UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

Genetic regulatory networks inference : modeling, parameters estimation & model validation

Fomekong Nanfack, Y.

Publication date
2010

Link to publication

Citation for published version (APA):
Fomekong Nanfack, Y. (2010). *Genetic regulatory networks inference : modeling, parameters estimation & model validation*. [Thesis, fully internal, Universiteit van Amsterdam].

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:16 Feb 2026

4

Inference of the Gap gene [1]

4.1 Segmentation in *Drosophila melanogaster*

Drosophila melanogaster is a little diptera insect that belongs to the fruit flies family. It quickly became one of the main model organisms in biological research, after pioneering works on genetic by Thomas Hunt Morgan. *Drosophila melanogaster* small size, minimal nutritional requirement and short life cycle (approximatively two weeks) makes it an ideal organism to work with. Since 1998, its complete genome has been sequenced and the presence of occasional variants in natural population with easily recognisable distinct features makes *Drosophila melanogaster* exceptionally suitable to investigate theories and applied problems in developmental genetics. Its importance for human health was recognized by the award of the Nobel prize in medicine and physiology to Edward B. Lewis, Christiane Nüsslein-Volhard and Eric Wieschaus in 1995 [197, 198].

Most of our actual knowledge regarding the anterior-posterior body plan development of organisms is provided from genetic analysis of a series of mutations occurring in three classes of genes in *Drosophila melanogaster*: maternal genes, segmentation genes and homeotic genes [196]. These three classes of genes are responsible for the cellular specialisation of the developmental embryo. The entire process is mainly conducted in four steps: maternal position, gap domains formation, pair-rule domains formation and finally the stable boundary or segment polarity. At each step, a basic regulatory mechanism is

[1] This chapter is partially based on the paper:
Yves Fomekong-Nanfack and Jaap A. Kaandorp and Joke Blom, "Efficient parameter estimation for spatio-temporal models of pattern formation: Case study of Drosophila melanogaster", Bioinformatics, 23(24): 3356 - 3363, 2007. [78]

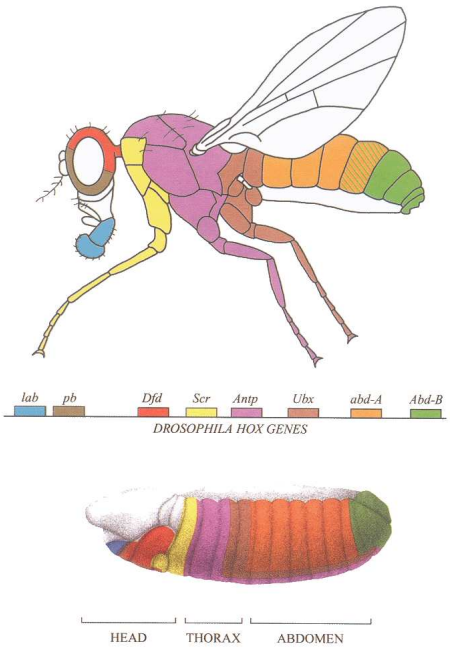


Figure 4.1: Eight Hox genes regulate the identity of regions in the adult (top) and embryo (bottom). The Hox genes are activated by the pair-rule genes but a subset of gap genes also influences directly the Hox genes. Picture taken from S.B. Carrol [38]

controlling the establishment of spatial expression genes also referred as "pattern formation" [52]. The compete development of the embryo is a cascade of successive segmentation. The body plan of *Drosophila melanogaster*, contains 14 segments as shown in Fig. 4.1, along the anterior-posterior axis, of which three establish the head (including its antennae and mouth). The next three segments constitute the thorax. Each of the thorax 'segments will make up a set of legs. Finally, the last eight segments will establish the abdominal.

4.1.1 Early segmentation of the anterior-posterior body formation

Contrarily to most organisms, the early *Drosophila melanogaster* embryo is not constituted of many cells, but instead, it is a single syncytium comprising a mass of cytoplasm and multiple nuclei (see Fig. 4.2) structure persists until successive rounds of nuclear division have produced some 1500 nuclei: only then do individual uninucleate cells start to appear around the outside of the syncytium, producing the structure called the blastoderm. Before the blastoderm stage has been reached, the positional information operated by maternal

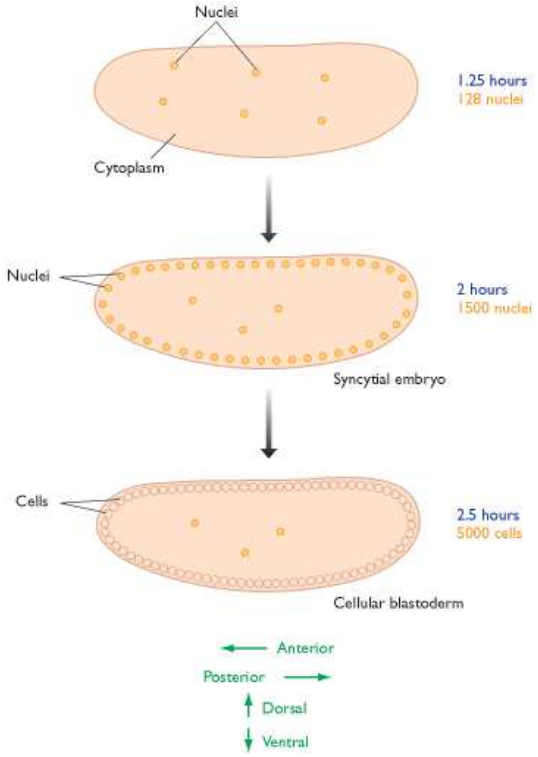


Figure 4.2: Early development of the *Drosophila melanogaster* embryo. At the early stage, the embryo is a single syncytium containing a gradually increasing number of nuclei. These nuclei migrate to the periphery of the embryo after about 2 hours, and within another 30 minutes cells begin to be constructed. The embryo is approximately 500 μm in length and 170 μm in diameter. Picture taken from A.T. Brown [31]

genes has begun to be established.

