INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Modeling Genetic Networks To Aid in Understanding Their Function

by

Eli Meir

A dissertation submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2003

Program Authorized to Offer Degree: Zoology

UMI Number: 3079237

UMI Microform 3079237

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

In presenting this dissertation in partial fulfillment of the requirements for the Doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of the dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to Bell and Howell Information and Learning, 300 North Zeeb Road, P.O. Box 1346, Ann Arbor, MI 48106-1346, or to the author.

Signature 

Date 1/13/03

University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

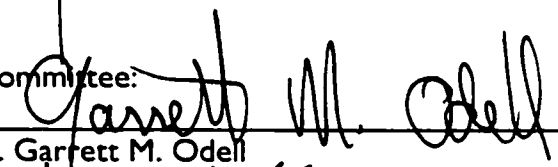
Eli Meir

and have found that it is complete and satisfactory in all respects, and that any and all revisions required by the final examining committee have been made.

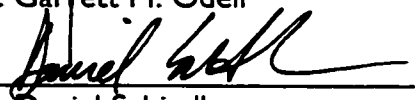
Chair of Supervisory Committee: [add additional signature lines as needed]

Dr. Garrett M. Odell

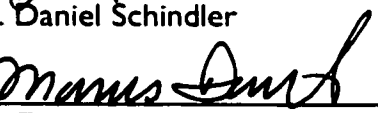
Reading Committee:



Dr. Garrett M. Odell



Dr. Daniel Schindler



Dr. Thomas Daniel

Date: 1/13/03

University of Washington
Abstract

Modeling genetic networks to aid in understanding their function

by Eli Meir

Chairman of the Supervisory Committee

Professor Garrett M. Odell
Zoology Department

Data on genetic interactions is accumulating at a very high rate. Here I introduce a new tool for analyzing this data at the level of genetic networks. The software program I helped write, called Ingeneue, lets the user take a picture of a genetic network and quickly turn this into a set of ordinary differential equations. It can then search for parameters of these equations that allow the network to produce specified patterns through time or space. I discuss some of the advantages of Ingeneue over other tools. I then discuss an exploration of the neurogenic network in *Drosophila* using Ingeneue. I find several interesting network level properties of the modeled neurogenic network. Among these are that it is highly robust to parameter variation, and that this variation leads to an evolutionary flexibility in the patterns it can form.

Table of Contents

List of Figures

Number

List of Tables

Acknowledgements

Thanks to James Defuria, Jose F de Celis, Bruce Edgar, Ed Giniger, Marc Haenlin, Xantha Karp, and Jaideep Mavoori for informative discussions. Thanks to Daniel Schindler and Thomas Daniel for reading through and commenting on this dissertation. Funding for this work came from a Howard Hughes Predoctoral Fellowship to Eli Meir, National Science Foundation grants MCB-9732702, MCB-9817081, and MCB-0090835, and the Mathematical Biology Training Grant at the University of Washington. George von Dassow drew Figures 2-1, 2-3, and 3-4, 3-5, and helped construct several of the other figures.

Thanks also to everyone in the Foe and Odell lab, including Jon Alberts, Dara Lehman, and most especially George von Dassow and Ed Munro, with whom I did all this work as a collaboration, surely one of the most rewarding and fun collaborations I will have in my life. Thanks also to Victoria Foe for much advice on experimental components of this project which did not make it to print, and to Garry Odell for being a wonderful advisor. Finally, I would like to thank the Zoology department in general, and especially the many other graduate students with whom I interacted in my time there, for being a great place to pursue scientific questions and just a generally nice atmosphere in which to work.

Chapter 1
Introduction: An overview of this thesis

Biologists are accumulating a phenomenal amount of information about genes, their functions, and their interactions. The advent of whole-genome sequencing, DNA microarrays, and proteomics promises an imminent embarrassment of riches. Soon, if not already, the available maps of known genetic interactions for any particularly well-studied cell physiological or developmental process will be so complex as to defy the ability of human brains to understand and manipulate those maps without help from computers. Most genes can be said to have a "function" only as part of a network of cross-regulatory and biochemical interactions with other genes and their products. Increasingly biologists think in terms of whole networks, the biological analog of the integrated circuit. The network, rather than any individual gene, is the functional unit that does useful things such as transduce, transmit or transmute signals, stabilize cell states, form patterns in groups of cells, etc. This is particularly evident for many paradigmatic developmental mechanisms in which genetic networks produce spatial patterns in a sheet of cells (e.g. the segmentation cascade in *Drosophila*) or temporal patterns (e.g. cell cycle oscillator or the circadian clock), or both.

There is growing interest in using computers to synthesize genetic data into mechanistic models at the network level (Bray, 1995; Hartwell *et al.*, 1999). Several inter-related goals motivate that interest: For some biological process of interest (e.g. bacterial chemotaxis or *Drosophila* segment boundary formation), is our map of interactions between molecular components complete enough actually to explain that phenomenon? If so, what systems-level properties such as robustness to perturbation, unanticipated from the nature of the parts themselves, emerge in the whole network? What rules, if any,

govern the "design" of genetic networks, and which details are crucial to making a mechanism that works robustly?

A natural approach for making computer models is to represent a network of interacting genes as a set of coupled ordinary differential equations (ODEs) and then integrate the system of equations over time to determine how the network behaves (Barkai and Leibler, 1997; Bray *et al.*, 1998; Laub and Loomis, 1998; Tyson *et al.*, 1996). (For alternate approaches see (Barkai and Leibler, 2000; Kauffman, 1993; McAdams and Shapiro, 1995; Reinitz and Sharp, 1995; Thieffry *et al.*, 1998)). Such equations keep track of continuous changes in the concentrations of gene products over time. Although excellent general-purpose computer programs exist for solving ODEs (e.g. Mathematica, Maple, or Matlab) we find them unwieldy for genetic networks. For instance, a simple representation of the neurogenic network in *Drosophila* (see below) involves 15 different components, operating across at least 16 cells, with the dynamics of the network governed by 53 free parameters and including ~500 coupled differential equations. Using a standard mathematics package, the task of constructing the model is slow and error-prone, and requires a degree of mathematical and programming sophistication which most lab-bench biologists don't possess.

We have produced a new program called Ingeneue which solves many of these problems. Ingeneue uses a library of stereotyped, biologically meaningful building blocks to assemble models of genetic networks. This makes it straightforward to translate network diagrams, such as those that often appear as the last figure in molecular genetics articles, into systems of ODEs. Having assembled a model, Ingeneue imposes a user-specified initial pattern, and then integrates the ODE system over time to find the temporal and spatial pattern it produces. Ingeneue can search for combinations of parameter values

that confer on a network the ability to make particular target patterns. Once sets of "working" parameter values are discovered, Ingeneue facilitates testing how sensitive the model is to changing those values. Ingeneue is modular, and is designed to be extended with a minimum of effort. Most features can be accessed through a point-and-click interface.

