University of Illinois, Urbana-Champaign.

Chu,K.-W. *et al*. (1999) Parallel simulated annealing by mixing of states. *J. Comput Phys*., **148**, 646–662.

de Jong,H. (2002) Modeling and simulation of genetic regulatory systems: a literature review. *J. Comput Biol*., **9**, 67–103.

Foe,V.E. and Alberts,B.M. (1983) Studies of nuclear and cytoplasmic behaviour during the five mitotic cycles that precede gastrulation in Drosophila embryogenesis. *J. Cell Sci*., **61**, 31–70.

Fogel,L. *et al*. (1966) *Artificial Intelligence through Simulated Evolution*, John Wiley & Sons, New York.

Gilbert,S.F. (2006) *Developmental Biology*. 8th edn. Sinauer Associates, Sunderland, MA.

Goldberg,D.E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley: Boston, Ma, USA.

Gursky,V.V. *et al*. (2004) Pattern formation and nuclear divisions are uncoupled in Drosophila segmentation: comparison of spatially discrete and continuous models. *Physica D*, **197**, 286–302.

Holland,J.H. (1992) Genetic algorithms. *Sci. Am*., **267**, 66–72.

Hooke,R. and Jeeves,T.A. (1961) Direct search solution of numerical and statistical problems. *J. Assoc Comput. Mach*., **8**, 212–229.

Houchmandzadeh,B. *et al*. (2002) Establishment of developmental precision and proportions in the early Drosophila embryo. *Nature*, **415**, 798–802.

Jaeger,J. *et al*. (2004a) Dynamic control of positional information in the early Drosophila embryo. *Nature*, **430**, 368 – 371.

Jaeger,J. *et al*. (2004b) Dynamical analysis of regulatory interactions in the gap gene system of Drosophila melanogaster. *Genetics*, **167**, 1721–1737.

Janssens,H. *et al*. (2006) Quantitative and predictive model of transcriptional control of the Drosophila melanogaster even skipped gene. *Nat. Genet*., **38**, 1159–1165.

Katare,S. *et al*. (2004) A hybrid genetic algorithm for efficient parameter estimation of large kinetic models. *Comput. Chem. Eng*., **28**, 2569–2581.

Kolda,T.G. *et al*. (2004) Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Rev. Soc. Ind. Appl. Math*., **45**, 385–482.

Kosman,D. *et al*. (1998) Rapid preparation of a panel of polyclonal antibodies to Drosophila segmentation proteins. *Dev. Genes Evol*., **208**, 290–294.

Lam,J. and Delosme,J.-M. (1988a) An efficient simulated annealing schedule: derivation. *Technical report 8816*. Electrical Engineering Department, Yale, New Haven, CT.

Lam,J. and Delosme,J.-M. (1988b) An efficient simulated annealing schedule: Implementation and evaluation. *Technical report 8817*. Electrical Engineering Department, New Haven, CT.

Lewis,R.M. *et al*. (2005) Implementing generating set search methods for linearly constrained minimization. *Technical report WM–CS–2005–01*. Department of Computer Science, College of William & Mary, Williamsburg, Va, USA. Revised July 2006.

Lohmann,R. (1991) Application of evolution strategy in parallel populations. In Schwefel, H. P. and Männer, R. (eds.), *Parallel Problem Solving from Nature - Proceedings of 1st Workshop,* (*PPSN 1*), volume 496 of *Lecture Notes in Computer Science*. pp. 198–208, Springer-Verlag, Berlin, Germany.

Marnellos,G.E. (1997) *Gene Network Models Applied to Questions in Development and Evolution. Ph.D. Thesis*. Yale University, New Haven, Ct, USA.

Mendes,P. and Kell,D.B. (1998) Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. *Bioinformatics*, **14**, 869–83.

Mjolsness,E. *et al*. (1991) A connectionist model of development. *J. Theor Biol*., **152**, 429–453.

Moles,C.G. *et al*. (2003) Parameter estimation in biochemical pathways: a comparison of global optimization methods. *Genome Res*., **13**, 2467–2474.

Mühlenbein,H. *et al*. (1991) The parallel genetic algorithm as function optimizer. *Parallel Computing*, **17**, 619–632.

Myasnikova,E. *et al*. (1999) Spatio-temporal registration of the expression patterns of Drosophila segmentation genes. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, AAAI Press, Menlo Park, Ca, USA, pp. 195–201.

Myasnikova,E. *et al*. (2001) Registration of the expression patterns of Drosophila segmentation genes by two independent methods. *Bioinformatics*, **17**, 3–12.

Nelder,J. and Mead,R. (1965) A simplex method for function minimization. *Comput. J*., **7**, 308–313.

Perkins,T.J. *et al*. (2006) Reverse engineering the gap gene network of Drosophila melanogaster. *PLoS Comput. Biol*., **2**, e51.

Poustelnikova,E. *et al*. (2004) A database for management of gene expression data in situ. *Bioinformatics*, **20**, 2212–2221.

Reeves,G.T. *et al*. (2006) Quantitative models of developmental pattern formation. *Dev. Cell*, **11**, 289–300.

Reinitz,J. and Sharp,D.H. (1995) Mechanism of eve stripe formation. *Mech. Dev*., **49**, 133–158.

Reinitz,J. *et al*. (1998) Stripe forming architecture of the gap gene system. *Dev. Genet*., **23**, 11–27.

Runarsson,T.P. and Yao,X. (2000) Stochastic ranking for constrained evolutionary optimization. *IEEE Trans. Evol. Comput*., **4**, 284–294.

Runarsson,T.P. and Yao,X. (2005) Search biases in constrained evolutionary optimization. *IEEE Trans. Syst. Man Cybern. Part C*, **35**, 233–243.

Saltelli,A. *et al*. (2004) *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models*. Halsted Press, New York, USA.

Spears,W.M. *et al*. (1993) An overview of evolutionary computation. In *ECML '93: Proceedings of the European Conference on Machine Learning*. Vol. 667. pp. 442–459, Springer-Verlag, London, UK.

van Riel,N.A. (2006) Dynamic modelling and analysis of biochemical networks: mechanism-based models and model-based experiments. *Brief. Bioinformatics*, **7**, 364–374.

Wolpert,L. (1969) Positional information and the spatial pattern of cellular differentiation. *J. Theor Biol*., **25**, 1–47.