Step 1: Positional information by maternal genes Maternal genes determine the embryo's polarity. Initially the positional information that the embryo needs is a definition of which end is the front (anterior) and which the back (posterior), as well as similar information relating to up (dorsal) and down (ventral). This information is provided by concentration gradients of proteins that become established in the syncytium. The bulk of these proteins are not synthesized from genes in the embryo, but are translated from mRNAs injected into the embryo by the mother. There is four proteins or so called maternal-gene, involved in determining the A-P axis. These four proteins are:

- Bicoid (Bcd)
- Caudal (Cad)

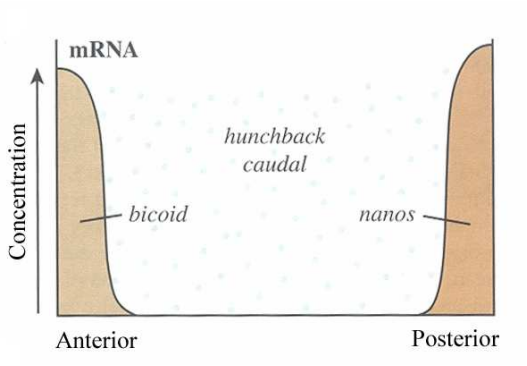


Figure 4.3: Maternal genes along the A-P axis. Picture taken from S.B. Carrol [38]

- Hunchback (Hb)
- Nanos (Nos)

bicoid (*bcd*) is transcribed in the maternal nurse cells, which are in contact with the egg cells, and the mRNA is injected into the anterior end of the unfertilized egg [64]. This position is defined by the orientation of the egg cell in the egg chamber. The bicoid mRNA remains in the anterior region of the egg cell, attached by its untranslated region to the cell's cytoskeleton. Bicoid protein then diffuses through the syncytium, setting up a concentration gradient, highest at the anterior end and lowest at the posterior end. The nanos mRNA is transported to the posterior part of the egg and attached to the cytoskeleton while it awaits translation. *Hb* and *cad* mRNAs become distributed evenly through the cytoplasm, but their proteins subsequently form gradients through the action of Bcd and Nos. The genetic regulation is described as follow:

- *bcd* activates the maternal hunchback gene in the embryonic nuclei and represses translation of the maternal caudal mRNA, increasing the concentration of the Hunchback protein in the anterior region and decreasing that of Caudal [229].
- Nanos represses translation of hunchback mRNA, contributing further to the anterior-posterior gradient of the Hunchback protein [86].

cad is both a maternal and zygotic gene, as well as *hb* [268]. The net result is a gradient of Bcd and Hb, greater at the anterior end, and of Nos and Cad, greater at the posterior end (see Fig. 4.3). In addition, two other genes control the terminal system of the embryo, *tailles,(tll)* and *huckebin ,(hkb)* [39, 223].

Gap gene The gradients established in the embryo by the maternal-effect gene products are the first stage in formation of the segmentation pattern.

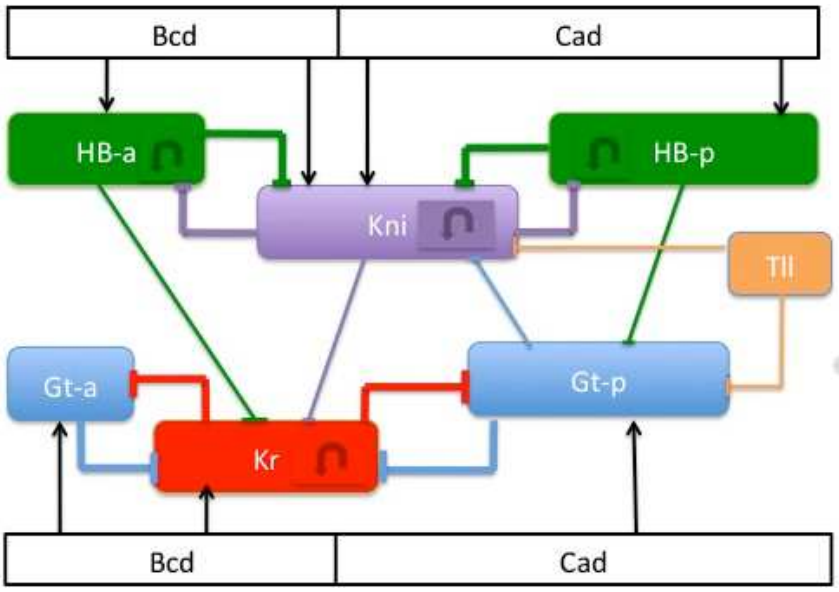


Figure 4.4: Gap gene network proposed by Jaeger et al. [120]. The figure illustrates the genetic network that defines the gap gene expression domains in the head, the trunk and the tail regions of the blastoderm embryo. Maternal morphogen Bcd activates both anterior and central gap genes while Cad activates *kni*, posterior *gt* and posterior *hb*. Hb, Kni and Gt have an autoactivation. Only repressive interactions are present among gap-gap genes. The width of the arrows determine the strength of the interactions. Inner arrows represent autoactivation. Arrows → indicate activation while T-connectors ⊣ represent repression. Image modified from Jaeger et al. [120]

These gradients provide the interior of the embryo with a basic amount of positional information, each point in the syncytium now having its own unique chemical signature defined by the relative amounts of the various maternal-effect gene products. This positional information is made more precise by the expression of the gap genes which are: zygotic *hunchback*, *Krüppel (Kr)*, *giant (gt)* and *knirips (kni)*. After the ninth divisions, they start to be zygotically expressed and are established at their maximum level at cleavage cycle 14 (see Section. 4.2.1). The maternal genes *bcd*, *cad* and *hb* regulate their transcription [229, 268] in two ways: *bcd* and *cad* activates their transcription while *hb* can either repress [268] or activate the anterior domain (in combination with bcd, [253]) their transcription. In addition to the regulation by maternal genes, experimental evidence [36, 54, 70, 268] suggest that interactions between the gap genes also contribute in the establishment of their precise expression further as shown in Fig. 4.4.

Pair rule The next set of genes to be transcribed, the pair-rule genes, establish the basic segmentation pattern. Transcription of these genes responds to the relative concentrations of the gap gene products and occurs in nuclei that have become enclosed in cells. The pair-rule gene products therefore do not diffuse through the syncytium but remain localised within the cells that express them. The result is that the embryo can now be looked upon as comprising a series of stripes, each stripe consisting of a set of cells expressing a particular pair-rule gene. In a further round of gene activation, the segment polarity genes become switched on, providing greater definition to the stripes by setting the sizes and precise locations of what will eventually be the segments of the larval fly. Gradually the imprecise positional information of the maternal-effect gradients are converted into a sharply defined segmentation pattern.

4.2 Gene circuit

As discussed above, many different developmental steps leading to pattern formation are taken place before the embryo segments are precise. In this dissertation we only focus on the gap gene segmentation. It is known that the maternal genes are responsible for the initial transcription of the gap gene. It is also acknowledge that the gap gene undergo some mutual repression, but their precise role is still unclear as well as the mechanism that control the dynamics behind the precise positional information. In 2004, Jaeger et al. [121] provided one of the first (if not the first) quantitative model that simulate the wild type gap gene expression patterns dynamics. The motivation was to understand the role of the gap-gap gene interactions. They have inferred a network by means of reverse engineering that can reproduce the measured spatial and temporal gene expression patterns. The gap gene model involves seven different genes, *bcd, cad, hb, gt, Kr, kni* and *tll*. The experimental data used to fit the model were obtained from the FlyEx database, where an extensive amount of accurate quantified spatio-temporal expression data for all genes is stored [189, 211]. A connectionist description was used to model the gene regulatory network.