In this thesis I use Ingeneue to discuss some system-level properties of the *Drosophila* neurogenic network. The neurogenic network is best known for its role of selecting the cells that are precursors to many of the central and peripheral nervous tissue in *Drosophila*. Many of the genes of the neurogenic network are also used in a wide variety of other contexts both within flies and across the animal kingdom. The network has different modes of action in these different contexts, but in many cases it appears to perform "lateral inhibition", in which a cell that somehow becomes biased towards one fate inhibits its neighbors from adopting that same fate. Using Ingeneue, I show that our model of the neurogenic network, based on the published experiments of many researchers, can, in fact, perform lateral inhibition. Moreover, it can perform this behavior with a stunning insensitivity to both parameter values and initial conditions. We previously found a similar result for the segment polarity network in *Drosophila* (von Dassow et al, 1999). I show that the region of parameter space within which the network operates successfully has a large central basin with long arms extending away from it, a topology that has implications for the evolution of the network. Moreover, the network can produce qualitatively different behaviors in overlapping regions of parameter space, where the final spatial pattern made is selected by the pre-pattern (initial conditions) we impose. This also has evolutionary implications, showing how the robustness of the network can lead to an evolutionary flexibility.

Chapter 2
Ingeneue: a versatile tool for reconstituting genetic networks *in silico*

Note: This chapter is based on a paper in press at the Journal of Experimental Zoology (Meir *et al.*, in press).

Before discussing the properties of a particular genetic network, we need a tool that lets us conduct these explorations accurately, easily, and in a reasonable amount of computer time. In this chapter I describe a new program called Ingeneue which we (George von Dassow, Edwin Munro, Garrett M. Odell, and I) designed specifically to build and explore models of genetic networks (Meir et al, in press). Ingeneue lets the user take a diagram of a genetic network and quickly turn this into a set of ordinary differential equations that represent identical copies of the network installed in each of many cells, hexagonally packed in a two-dimensional sheet. It can then search for parameters of these equations that allow the network to produce specified temporal or spatio-temporal patterns. Ingeneue has three principal advantages over other available mathematical software: It can instantiate the same network model in each cell in a field; it constructs its equations from pre-made building blocks corresponding to common biological processes; it automates searching through parameter space, sensitivity analyses, and other common tasks. In describing Ingeneue here, I occasionally make the ideas concrete by referring to our models of the segment polarity network, as described in (Dassow et al, 1999) (Fig. 2-1).

Overview of Ingeneue

We wrote Ingeneue in the object-oriented language Java. Object-oriented programming means dividing a program into classes of "objects", where each object knows its own

state and how to perform its own behaviors. This paradigm suits many biological problems because biological systems tend to divide naturally into distinct objects (e.g. species, individual organisms, genes, neurons, etc.). Thus, Ingeneue consists of inter-dependent, extensible libraries of object types. Some of them encapsulate representations of biological entities or their properties, whereas others are tools for manipulating those objects or conducting numerical analyses. Java is rapidly gaining favor as a language for constructing scientific software, both because it is well-designed and also because a program written in Java runs on all kinds of computer platforms without major changes. We find this to be true in practice as well as in theory. Early in the evolution of the Java language and platform, Java programs suffered a serious performance deficit compared to programs written using more traditional languages such as C/C++ or Fortran. However, by now very efficient Java runtime environments are available for the most commonly used computer platforms, and our experience is that numerical routines written in Java compete with similar routines coded in C++.

Ingeneue's core construction and analysis module creates a system of ordinary differential equations from a textual description of the network that the user types, and integrates those equations over time. A graphical interface then allows the user to modify quantitative (but not yet topological) properties of the network and view its dynamic behavior. The core can run without the interface so one can run the program remotely, e.g. as a batch job on a Unix server. Ingeneue represents a genetic network using 3 types of objects: Nodes, Cells, and Affectors (Fig. 2-2). Nodes are the network components, such as mRNA's, proteins, and protein complexes; these are the dependent variables in the model representing time-varying concentrations of gene products within a single cell (or cell compartment). Since gene copy number does not change over time we do not include a node to track DNA. For reference, the simplest version of the segment polarity

network we have modeled requires 13 distinctly named Nodes per Cell: 5 mRNA's, 5 directly transcribed proteins, 1 protein fragment produced by cleavage of the Cubitus interruptus protein, 1 protein complex between HH and PTC, and an extracellular as well as intracellular form of the Wingless protein (Fig. 2-1). A 14th "dummy" Node provides a basal transcriptional input to the Node ci (which represent the *cubitus interruptus* mRNA). In the figures, Nodes are represented by oval icons for mRNA's, rectangular icons for directly transcribed proteins and their cleavage products, and hexagonal icons for dimers/polymers of those proteins.

Each cell in an epithelial sheet corresponds to one of Ingeneue's Cell objects which stores a complete set of that Cell's Nodes in the network along with the identities of each neighboring Cell. Epithelial layers in developing *Drosophila* embryos generally consist of polygonal cells with roughly six sides. The issue of how cell shape affects patterning is worth exploring, but currently, for simplicity, we represent Cells in Ingeneue as two-dimensional regular hexagons which do not move. No constraint inherent in Ingeneue's design prevents us from eventually adding a more sophisticated representation of cells, sheets of cells, movement of cells within sheets, and mitoses of cells. Each Ingeneue Cell has seven compartments – a "cytoplasmic/nuclear" compartment where all intracellular Nodes reside, and six "membrane" compartments, one for each side of the cell. Ingeneue tracks concentrations of membrane-bound components (for instance trans-membrane ligands and receptors such as WG, PTC, HH, and PH in Fig. 2-1) separately for each cell face, and each pair of neighboring cell faces interact independently. Thus each membrane-bound component is represented by 6 Nodes per cell, although it is only represented once in Fig 2-1. Ingeneue allows these membrane-bound components to flow from each side of a cell to the two neighboring sides of the same cell. Ingeneue also allows exchange between apposite faces of neighboring cells. Together, these two

features allow molecules to diffuse across the sheet of cells (see Fig. 2-2, middle panel). Ingeneue can easily accommodate other notions of cell compartmentalization which certain other applications may require.

Ingeneue uses arrays of hexagonally close-packed cells (Fig. 2-2). In the current version boundaries wrap around both horizontally and vertically so, for instance, the left-hand neighbor of a cell on the left edge is a cell on the right edge. Our wrap-around boundaries, which give our cell sheets the topology of a torus, eliminate boundary effects. This is an advantage for spatially periodic patterns such as those the segment polarity network typically makes. In cases where wrap-around boundaries are inappropriate, such as the neurogenic network, we add an extra strip of "dead" cells around the edge of the sheet with initial conditions that cause them not to participate in the patterning process being modeled. The initial segment polarity pattern is composed of vertical stripes that repeat every four cells, so for the simplest test of this network we use a 4 x 1 rectangle of cells.

An advantage of using Ingeneue for genetic network modeling is that the user need specify the network only once in a typical cell. Ingeneue does the drudgery of instantiating the network in each Cell in the model and setting up the appropriate relations between neighboring cells. Thus in Figure 2-2, there are only 3 components we need to specify: wg (mRNA) and I-WG (protein) in the intracellular compartment; and E-WG (protein) in the typical membrane compartment. There are 4 interactions: wg->I-WG (translation); I-WG <-->E-WG (bi-directional exchange between compartments); the transfer rate of E-WG from edge to edge around the cell periphery; and the transfer rate of E-WG from cell to cell between apposite faces. This is much faster and more reliable than enumerating by hand, as one would have to do absent a tool like Ingeneue, all 120

nodes and 285 connections required for the 15-cell grid at the top of Fig. 2-2. For the minimal segment polarity model in a mere 4 x 1 cell grid, the use of a standard symbolic mathematics package would require the user to type out a system of nearly 140 coupled ODEs. Ingeneue reduces this to 13, and uses stereotyped building blocks, Affectors, to construct the differential equations.

Affectors represent the interactions between Nodes. Each Affector encapsulates a formula corresponding to a physical process involving one or more Nodes. Each Node computes the change per unit time in its concentration (its time derivative) by combining the values all its Affectors contribute (that is, the affectors combine to constitute the right hand sides of the ODE system). Table 2-1 illustrates the four conceptual groups which most of Ingeneue's Affectors fall into. Some Affectors govern synthetic processes such as transcription and translation. A second group governs spontaneous decay of each molecule. A third group represents transformations of proteins from one form to another (via reactions such as cleavage and heterodimerization), some of which may be reversible. The fourth group mediates exchange of molecules between different compartments within and among cells. Ingeneue currently includes approximately 60 Affectors (all in dimensionless form, as described in the supplement to (von Dassow *et al.*, 2000)) which we developed to construct our segment polarity and neurogenic network models. These provide a versatile, expanding basis for constructing models of similar scale. Furthermore, it is a simple Java programming task to create a new Affector type, especially given the existing library as exemplars. This is a key feature of our design goals for Ingeneue: it should be easy for a biologist possessing an acquaintance with the mathematics and only a little familiarity with Java programming to extend Ingeneue to accommodate a wide variety of similar gene network modeling tasks.

For some processes we use Affector formulae that represent the exact chemistry involved – examples include first-order reactions like decay and second-order reactions such as ligand binding. In other instances we only approximate the kinetic process. The most common approximation we adopt is to use generic sigmoid curves to model how transcription factors work. We digress a moment to explain that we use approximations because we prefer formulae that correspond to a biologist's description of a gene network quantified by parameters which, in principle, biologists could measure experimentally. For instance in most contexts biologists explaining how a group of genes regulate each other would not detail all the steps of transcription factor binding and sequential assembly of the generic transcriptional machinery. Instead they would say things like, "Engrailed represses *cubitus interruptus* transcription". The details of how it does so, even though they might matter a great deal, often remain unknown. What one *does* know is that every target gene has *some* maximal rate of transcription. That maximal rate is one good parameter for quantifying the synthesis rate function for that gene's product. Saturation at a maximum possible transcription rate implies that, for every regulator, there is some concentration at which the promoter would operate at one-half its maximal rate. That half-maximal concentration is a good generic parameter for quantifying how that regulator works.

The functional form quantifying the dose-response relation between the concentration of a transcription factor and the transcription rate it modulates *must* be non-linear if that transcription rate saturates. The practically important feature of that non-linearity is how sharply non-linear it is. The parameter we find most useful for that purpose is the slope of the function at the half-maximal point. We presume that greater slopes at that point would arise from higher-degree cooperative binding effects, e.g. as would arise if only tetramers of a transcription factor activated/repressed transcription, or equivalently if four

monomers of some transcription factor must bind to a promoter to modulate transcription rate. This reasoning leads us to adopt the kind of curve Fig. 2-3 shows and the three parameters that sculpt it. Ingeneue uses combinations of such curves to model transcriptional regulatory interactions. By neglecting to represent explicitly the assembly of the transcriptional machinery we assume that this process is not a rate-limiting step. We use similar reasoning to parameterize generic approximating functions for most of the complicated processes in our network models. But we emphasize that Ingeneue puts no constraints whatever on an Affector's formula so others could make different choices for modeling processes such as transcription. This includes the option of giving stochastic traits to Affectors to model the behavior of genetic networks in which so few molecules inhabit each cell as to make transcriptional regulation and other processes noisy.

Currently, the most mathematically-complex Affectors are those governing transcription. This reflects the inherent complexity of the transcription process which currently transcends biologists' full understanding. Multiple transcription factors activate/inhibit most enhancer regions through detailed interactions we do not know. In cases where only a single activator acts on a target gene, we use the simple parameterized sigmoid function in Fig. 2-3 to quantify the transcription rate. To add a single inhibitor we might choose to multiply the activation function by (one minus a similar function) (Table 2-1). Even in such a simple one activator/one inhibitor case we are already making assumptions about the physical mechanism of inhibition. Multiplying the activation by an inhibitory term means the inhibitor can shut down transcription regardless of how high the activator's concentration rises. An alternate form might postulate that the activator and inhibitor compete for the same binding sites on the enhancer. Then the relative concentrations would be critical in determining the activation level meaning the inhibition would enter in a qualitatively different way (see appendix to Meir et al, in press). With multiple

activators and inhibitors, the potential combinations get even more complicated. In our initial segment polarity and neurogenic models we assumed particular interactions, usually multiplicative, between the activators and inhibitors at each enhancer. Ingeneue includes the rudiments of a more flexible system using "meta-affectors" that add and multiply together simpler activation and inhibition formulae while still enforcing an overall maximum transcription rate for the target gene.