Related work For modeling the segmentation mechanism of the early *Drosophila melanogaster* embryo, two main formalisms have been proposed: a logical formalism (tackling qualitative aspects) proposed by Sánchez and Thieffry [237], and continuous models proposed by Mjolsness and Reinitz [182] used to simulate the dynamics of a system. They have developed gene circuit data-driven mathematical modelling method whose main goal is to reveal hidden information about the dynamical mechanism of gene regulation. Using the latter formalism, because the detailed spatio-temporal data is available Reinitz and co-workers formulated the inference problem as an inverse problem using the connectionist model [224]. Given a mathematical model and sufficient accurate quantitative data, the parameters in the model can be estimated by optimization techniques, i.e. by fitting the model to the data. Except for box con-

straints, only little experimental information is used to constrain the parameter values in the model.

4.2.1 Quantitative gene expression data

This section gives more information on the data used in the course of this thesis to fit the model parameters for simulating the gene expression patterns of the *Drosophila melanogaster* in the early blastoderm stage. The lab methodology was not experimentally obtained by our group, but was developed by John Reinitz's group in Stony Brook and Maria Samsonova's group from Saint Petesbourgh. The overall result is presented as a numerical atlas of segmentation gene expression in the blastoderm containing quantitative spatial and temporal measurements of gene expression obtained from individual embryos and an spatial temporal average data representation [211]. Although not developed by us, for completeness, we describe the methodology used to obtain the quantitative spatio temporal gene expression data.

Acquisition of quantitative data Each *Drosophila melanogaster* blastoderm's embryo was collected, fixed and immunostained for three segmentation genes products as described by Kosman et al. [148]. Each embryo was fluorescently stained for Even-Skipped (*Eve*) protein and two other gene products.

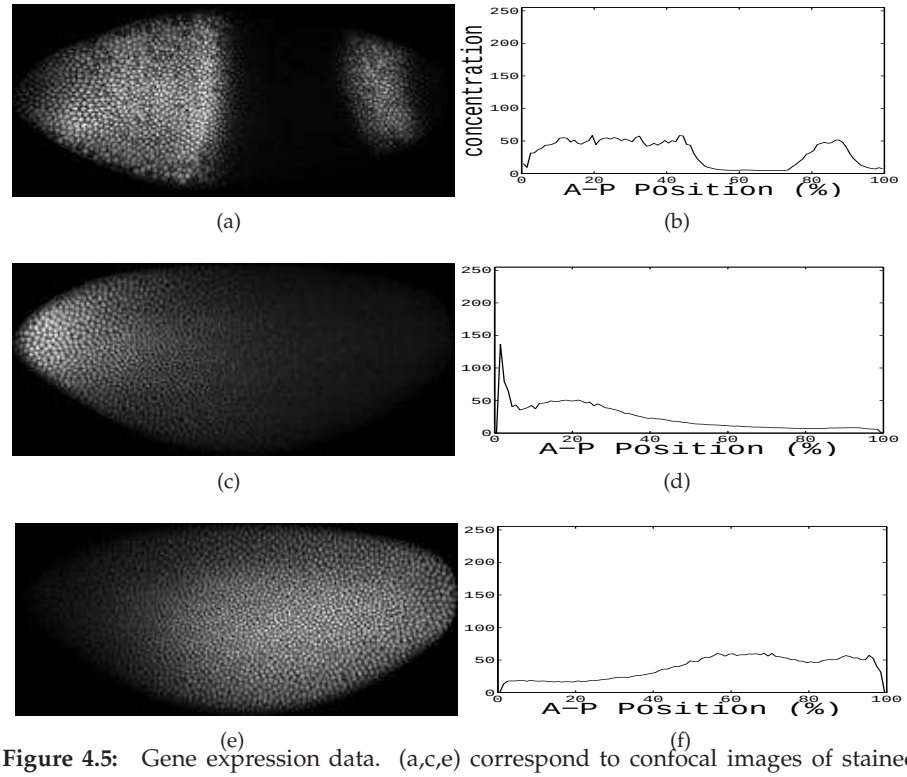


Figure 4.5: Gene expression data. (a,c,e) correspond to confocal images of stained *Drosophila melanogaster* blastoderm embryos. Staining is done by fluorescent immunohistochemistry [148]. (b,d,f) are the average quantitative gene expression levels obtained by successive image-processing operations [189, 191]. 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). 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. Images are from the FlyEx database `http://flyex.ams.sunysb.edu/flyex`. [211]

Image processing for quantitative data extraction This part was done in four major steps:

1. For normalization purposes, quantitative protein concentrations were all set to a relative fluorescence intensity range of [0,255] on the basis of the most intensely fluorescent pattern on each slide with multiple embryos. Based on an embryo mask, all embryo images were horizontally oriented along the Anterior-Posterior (A-P) axis and cropped to the dimensions of the mask. Image embryo were then all segmented to obtain exact nuclear positions.

2. Time classification was applied to each embryo to determine the corresponding cleavage cycle [189, 191]. A cleavage cycle is the development period between two mitosis. At the early blastoderm stage, several cleavage cycles occur. Foe et Albert [76] have determined the exact duration of these cycles (8 to 12 minutes for cycle 10 to 13 and approximatively 50 minutes for cycle 14A). Between each cycle, a nuclear division takes place preceded by a mitosis. From cycle 10 to 14, the average number of nuclei follows: 130, 260, 450, 1000, 2000. Embryo's age below cleavage cycle 14 was then established based on their average number of nuclei in the transverse section. Since during cleavage cycle 14 there is no mitosis, the duration is long and the gene expression level changes considerably, additional benchmark time were included (T1-T8) and embryos timing was done carefully based on the *eve* gene pattern and nuclei morphology [192]. Fig. 4.6 illustrates the time schedule. The image is modified from [120] where more details on the gap-circuit are given

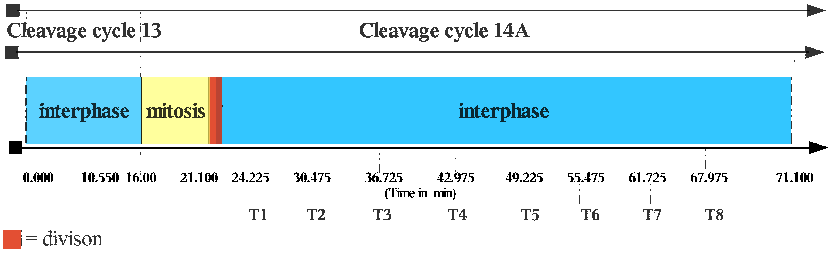


Figure 4.6: The model simulates cleavage cycle 13 and cleavage cycle 14A until gastrulation at time t = 71.10 min. Interphase takes places at the beginning of the process from time t = 0.00 until t = 16.00 min, followed by mitosis, occurring from t = 16.00 min until t = 21.10 min. *Image is modified from [120] where more details on the gap-circuit are given.*

3. Background removal and data registration were applied to transform the different sets of data into one coordinate system. This is necessary in order to be able to compare or integrate the data obtained from different

measurements. Background removal done by Myasnikova et al. [193] aims at removing signals that are due to non-specific binding of the antibodies. Data registration [189] goal is to map (A-P and D-V position) the patterns obtained from individual embryos for averaging purpose. Embryos were aligned on the basis of *eve* patterns features.