Partitioning the equations for a network model into stereotyped, reusable function fragments is another major advantage of using Ingeneue for modeling genetic networks. The Affectors constitute a parts kit for translating the cross-regulatory connections in a typical network diagram such as Fig. 2-1 into a set of ODEs representing those connections mathematically. This tactic enables users of this software to build and modify a network model without needing to derive equations themselves for each case. (Nevertheless, we advise users to understand the generic nature of the kinds of ODEs Ingeneue uses, and to understand how each choice differs mechanistically.) This tactic also greatly speeds up the process of changing network topologies, even for mathematically sophisticated users (see example below).

Creating a model of a gene network using Ingeneue thus consists of specifying the Nodes, Affectors, and the size/geometry of the Cell array. In order to do anything with that topology one must assign numerical values to the Affectors' parameters and to the initial concentrations of each Node in each Cell. Then Ingeneue "runs" the model by integrating the equations over a user-specified time interval. Running the model produces dynamic behavior, which we generally want to compare to gene expression patterns in the real biological system. The network's behavior, and the final state which the pattern may approach asymptotically after long times (if such a state exists), depend

on all the models ingredients and initial conditions. This behavior may be sensitive to any particular ingredient, or may hardly be sensitive to any of them. I now describe some of Ingeneue's tools for exploring those dependencies.

Wandering through high-dimensional parameter space seeking zones in which model networks emulate real networks

In the classical context of genetics, a gene's effect(s) on phenotype characterize its *function*; the *function* of *wingless*, for instance, is to specify regions of naked cuticle in each segment (this is one "output" of wingless expression). In the context of molecular genetics, gene function involves the way the gene product interacts with other genes and their products within some pathway or program. That is, the *function* of *wingless* is to respond to certain intra- and inter-cellular signals and transmit them to certain targets. How does one ascribe a function to a network of genes? Going from individual genes to networks, it seems sensible to try abstracting functional "behaviors", much as one would do to comprehend the highest-level function of integrated circuit chips. One *function* of the segment polarity network is *to do a certain task*, and the question is, what task? Another function of the same network may be to do another task at a different time in development in the same organism.

A simplistic description of the segment polarity network's task in early fruit fly development is to sharpen initial condition cues from pair rule genes expressed earlier into the parasegmental boundaries and then maintain these boundaries for the rest of the fly's life. The boundaries are defined by expression of the network's constituent genes in a particular pattern. Another way of saying this is that if somehow we could make a naïve field of cells capable of expressing only the segment polarity genes (and all the generic machinery for basic cell function), and if somehow we could provide an initial

pre-pattern equivalent to input from the pair-rule genes prior pattern in the blastoderm, we would expect this network stably to maintain an asymmetric spatial regime of gene expression. That, we propose, is the *function* of this network. Ingeneue allows one to test that *in silico*. One can reconstitute a working circuit from the parts list deciphered by molecular geneticists and inquire whether that circuit indeed does the task it is supposed to do. Incidentally, we find the abstraction of a network's functional task one of the most difficult (and most critically-vulnerable) parts of the entire modeling process. Given a group of genes that interact, how are we to characterize what it is that they *do*? Which aspects of what they are *observed* to do are the important aspects?

No matter how difficult it is to deduce the function of a specific network, that function usually reduces to producing a pattern of gene expression in either time or space. The spatio-temporal pattern that emerges depends not only on the network's topology but also on the values of parameters in the model and on the initial conditions. These parameters quantify the shapes and strengths of the network's connections by specifying biochemical reaction rates for translation, degradation, dimerization, and so on. Exploring how a network's behavior may (or may not) change as parameter values change is the central, inescapable function of all gene network modelers (until actual values for all the parameters have been measured - thus, realistically, until the author of this thesis has passed away). Even the simplest realistic networks involve extravagant numbers of parameters (48 in Fig. 2-1). For much simpler networks than Fig. 2-1 it is possible to catalog exhaustively the networks' behavior across all biologically realistic parameter combinations, but this is out of the question for most networks. If we assign each parameter either a high, medium, or low value, it would require roughly $3\text{^}48 \sim 8 \times 10^{22}$ samples to try every possible combination in the segment polarity network. Our lab's existing computers could perform the necessary calculations in 10^{14} years. So, somehow,

one must estimate plausible values from circumstantial evidence, sample a reasonable distribution in parameter space stochastically followed by non-linear parameter optimization algorithms, or just guess values that might work. We have thought of four schemes for sampling parameter space:

- Using non-linear optimization techniques that start from a guessed set of parameter values and search iteratively for nearby parameter values that better match the target behavior;
- Using empirical information to constrain the problem, such as the knowledge that transcription factor X is a potent activator of gene y but a poor activator of gene z, or that protein A is much more rapidly degraded than protein B
- Using intuition about what values will make the network perform its "function";
- Having the luck to model a network whose connection topology, we presume, natural selection has optimized over deep time to have the mysterious property that it is trivially easy to find, by accident, sets of parameter values that confer the desired behavior on the network.

We confess to relying primarily, though not exclusively, on the fourth option. For the segment polarity network model we were able to find "good" parameter sets that caused the network to make a reasonable match to Fig. 2-1b merely by randomly choosing values for each parameter. Naming mathematical parameters that confer shape and strength on network connections, then seeking "good" values for them, may seem artificial mumbo jumbo with which mathematical modelers merely confuse an already difficult debate. Regardless of what jargon one uses to describe the formidable task of navigating high-dimensional parameter spaces, however, performing that task is something evolution by natural selection must somehow accomplish.

The classes of parameters in a network model depend on the types of equations used, and vis-versa. Table 2-2 shows the classes of parameters (all dimensionless) appearing in our segment polarity models. We have tried to ensure all parameters in our models are, at least in principle, measurable quantities. It follows that the parameters are thus physically intuitive quantities, for instance the rate at which each component decays over time or the concentration at which a transcriptional activator half-maximally turns on transcription. In each model, we set bounds on the realistic range of each parameter. Where available, we used published values as guides to the general range, and then expanded bounds to span the smallest and largest biologically reasonable values. As an example, in exploring the segment polarity network we let what we call "half-lives" (the degradation time constants) vary between 5 and 100 min. We considered it unlikely that these molecules were degraded much more rapidly than with a 5 min. time constant (corresponding to a true half-life of about 3 min., about what has been measured for *fiz* mRNA). At the upper end, the segment polarity pattern forms over the course of hours or less, so the gene products involved must degrade fast enough to change on that time scale. We thus set the upper limit on degradation time constants at 100 minutes. Published measurements of half-lives in the segmentation network all fall within this range: Engrailed protein has a half-life of < 15 min. (Weir *et al.*, 1988); Armadillo protein (involved in segmentation though not explicitly represented in the model discussed here) has a half-life of around 12 min. (Pai *et al.*, 1997); Cubitus interruptus protein, in cultured cells, has a half-life of 75 min. (Aza-Blanc *et al.*, 1997).

Our approach to automating the search for "working" sets of parameter values requires a goodness-of-fit function (of the values of all parameters) which we craft to assess how close the network comes to matching the target behavior. Pattern-matching is currently the weakest part of Ingeneue in terms of making a program that can be used by biologists

without additional customization, and we do not yet have a full solution. Presently the user must custom-code their own pattern recognizer if the few we supply do not suit the task. It is much less clear in the case of pattern recognition than it is for the Affectors what the primitive unit for general applications should be, so to date we have supplied very few. The general strategy in Ingeneue is to allow the user to assemble, from a library of primitive parts or by custom coding, so-called "Stopping Condition" objects. Stopping Condition objects monitor the progress of the integration run and return a scalar score measuring how well the run conforms to some ideal behavior (large values mean bad behavior). Our current Stopping Conditions can look for primitive behaviors such as "Node X on in Cell 1," "Node X off in Cell 2," "Node Y oscillating in Cell 3". We then have the facility to combine several Stopping Conditions together to look for more complex patterns.

For example, the target pattern for the segment polarity network is a set of stripes: a stripe of en in the first column of each parasegmental repeat, a stripe of wg in the fourth column, a stripe of hh where en is on, and so on. Our pattern recognizer function for the segment polarity network is thus a set of several Stopping Conditions that combine to recognize multi-gene stripes. Together these assess whether the model achieves the correct on/off pattern of wg, en, and hh, in the desired positions and in a sufficiently short time. The stripe recognizer returns a better score as the difference between the concentrations within that column and in neighboring columns grows larger, and its score also improves if the concentration within the column is stable over time. We laboriously hand-tuned the recipe for the segment polarity model's pattern recognizer function until it recognized only the spatio-temporal patterns we wanted. We doubt we will ever be able to eliminate the need for such hand-tuning, but the challenge for future development of

this aspect of Ingeneue is to incorporate a versatile library of pattern recognizer modules that allow one to avoid custom coding for most applications.

Given a function that scores patterns, Ingeneue can search parameter space automatically for sets of parameter values that produce that pattern (i.e. that achieve a sufficiently low value of the "objective function" which measures distance away from the ideal pattern). This is another major advantage of using Ingeneue to build and analyze gene network models. The basic framework for automated searching is a collection of "Iterator" objects. We have been surprised to find that so far, for several different networks and target patterns , the most useful algorithm has been random sampling of parameters from within biologically realistic ranges of values. Other Iterators implement various algorithms for navigating the landscape in parameter space including a variety of standard and custom-made non-linear optimization schemes. Because most of our work to date has, fortuitously or otherwise, allowed us to take the most simple-minded of approaches, we have yet to thoroughly explore the effectiveness of various sophisticated strategies for searching parameter space. We expect, however, that since these network models all use similar equations, it may be possible to identify which search algorithms work best on this whole class of models. Once (if) one finds sets of parameter values for which the model works, the Iterator framework enables automated sensitivity tests. For example, Fig. 3 of (von Dassow *et al.*, 2000) was made using the "TransectIterator" object, which, starting with an input parameter set, simply scans along the entire range of one or more parameters whilst keeping all other parameter's values fixed, and records the score at each point. Ingeneue's modular design allows one to add any recipe one wants to try.

Using Ingeneue with a Mouse

The Ingeneue engine described above can run on its own, without a graphical display, simply based on text file input specified in a Unix command line. This is useful for running Ingeneue remotely, once one has developed a reliable pattern recognition function and devised a strategy for searching parameter space. When starting a new task with the program, though, it's vital to see the dynamic behavior of the model as it runs. This develops the user's intuition about the model, lets the user fine-tune automated strategies, and helps to catch mistakes. For such interactive uses, Ingeneue provides a simple graphical interface for making certain kinds of changes to the model (for instance, changing parameters and initial conditions, but not (yet) the topology of network connections), and viewing the time-varying concentrations of any Node in any Cell as a model runs.

Ingeneue has four main windows in addition to the console (Fig. 2-4). The Cell View (upper right) shows the concentrations of different Nodes within each Cell in the model. Clicking on one of the hexagons representing an individual Cell brings up the Inspector window which shows a numerical display of the concentration of that Node in that Cell. The user can change the initial or current concentrations of a Node in any Cell. On the upper left is the Network View, which displays the topology of the currently loaded network. Different shapes correspond to different types of Nodes (mRNA, protein, protein complex, etc.), and the lines indicate which Nodes influence each other through Affector objects. For instance, the line from en to EN represents the EN translation Affector. Clicking on the circle within this line causes the Inspector window to show a list of the parameters that appear in that Affector's formula, and allows the user to change the parameter values.

The segment polarity model has too many parameters to view entire sets using standard graphs, yet it is very useful to compare different parameter sets visually. To display multiple parameter sets, each containing dozens of parameters, we use "wheel" plots where each parameter set is displayed as an irregular polygon intersecting the many spokes of a wheel (Fig. 2-5). Each spoke represents the axis along which we allow an individual parameter to vary. Ingeneue uses either linear or log scales along these spokes as the user specifies. The inner circle defines the minimum and the outside circle the maximum possible value. Thus a single polygon, intersecting one point on each spoke, represents a set of parameter values via a one-to-one, onto, map between these polygons and a rectangular prism in the high-dimensional parameter space. Fig. 2-5 shows a sample wheel plot of 5 different parameter sets that each confer upon the segment polarity network the ability to pass our most basic functional test. This makes it visually clear how far flung those parameter sets are as points in the parameter space. Buttons and menu choices in the wheel plot window (bottom of Fig. 2-4) allow the user to flip quickly through many parameter sets and impose any of them on the currently-loaded model. This window also allows one to calculate simple statistics (standard deviations, cross-correlation coefficients, etc.) from a group of parameter sets.

Above we highlighted three major advantages of using Ingeneue to model gene networks instead of general-purpose mathematics software: 1) Ingeneue handles the busy-work of stamping out copies of the gene network template into each cell in an arbitrary-sized field of cells; 2) From a text description of a network, Ingeneue constructs its ordinary differential equations using stereotyped building blocks; 3) Ingeneue facilitates automated searches through parameter space, sensitivity analyses, and similar tasks. All these combine to make possible a fourth major advantage, which is that Ingeneue makes it easy for the user to test systematically the effect of changing the parameters,

architecture, and components of gene network models. These advantages are all illustrated by our results in the next two chapters.