4. Integrated data were obtained by averaging the middle 10% of dorsoventral (D-V) positional values of each embryo for each gene and time class.

The final data contain gene products for: *bicoid* (*bcd*), *caudal* (*cad*), *hunchback* (*hb*), *Krüppel* (*Kr*), *knirps* (*kni*), *giant* (*gt*), and *tailless* (*tll*). This results in a database of integrated data. Quantitative gene expression data are not always complete for all genes at all time points. For instance, *tll* is not available at $t = 10.55$ min, T1 and T2. The gene product for *cad* is also not available at times T7 and T8. Initial conditions are given by data obtained at cleavage cycle C12. The maternal contribution of *bcd* stays constant during the whole process. Fig. 4.7 shows the gene expression pattern of the integrated real data used in the current study.

The integrated data on the expression of 14 genes presented here were assembled from many individual isogenic embryos, each stained for the products of three genes. Even in an isogenic population, there are differences between individuals. The fundamental criterion for the validity of our integrated data is that it should represent the possible actual dynamics of one individual in the isogenic population.

4.2.2 Connectionist model of the Gap Gene

The pattern formation at the early stage of the *Drosophila melanogaster* blastoderm results from the interactions among segmentation genes, by affecting the gene expression of other segmentation genes. At this stage, a *Drosophila melanogaster* embryo consists of a syncytium containing nuclei not surrounded by a membrane. The developmental time of interest is between cycle 13 and 14A, before gastrulation at time $t = 71.10$ min (see Fig. 4.6). To simulate the pattern formation, we use the model given in Equation (4.1) [121] based on a connectionist model [182]. It is a dynamical model consisting of a discrete representation in space of the nuclei with discrete cell division and a continuous regulation of the genes in time. For each nucleus, a system of lattice differential equations describes the change in concentration of gene products:

$$\frac{dg_i^a}{dt} = \underbrace{R_a \Phi_a \left(\sum_{b=1}^{N_g} W_a^b g_i^b + \sum_e m_a^e g_i^e + h_a \right)}_{\text{regulation}} \underbrace{-\lambda_a g_i^a}_{\text{decay}} \underbrace{+D_a \left(g_{i+1}^a - 2g_i^a + g_{i-1}^a \right)}_{\text{diffusion}} \quad (4.1)$$

where a and b denote gene products and i the nucleus number. In Equation 4.1 gene product concentrations depend on three main factors:

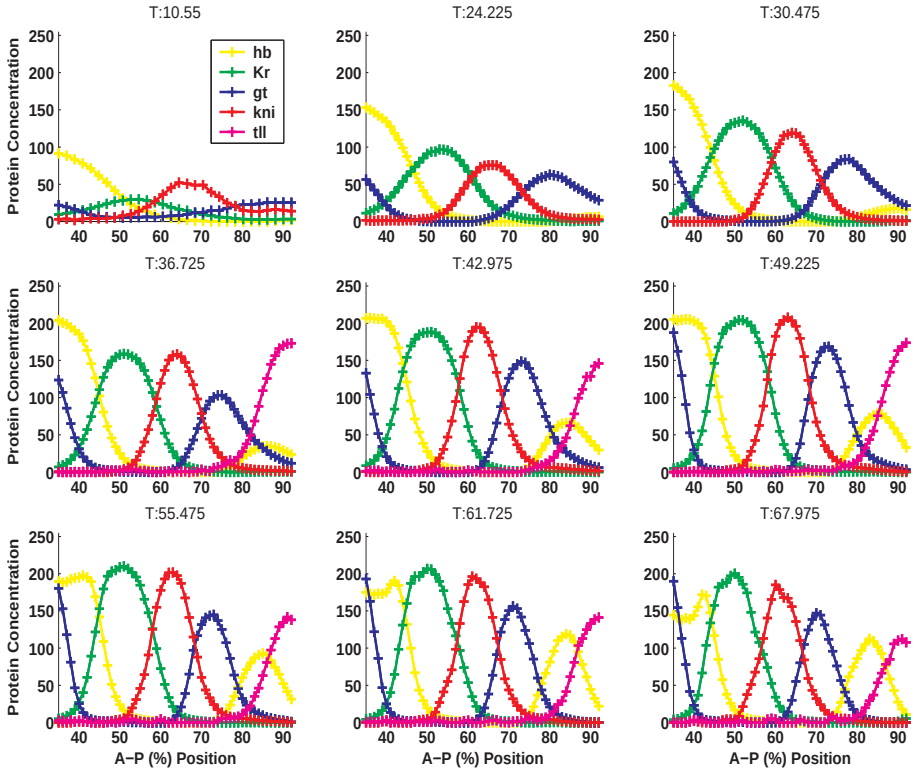


Figure 4.7: Observed expression of *hb, Kr, gt, kni, tll* used to fit the simulated gap-gene expression. Time point $t = 10.55$ shows the gene expression at cleavage cycle 13 where only 30 nuclei are present. The other 8 images show the 8 time points of cleavage cycle 14A after division with 58 nuclei in each. The y-axis gives the relative protein concentration expression level. The x-axis corresponds to the anterior-posterior (A-P) axis of the embryo. The integrated data intends to represent the possible actual dynamics of one individual embryo in the isogenic population. There exists an amplitude variation and positional variation from embryo to embryo. Although It is not possible to quantitatively measure the expression of all the genes involved, the integrated data seems to be a well representation of individual embryo gene expression. Surkova et al. [263] compared the integrated data with individuals data and concluded that the gene means from individual pattern is very close to the median individual pattern.

- The first term describes the regulation of protein synthesis that takes place in the nucleus. The genetic regulation model is the same in each nucleus. This is represented by the weight matrix W. W_a^b characterizes the regulatory effect of gene b on gene a. N_g is the total number of zygotic genes in the model. The sum over e denotes the external influence from maternal genes such as *bcd* or *cad*. In this paper only *bcd* is present as maternal gene. Φ is a sigmoid function with range $(0, 1)$ to prohibit negative

values resulting from inhibitors, and to saturate the effect of activators [224]. h_a denotes the shift for the transition of Φ from 0 to 1 given as:

$$\Phi_a(u^a) = \frac{1}{2}\left[\left(u^a/\sqrt{(u^a)^2+1}\right)+1\right] \tag{4.2}$$

where $u^a = \sum_{b=1}^{N_g} W_a^b g_i^b + m_a g_i^{bcd} + h_a$. The unknown parameters of the model are: the regulation matrix W_a^b, 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 of maternal gene *bcd* m_a. The model simulates the time evolution for the concentration of the genes *cad, hb, Kr, gt, kni, tll*.

- The second term is the decay of the gene products. The decay rate λ_a is related to the protein half-life of the product of gene a by $t_{1/2}^a = \ln(2)/\lambda_a$.
- The third term represents the exchange of diffusible products between neighboring nuclei.