Discussion

Ingeneue is an early version of what is sure to become a whole breed of software for dealing with large tangles of molecular genetic data as networks, circuits, and systems. Because our knowledge of how genetic regulatory systems operate is advancing so rapidly, computer tools that help integrate and interpret these data will soon become critical adjuncts to lab-bench molecular biology, just as they have become for various fields from enzymology to ecology. Yet most biologists do not presently receive or seek the mathematical and computer science training they would need to develop such tools on their own, or even to use general-purpose mathematical software. A program like Ingeneue can build mathematically rigorous models using a syntax that biologists with brief training in differential equations can learn easily. This biologist can then explore his or her favorite networks through the graphical interface and gain an intuitive understanding of its dynamical behavior as a whole mechanism. This kind of computer-assisted synthesis, if made accessible to the scientists that actually confront the biological subject every day, will help us all understand how gene networks operate.

We designed Ingeneue as a general tool. Because the code is highly object-oriented and separated into distinct pieces that don't rely explicitly upon each other's details, Ingeneue can be extended easily to add features, methods, and facilities. This is particularly evident in the Affectors, where a trivial amount of code adds a new formula. The same thing is true for pattern-matching algorithms, search strategies, initial conditions, and the graphical interface. In addition to ease of modification, the program's structure has made it easy for us to replace mathematical formulas with names (of Affectors, for instance),

and in the future with graphical symbols, that can be combined together to make models. Ultimately this architecture will become a grammar and syntax for translating a diagram of a gene network to a set of mathematical equations and back again. Using this syntax we can construct new networks almost entirely from the existing Affectors in relatively short times. Even those of us who have some mathematical training find this tremendously helpful! Perhaps even more important, the stereotyping of building blocks makes it easier to compare one model to another; if two models have been built from the same parts and analyzed the same way, the results should be more readily comparable than if two different artisans crafted models in their own styles. Just as helpful is Ingeneue's ability to instantiate a network over a field of cells, correctly hooking up exchanges of all the transmembrane components, and eliminating many of the small mistakes that would inevitably creep in when trying to keep track of so many equations by hand.

Ingeneue is still very much a work in progress and has several serious conceptual and technical weaknesses which I list for completeness.

One deficiency is Ingeneue's current inability to deal with morphogenesis or cell movement. In many cases, developmentally interesting patterns are formed at the same time that cells are dividing, moving around, and changing neighbors. These movements may be intimately cross-coupled to the pattern formation process. Furthermore, Ingeneue assumes cells to be well-stirred reaction beakers, an approximation that is not even remotely correct. Indeed, almost every well-understood developmental mechanism includes some fascinating and functionally important instance in which the structure of cells plays a crucial role: apical-basal sorting of receptors and ligands; the flexibility of stretches of chromatin; clustering of receptors; etc. Models of all kinds must balance

tractability with realism. In Ingeneue we have chosen a certain level of realism which may limit its usefulness but which allows us to use it to explore parameter space in a way that a more realistic modeling framework would preclude because of the computational burden.

Another deficiency is Ingeneue's primitive method for recognizing patterns. Although we have a mechanism in place to build complex pattern recognizers from smaller pieces, pattern recognition is a difficult problem and it's not clear whether our mechanism will work in general without imposing an onerous coding and training burden on the user. No matter what, we doubt very much that any computer recipe can, without extensive training by a human, replace the intuition of the human biologist when it comes to pattern recognition.

We have been using Ingeneue to address a variety of questions in developmental and evolutionary biology ((von Dassow *et al.*, 2000);Meir et al., in press; von Dassow et al., in press). These include simply asking, how complete our current understanding of a gene network is. Given the known facts, is the proposed mechanism plausible? And, how do the different components of a network contribute to its behavior? We have also used Ingeneue to ask how important particular classes of interactions are to the functioning of a network, and to explore how the topological structure of a network affects its function. The examples above illustrate some of these results. We find, for instance, that adding redundant genes can potentially reduce the robustness of a network to parameter changes, even though it may increase its robustness to gene knock-outs. Most of all, we feel that through model-building we can translate what the community of developmental biologists knows about developmental mechanics into the theoretical framework for evolutionary biology. For example, our model of the segment polarity network

unexpectedly yielded an example of the mechanistic origins of canalization. We hope that these beginnings and the free availability of this program will inspire other biologists to try similar explorations.

Table 2-1. Partial menangerie of Ingeneue's Classes of Affectors. We combine each Affector's formula into the right hand side of the ODE that specifies the time rate of change of a Node's concentration. Under "Synthesis" the formulae characterize transcription regulated by a single activator, and transcription regulated by an activator working against a single *global* repressor. The "Decay" category consists of a single Affector that should attenuate every component of the model (although Ingeneue never checks). We include a non-specific decay term for every Node because all proteins decay. Since this term is linearly dependent on the concentration of the Node (we assume decay machinery is so common that no gene product we are modeling can saturate it), and all synthetic processes should saturate to be biologically realistic, the decay term ensures a maximal steady state level for every Node. Under "Transformation" the first formula represents heterodimerization between X and Y, while the second gives the rate of cleavage (or some other transformation) of X regulated by Y. Under "Transfer" the exhibited formula characterizes transfer between the apposite faces of two neighboring cells. This formula is in fact represented by an obligate pair of Affectors, thus maintaining a one-to-one relationship between Affectors and additive terms in the ODE.

Affector Type	Description	Examples
Synthesis	Transcription of mRNAs and translation of mRNAs into proteins.	$\frac{T_c}{H_x}\left(\frac{Y^{\nu_{hs}}}{K_{Yx}{}^{\nu_{hs}} + Y^{\nu_{hs}}}\right)$ $\frac{T_c}{H_x}\left(\frac{A^{\nu_{hs}}}{K_{Ax}{}^{\nu_{hs}} + A^{\nu_{hs}}}\right)\left(1 - \frac{I^{\nu_{hs}}}{K_{Ix}{}^{\nu_{hs}} + I^{\nu_{hs}}}\right)$
Decay	First-order generic, non-specific decay that (usually) affects all Nodes.	$-\frac{X}{H_X}$
Transformation	Reversible and irreversible changes of one Node into another, e.g. cleavage, phosphorylation, dimerization, etc.	$T_c Y_c\left(-k_{X+Y\rightarrow XY} XY\right)$ $-C_{YX} X \frac{T_c}{H_X}\left(\frac{Y^{\nu_{hs}}}{K_{YX}{}^{\nu_{hs}} + Y^{\nu_{hs}}}\right)$
Transfer	Various transfers: transfer of membrane-bound Nodes among cell faces, transfer between cells, endo- and exo-cytosis, etc.	$r_{flux\,of\,X}\left(X_{appositeface} - X_{thisface}\right)$

Table 2-2: Common classes of parameters. See the Supplement to von Dassow et al. (00) for a more extensive discussion.

Parameter Type	Symbol	Description
Half-maximal activity	K_{YX}	Dimensionless concentration of a Node at which that element half-maximally activates or inhibits some process.
Cooperativity (or Hill) coefficient	ν_{YX}	An exponent that determines how steeply the rate of some process changes as some regulator Node increases (i.e. how S-shaped the dose-response curve is).
Half-life	H_X	The half-life of each Node with respect to non-specific degradation – usually this is actually the time constant, which is the half-life divided by ln(2).
Maximal rate	C_X, V_{YX}, etc.	Miscellaneous rates, e.g. for a cleavage reaction, equivalent to V_{max} for an enzyme.
Maximal concentration	X_o	The maximum dimensional concentration that a Node can achieve at a steady state. This parameter is required chiefly in heterodimerization reactions.
Flux rate	r	Rates at which transfer process equilibrate Node concentrations between various compartments
Activation strength	α	The relative strengths of different activators and inhibitors in complex enhancer regions.

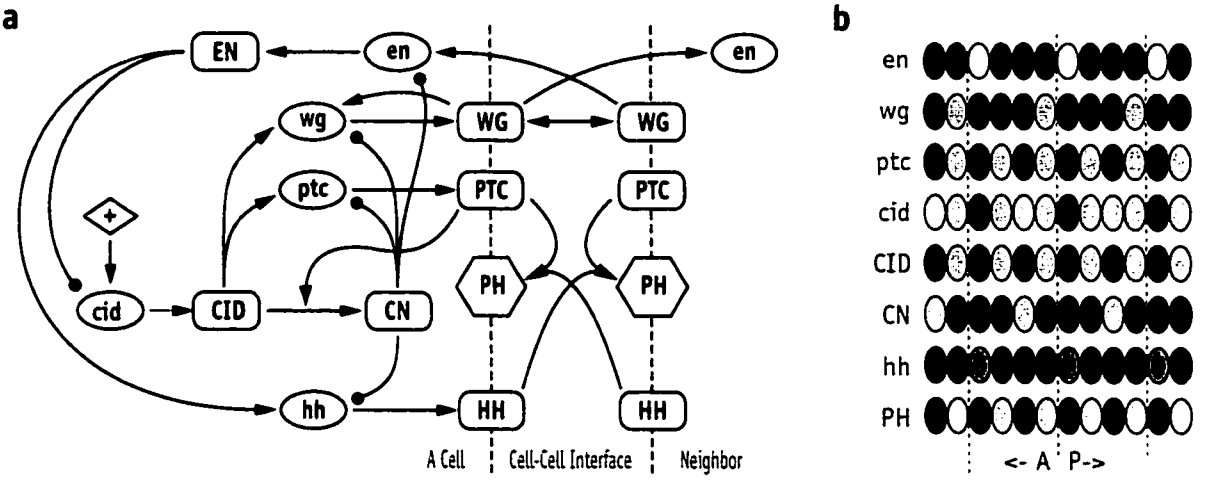


Figure 2-1: The segment polarity network and the target pattern. (a) Core components of the segment polarity network in *Drosophila*, and their interconnections; this diagram is a "minimal subset" that we used in all the work described in this paper and in von Dassow et al. (00). (b) The pattern which we asked the network to achieve and to stably maintain. Both the network diagram and pattern are based on our synthesis of the work of hundreds of researchers on these genes, and while greatly simplified relative to the real network, they are adequate to capture the most essential function of the segment polarity network. A description of Fig. 2-1b with references to the primary work is contained in the Supplement to von Dassow et al. (00), along with a description of how we translated the network map into maths.

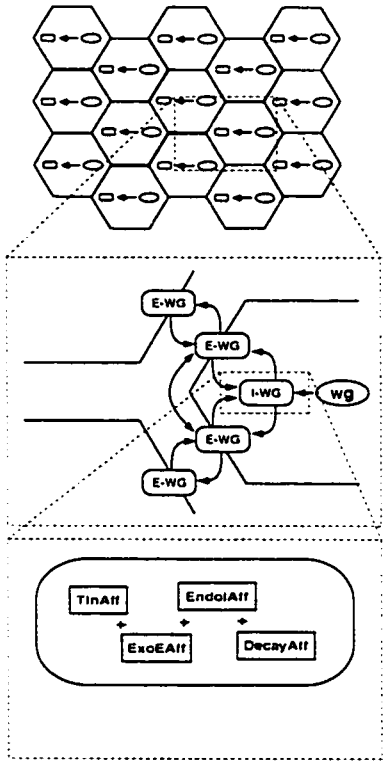


Figure 2-2: Pieces of an Ingeneue model. Ingeneue models are made of Cells, Nodes, and Affectors. Cells are hexagonal with one cytoplasmic compartment and six membrane compartments. We arrange cells in a grid, with each face of a Cell in contact with a face of one of its neighbors. Cell faces on the edges of the grid are wrapped around as if on a torus to be in contact with "neighbors" on the opposite side of the grid. Each Cell contains a complete copy of a network, where the network is composed of Nodes (that is, molecular species; ovals and polygons in middle panel) and Affectors (arrows in middle panel, boxes in lower panel). The middle panel shows a subset of the Nodes and Affectors from the segment polarity network. Transcription of the wg mRNA produces an internal WG protein pool (I-WG). This internal WG exocytoses onto each face of the Cell (E-WG), from whence it can diffuse to the apposite faces of neighboring cells. The bottom panel shows the four Affectors which are summed together to compute the time rate of change in I-WG concentration. They represent, from left to right, translation of I-WG from wg mRNA, exocytosis to the membrane, endocytosis from the membrane, and non-specific decay.

$$\frac{dX}{dt} = Trans_{max} \frac{A^{\nu}}{K_A^{\nu} + A^{\nu}} - \frac{X}{H_X}$$