4.2.3 Numerical implementation of the model

The developmental time of interest is between cycle 13 ($t = 0.00$ min) and 14A, before gastrulation at time $t = 71.10$ min as given in Fig. 4.6. Gap gene proteins appear only at cycle 13 and maternal genes *bcd,cad* and maternal *hb* are already present, initial conditions for the gap genes are all set to 0 and initial conditions for the maternal genes are taken from data at cycle 12 [211].

Three different rules describe the phenomena that occur during this time: interphase, mitosis and division. Interphase and mitosis are continuous rules describing the spatio-temporal evolution of gene expressions, while division is a discrete rule that gives the number of nuclei at a time point. Division is modeled by duplicating instantaneously all nuclei and halving the distance between them. The diffusion coefficient depends on the number of nuclear division that occurs before the current time t. It is assumed to vary inversely with the square of the distance between neighboring nuclei and this distance is halved upon nuclear division.

The two continuous rules cannot be solved analytically because of their dimensionality and complexity, in which case we have to resolve the problem by an approximating the exact solution. The general form of the continuous rules is given as a system of ordinary differential equations of the form:

$$y'(t) = f(t, y(t)), \qquad y(t_0) = y_0 \tag{4.3}$$

where f is some known function of the state vector of the system and the initial condition y_0 is a given as a vector. ODEs /PDEs can be stiff and ill-conditioned, resulting in non-unique, non-existing, or non-reproducible modeling solutions.

Consequently, one has to carefully choose an intelligent optimal numerical solver. There is an extensive number of numerical solvers.

Since a numerical solution is just an approximation of the real solution, an accurate numeric solver should reduce as much as possible the difference between the real solution and the approximation. To efficiently choose the most suitable numerical solver for the equations given in Equation 4.1, a number of properties has to be checked such as: constant or variable time step, one-step or multistep, explicit or implicit, low-order or high-order. One important aspect of a numerical solver is the time-step or step size h. A very small time step is preferable but it will slack the solver. In the current problem, we are confronted with an optimisation problem: i.e the ODEs are solved an unlimited number of time. Therefore the solver should be fast while still being accurate. However, fast solver might require a larger time-step that might lead to an unstable problem. A numerical unstable problem is a problem for whom the equation include some terms that can lead to rapid variation in the solution. In this case, the ODEs are said to be stiff. Summing all these considerations, the chosen numerical solver has to deal with the potential stiffness of the ODEs, it has to be accurate and computationally efficient. The main considerations to choose a numerical solver are:

Constant or variable time Simple numerical solvers use a fixed constant time step h. Nowadays, most numerical solvers use adaptive time step during the course of the computation, mainly to maintain a consistent level of accuracy. The step size may change many times during the course of the computations, as larger time steps are used where the solution is varying slowly, and smaller steps are used where the solution varies rapidly.

One-step or multistep Numerical solvers require to know at least the previous step to compute the current solution. In this case, one talks about one-step algorithms. In contrast to one-step methods, multistep algorithms require the $k + 1$ previously computed solution values to compute the next solution. They are less sensitive to initial conditions and require fewer evaluation of the f per time step h. However, multistep algorithms are often slower because of differences in accuracy and computational complexity.

Explicit or implicit Explicit solvers require the current state of the system at t to compute the solution at $t + \Delta t$. Implicit algorithms such as the backward Euler method contain algebraic formulas that need to be solved using iterative processes like Newton iteration. One limitation of implicit methods is the non-guarantee of convergence, since it heavily depends on the termination criteria. However, these algorithms are very suitable for stiff problems because of their iterative form.

Low-order vs. high-order All numerical solvers compute their approximation of the solution in a finite number of values of the function f. The error that arises in this approximation is called truncation error. For constant time step algorithms, the maximum truncation error when solving a differential equation over a fixed time interval $t_0 \leq t \leq T$ is proportional to the time step h^p with h being the time step and p the order of the method. Therefore, a high-order method is more efficient than a low-order algorithm in most practical problems. However, if the solution is not smooth enough, a high-order algorithm will not be very accurate and a tradeoff between the order and the computational cost should be consider.

Comparisons of numerical solvers Based on literature reviews [7, 12, 139, 214, 215, 260] and the criteria mentioned above, we have compared different numerical solvers in order to decide which one should be appropriate to the gene circuit. We have tested the following methods:

1. Adams: explicit multistep algorithm
2. Euler: explicit one step algorithm
3. Heun: explicit multistep method with additional predictor an corrector steps to euler solution
4. Bulirsch-Stoer: implicit adaptive step-size algorithm.
5. Bader-Deuflhard: semi-implicit mid-point algorithm for stiff systems of ODEs
6. Runge-Kutta 4: explicit single step method

Table 4.1 shows the comparison of the five different numerical solvers. The comparison is made on three main specific performance measure such as computational cost, accuracy and convergence. All methods converge while we lower the accuracy or step size, but most of them have the computational time that considerable increases. Since we want the CPU time to be relatively low, from the table, we see that only Bulirsch-Stoer achieve a very good score in a relative small computational time with a satisfactory accuracy. On this basis, we have chosen to use Bulirsch-Stoer for the rest of the simulations in the course of this thesis. This solver is an adaptive step-size algorithm based on a modified midpoint method and the Richardson Interpolation and Extrapolation [59, 214] As shown in Tab. 4.1, it is a lot cheaper than the Runge-Kutta method, while at the same time, offering a better stability than the Euler method. Since the function evaluation is not very expensive, we did not opted for a predictor-corrector method (Adams-Bashforth-Moulton).

4.2.4 Optimisation

As in the previously mentioned *Drosophila melanogaster* studies by Jaeger et al. [121, 206], we have chosen to use as cost-function the least-squares of the

Solvers comparison			
Name	Acc/Step	Time (ms)	RMS
Adam	1.0	3.07	9.73
	0.1	21.52	9.72
	0.01	204.23	9.72
	0.01	31.69	9.72
Euler	1.0	0.976	14.11
	0.1	9.01	9.78
	0.01	88.65	9.72
	0.001	886.473	9.72
Heun	1.0	1.84	10.12
	0.1	18.22	9.72
	0.01	179.43	9.72
	0.001	803.60	9.72
Bulirsch-Stoer	1.0	4.634	9.81
	0.1	4.248	9.71
	0.01	5.154	9.72
	0.001	6.13	9.72
Bader-Deuflhard	1.0	847.91	17.31
	0.1	847.05	9.77
	0.01	115.73	9.72
	0.001	834.35	9.72
Runge-Kutta 4	1.0	3.83	9.74
	0.1	36.34	9.72
	0.01	360.70	9.72
	0.001	621.44	9.72

Table 4.1: Comparison of numerical solvers on a gap gene circuit. Explicit and implicit numerical solvers are presented on the first column and their setting and performance are given in the next columns. The main setting is the accuracy for adaptive stepsize algorithm and the stepsize for fixed-stepsize (column 2). Comparisons are made on the computational time require to achieve a reasonable score. All the simulations are performed on the same circuit. Almost all solvers lead to a relative low score (RMS, discussed in Section 4.2.4), but the Bulirsch-Stoer method is the most stable with a more or less constant CPU time independently of its accuracy. Tests were performed on a serial 3.4-GHz "Intel Xeon" processor.

difference of the simulated and the observed data to which a constraint- or penalty function is added:

$$E(\theta) = \sum_{i,t} (g_i^a(t,\theta)_{model} - g_i^a(t)_{data})^2 \tag{4.4}$$

$$E_{tot}(\theta) = E(\theta) + E_{penalty}(\theta) \tag{4.5}$$

where $g_i^a(t)$ 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. 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 Π_{u^a} is used [224], resulting in:

$$E_{penalty} = \Pi_{R_a} + \Pi_{\lambda_a} + \Pi_{D_a} + E_{\Pi_{u^a}} \tag{4.6}$$

where the first three terms represent functions with value zero when the respective parameters are within the search limit (Table 4.2) and infinite otherwise. If for any non-regulatory parametera (R, λ, D) the value is out of the search space given in Table 4.2, the penalty is extremely high (or infinity). The last term $E_{\Pi_{u^a}}$ gives the penalty on the search space of the regulatory inputs in order to limit the saturation of u^a in the sigmoid function given in equation 4.2 with

$$\Pi_{u^a} = \sum_{ab} (W_a^b v_{max}^b)^2 + (m_a v_{max}^{bcd})^2 + (h_a)^2 \tag{4.7}$$

$$E_{\Pi_{u^a}} = \begin{cases} \exp(\Lambda \Pi_{u^a}) - exp(1) & \text{iff } \Lambda \Pi_{u^a} > 1 \\ 0 \qquad \text{otherwise} \end{cases} \tag{4.8}$$

where v_{max}^b and v_{max}^{bcd} are the maximum values for gene b and *bcd* found in the database. The parameter Λ controls the size of the search space for parameters involved in the sigmoid Φ function given in Equation (4.2). The goal of the penalty function Π_{u^a} is to limit the maximum saturation of u^a to $(1 - \Lambda)$ with Λ a small parameter (in this study taken to be 0.001). When $\Pi_{u^a} > 1/\Lambda$, the penalty $E_{\Pi_{u^a}}$ will be extremely high. This implies that the parameter are taking the dynamics out of the regulated region $\Phi(u^a)$.

We use the root mean square (RMS) [224] as a measure of the quality of a model solution for a given set of parameters:

$$\text{RMS} = \sqrt{\frac{E_{tot}(\theta)}{N_d}} \tag{4.9}$$

where $E_{tot}(\theta)$ is given by Equation (4.5) and N_d is the number of data points.

parameters	units	search space
R^b	min^{-1}	[10.0, 30.0]]
D^{bl}	min^{-1}	[0.0, 0.3]
$t_{1/2}^b$	min	[5, 20]

Table 4.2: Parameter search space for the Gene Circuit based on [120]. We have enlarged the parameter search space for all parameters with an explicit limit.

4.3 Reverse-engineering the gap gene

The purpose of the model presented in Equation 4.1 is to simulate the pattern formation of the early*Drosophila melanogaster* embryo. The aim of the optimization is to find suitable model parameters that can simulate realistic patterns, in comparison with real quantified gene expression patterns. Different settings for (μ, λ) – ES are used followed by direct simplex search. The selected results are chosen based on the quality of the fit (RMS) and visual comparison of the simulated pattern and the quantitative data.

4.3.1 Comparison of different ES settings

Different settings for (μ, λ)-ES are used followed by Downhill simplex local search (see Sections 3.1, 3.3). The population size λ is varied, in ES λ = $\{200, 350, 500\}$ and in the island ES with 4 subpopulations λ = $500/4$ = 125. The other method parameters are in all cases μ = $\lambda/5$, γ = 0.85, and α = 0.2 [234]. In all settings 20 optimisation 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 Downhill simplex is applied to each resulting gap gene circuit and runs for 130000 iterations. All simulations are performed on a serial 3.4-GHz "Intel Xeon" processor and took 8–11 CPU-hours for the complete ES+DS search.

Although some of the circuits with a RMS in $(12.00, 14.00)$ could reproduce faithfully the gene expression patterns, we only focus on those with a RMS ≤ 12.00. Out of 240 simulations, 125 ES+DS runs have a RMS ≤ 12.00 representing 52% good solutions. In Fig. 4.8 we have visualized the results and Table 4.3 summarises the statistical differences of the different ES settings.

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 a RMS smaller than 12 and did not show any specific defect.

Reduced Search In this setting the 20 optimisations 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 [121]. 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 optimisation problem as can be judged from the fact that 16 out of the 80 runs result in a RMS less than 12 after the ES. Also the advantage of using the island search can be seen more clearly: 16 out of 20 runs result in a RMS less

than 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 Fig. 4.8 (3 &2) the results are comparable with the -3.5 setting.

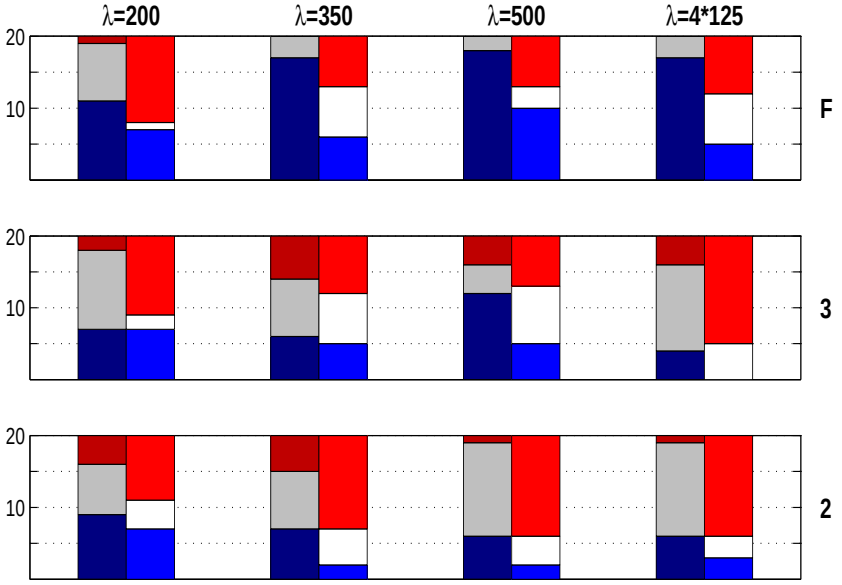


Figure 4.8: Comparison of the different optimisation runs for (F): Full Search, (3): Reduced Search with activation thresholds set at -3.5 and (2): Reduced Search with activation thresholds set at -2.5. Each bar-column represents 20 runs of a setting. Duo bar-columns are read as follows: Left: after ES, right: after DS; bottom bar (blue): $RMS > 14$, middle (white-grey): $RMS \in (12, 14)$, top (red): $RMS \leq 12$.

RMS	$\lambda = 200$ $N = 60000$			$\lambda = 350$ $N = 30000$			$\lambda = 500$ $N = 15000$			$\lambda = 4 * 125$ $N = 15000$		
	-2.5	-3.5	F	-2.5	-3.5	F	-2.5	-3.5	F	-2.5	-3.5	F
> 14	9/7	7/7	11/7	7/2	6/5	17/6	6/2	12/5	18/10	6/3	4/0	17/5
$(12, 14)$	7/4	11/2	8/1	8/5	8/7/	3/7	13/4	4/8	2/3	13/3	12/5	3/7
≤ 12	4/9	2/11	1/12	5/13	6/8	0/7	1/14	4/7	0/7	1/14	4/15	0/8

Table 4.3: Comparison of the results for 12 different settings. Twenty random seeds were generated and each configuration in a setting uses one of theses seeds. This results in 240 simulations. In all simulations, $\mu = \lambda/5$, $\gamma = 0.85$ and $\alpha = 0.2$. The $\lambda = 4 * 125$ is an island based ES with 4 sub-populations of each $\lambda = 125$. N is the number of generations of the ES. In all simulations, DS was run with 130000 iterations unless stopped before because no improvement was possible. The table header -2.5, -3.5 represents the fixed value of the promoter thresholds in the 62-dimensional case. F indicates a full search with 66 parameters to estimate. In each cell with value given as a/b, a is associated to the RMS after ES and b is associated to the RMS after DS.

In Appendix 8.4, the tables [8.2 – 8.5] gives details of the different ES setting scores.

λ	# after ES	percentage	# after DS	percentage
200	7	11.66	32	53.33
350	11	18.33	28	46.66
500	5	8.33	28	46.66
4*125	5	8.33	37	61.66

Table 4.4: In all different λ settings, 60 simulations were run. The island-ES followed by DS shows significantly better results than the simple ES combined with DS.

Visual comparison In all cases where a RMS smaller than 12 was obtained the simulated patterns match nicely the real spatio-temporal data (see Fig. 4.9 for an example). As in [121], in some other cases there is a small defect, especially for the late and posterior *tll* concentration.

4.3.2 Convergence of ES and Island based ES

In Fig. 4.10 we illustrate the convergence behaviour of the evolution strategy. In the left plot the average fitness evolution is given for the 20 optimisation 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 4 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 which might be stagnating, hence the staircase behaviour of the fitness curves (Fig. 4.10, right plot).

The four plots shown in Fig. 4.11 illustrate the convergence behaviour of all the different ES settings. All curves show a typical behaviour of a (μ, λ)-ES. In all cases, the fitness decreases quickly during the first generations. In Fig. 4.10, the right plot shows the convergence of 4 sub-populations of an island-based ES. After every 500 generations, migration is applied and some relative improvement can be observed in a sub-population receiving an individual with a better score than the actual best. This results in a sudden steep drop of the curve. Stagnation occurs when all sub-populations start to be homogeneous. The four sub-populations return model parameters with very small differences and very similar solution quality. The lower plot illustrates how the Downhill simplex can efficiently improve the solution after ES by reducing the RMS from 18.62 to 10.17. The best solution was used as input for the DS.

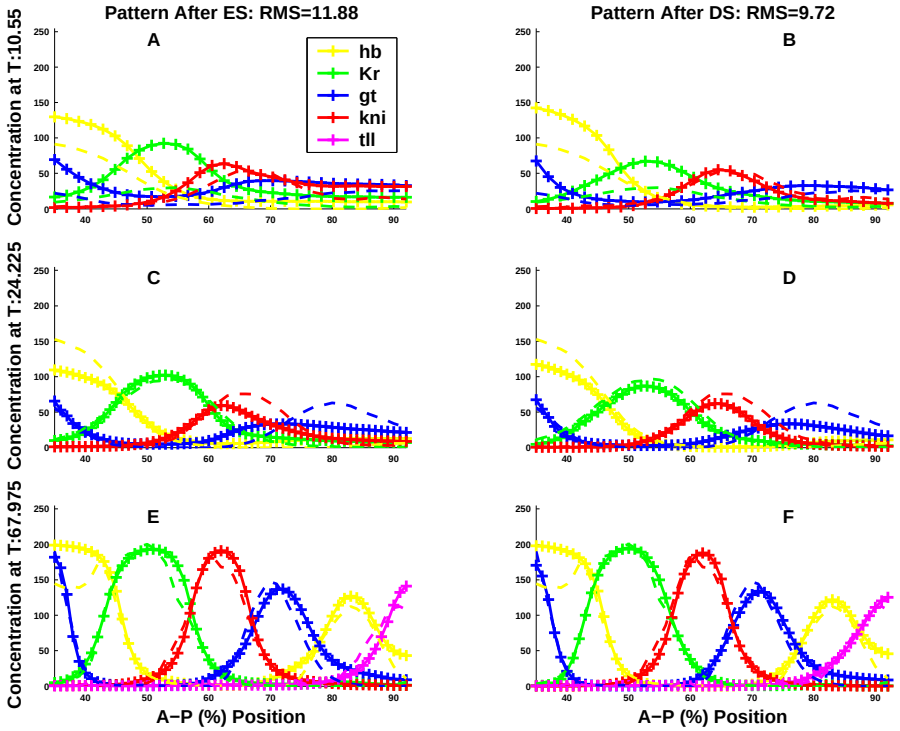


Figure 4.9: Solution of the gap-gene circuit gn52c13_200_62_25_14 at time points $T = 10.550$ and, after division, $T1 = 24.225$ and $T8 = 67.975$ obtained after parameter estimation using $(40, 200)$-ES (left) followed by Downhill simplex local search (right). Experimental (target) data is indicated with dashed lines.

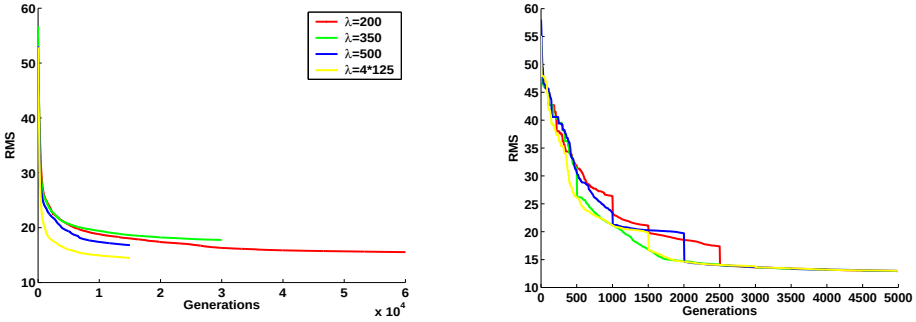
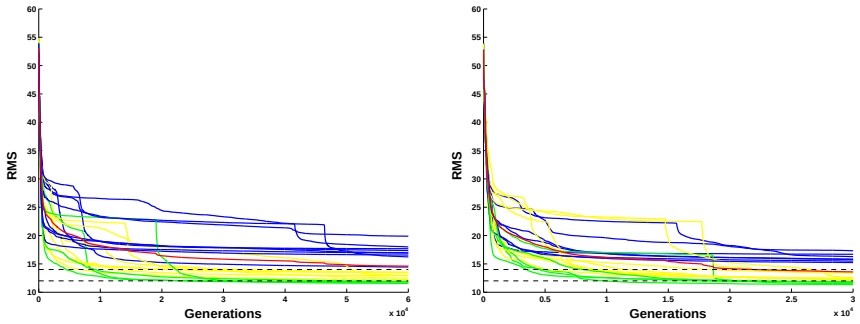
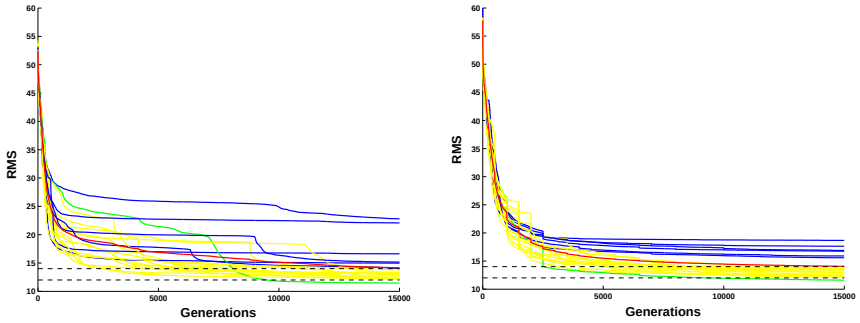


Figure 4.10: Convergence behaviour of the fitness of (left) the average of 20 experiments (with $N = 62$ and $h = -2.5$) for three different (μ, λ)-ES and the island (μ, λ)-ES and (right) the evolution of the fitness of the 4 subpopulations in the initial phase of a typical island-based (μ, λ)-ES run.



(a) $\lambda = 200$, $N = 62$, $h = -2.5$ (b) $\lambda = 350$, $N = 62$, $h = -2.5$



(c) $\lambda = 500$, $N = 62$, $h = -2.5$ (d) $\lambda = 4 * 125$, $N = 62$, $h = -2.5$

Figure 4.11: Convergence behaviour of (μ, λ)-ES (a,b,c) and (μ, λ)-ES island-based (d) for the 62-dimensional search with $h = -2.5$. (a) corresponds to a $(40, 200)$-ES, (b) is $(70, 350)$-ES , (c) is $(100, 500)$-ES and (d) island-based $4 * (25, 125)$-ES. Blue, yellow and green curves are respectively curves with a RMS ≥ 14.00, RMS $\in (12.00, 14.00)$ and RMS $\leqslant 12.00$ after a serial or an island (μ, λ)-ES. In (a,b,c) the red curve is the average of 20 runs and in (d) blue, yellow, and green give the fitness of the best sub-population; red is the average of 20 runs and 4 sub-populations.

Combining global and local search

Following the idea that heuristic search can not 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 (see Voogd et al. [281]) Simple (μ, λ)-ES could almost always find gap circuits with a RMS between 11.00 and 16.00 in an average of 8–11 CPU hours. As shown in Fig. 4.12-top, 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 as shown in Fig. 4.12-bottom.

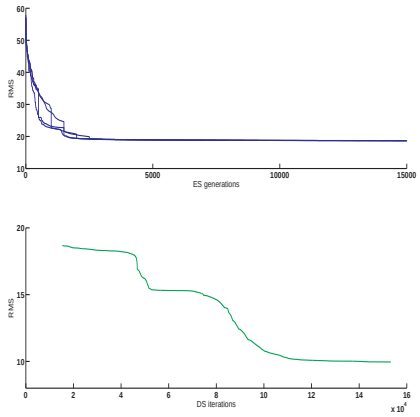


Figure 4.12: Convergence behaviour of an island-based ES followed by Downhill simplex. The upper plot shows the fitness evolution of the 4 sub-populations. The lower plot shows how the DS can improve the resulting ES solution. The RMS decreases from 18.62 to 10.17.

4.4 Discussion

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 circuits with a good RMS (≤ 12). This percentage is better than obtained by simulated annealing, as discussed in [120, 121] 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 Downhill simplex local search, 62% of the runs result in gap gene circuits with a RMS smaller than 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% vs. 60%, cf Fig. 4.8. 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.

Improvement of previous results

Jaeger et al. [120, 121] presented 10 gap-gene circuits including *bcd, cad, hb, Kr, gt, kni, tll* gene expression and covering a range of 35-92% of the A-P axis.

These ten gap gene circuits were selected among 40 results according to their RMS (≤ 12). Their results were obtained using a parallel Simulated Annealing (PLSA) method, and the computational time needed was between 8h and 160h using 10 2.4-GHz processors for each simulation.
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 solution quality compared to the 25% in [120,121]. 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. Recently, Jostins et al. [129] implemented a parallel island-ES. The compare the performance of the algorithm with the PLSA on reverse engineering of the gap gene. They show that parallel ES is significantly faster than PLSA and statistically, leads to a larger number of circuits with low score.

4.5 Conclusions

In this chapter, we have presented a brief biological background of early segmentation mechanism of *Drosophila* melanogaster. Interested in the gap gene segmentation and the formation of the spatio temporal pattern, we have presented the gene circuit developed by Reinitz et al. [224] which is a reverse-engineering approach that permits a dynamical representation of a dynamical system. Instead of using a parallel simulated annealing like elsewhere [121, 222, 224], we have used evolution strategy (ES) in combination with direct search methods presented in Chapter 3 to estimate the unknown parameters. The choice of this method was motivated by results obtained by Moles et al. [184] where the authors have compared different stochastic algorithm on a benchmark problem of 36 parameters and have showed that only a certain type of stochastic algorithm (ES) was able to solve the test problem successfully. Most stochastic methods cannot guarantee global optimality and present rather slow convergence rate, particularly in the final stage of the search. In order to surmount these difficulties, following Voogd et al. [281], we have used the direct search method to improve the solution when it was around a global minima. The ES termination criteria was set to be the number of iterations. As shown in Figs. 4.11 and 4.12, the algorithm quickly drops to a relative low score and stagnates for a considerable high number of iterations. Effectively, we observe that the fast convergence during the first iterations only represents $1/3$ to $1/5$ of the total time computational time. It will be therefore important to detect the stagnation stage in order to automatically switch to the local search. This would considerably reduce the total number of iterations. Nevertheless,

the overall time of the hybrid- algorithm is still lower than the PLSA used by Jaeger et al. [121]. We have demonstrated that our method, (μ, λ)-ES followed by a Downhill simplex search, gives solutions comparable to their solutions in terms of the RMS and in simulation results. This chapter only presented the computational results of the optimization in terms of effectiveness. In the following chapters, we will present and analyze the circuits and their associate simulated profile and investigate their robustness properties.