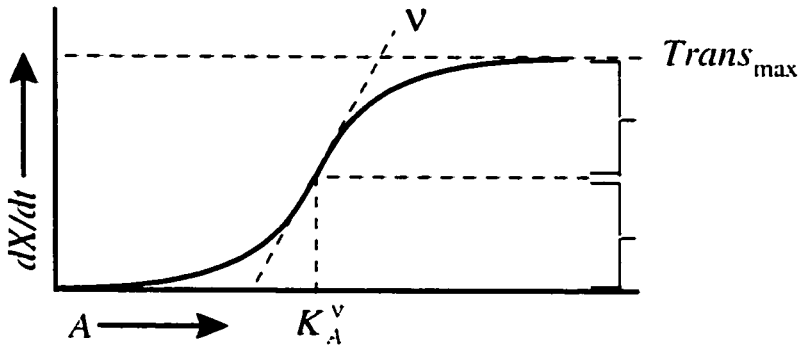


Figure 2-3. Standard dose-response curve. This S-shaped curve, generated by the differential equation shown above, is a fundamental approximation for regulatory relationships in Ingeneue. The equation in this figure is a simple case in which synthesis of X is promoted by activator A, and X degrades non-specifically. The curve represents the rate of synthesis of X as a function of A, absent decay. The salient features are that at high activator concentration, the response saturates at some maximal value, determined by the properties of the biosynthetic machinery (e.g. RNA Polymerase). It follows that at some lower activator concentration (K_A) the response reaches half its maximal value. We term the steepness of the curve's non-linearity "cooperativity" (since it could be produced by cooperative interactions among activator molecules). With a cooperativity of 1.0 (that is, no cooperativity) the dose response curve is nearly linear at low regulator concentrations, has no inflection, and saturates slowly. With a cooperativity above 10.0 or so the curve is nearly a step function. In Ingeneue models the half-maximal activation value and the cooperativity become free parameters of this formula, but the equations are normalized and rendered dimensionless such that, for processes like transcription, the maximal rate is 1.0. An inhibitory dose-response curve is thus obtained by subtracting the curve shown from 1.0. Complicated terms, such as would govern transcriptional regulation by several factors, are obtained by nesting, adding, and multiplying these curves in various combinations determined by the mechanism in question. We non-dimensionalize all equations as described in (von Dassow *et al.*, 2000).

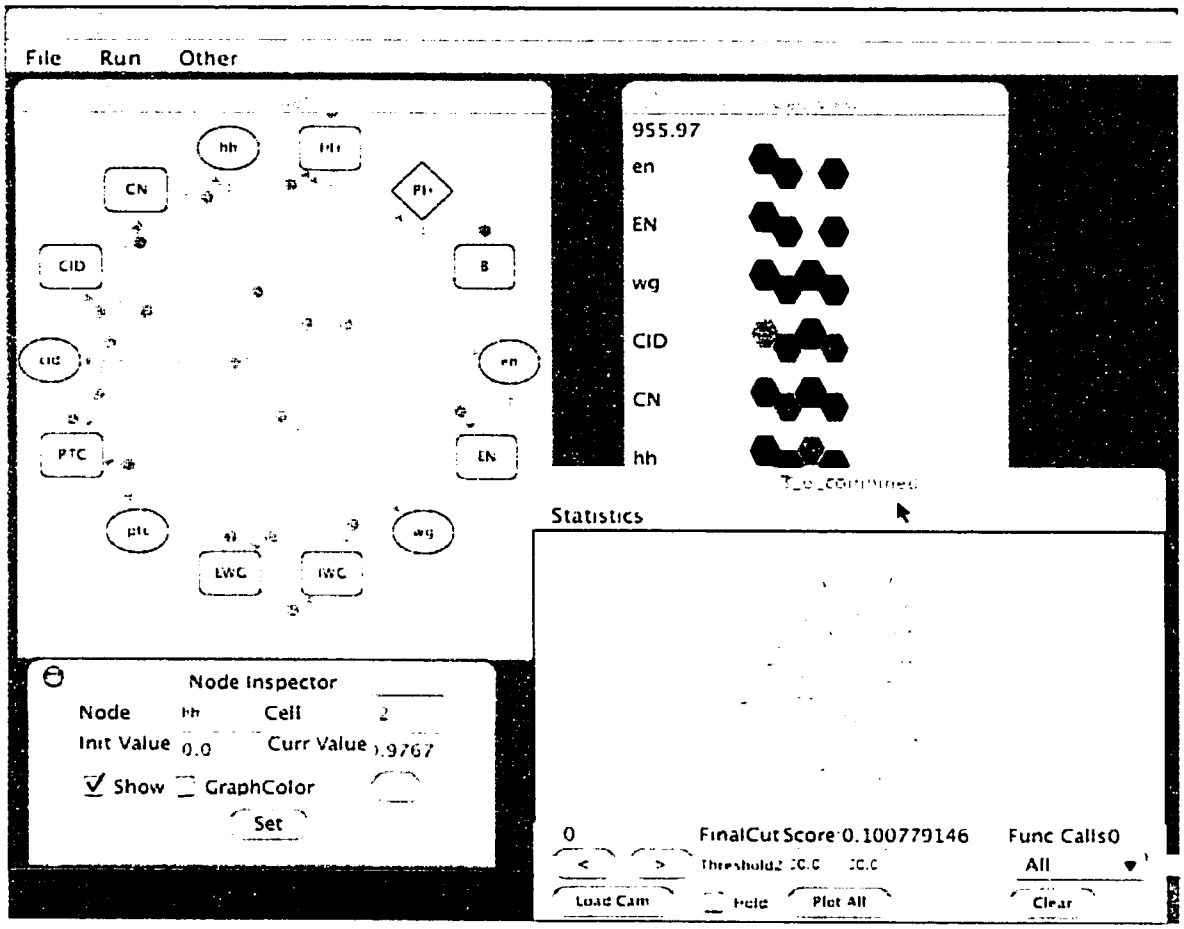


Figure 2-4. The Ingeneue interface. Screen snapshot showing the main windows in Ingeneue's interface. The top middle window shows the Nodes and Affectors in the segment polarity network. Clicking on one of these nodes brings up the Inspector window below, which allows the user to inspect and change concentrations of any Node in any Cell, and change parameter values for any Affector. To the right is the Cell View window which graphically shows the concentrations of a user-selected set of Nodes in each of the Cells in the model. In the bottom right is a window which display a battery of parameter sets on a wheel plot, and allows the user to impose one set at a time on the loaded model (see Fig. 2-5).

(a) (b)

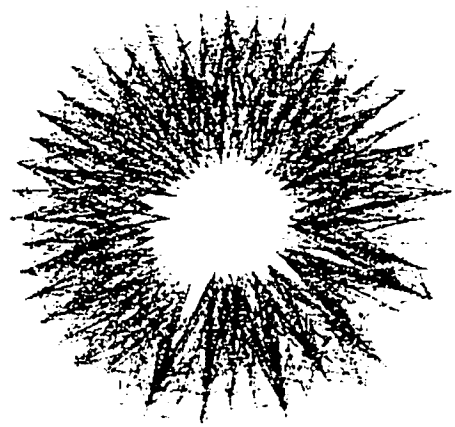


Figure 2-5. Wheel plot. Wheel plots can plot multiple parameter sets, each composed of dozens of parameters. Each spoke on the wheel is the axis for one of the 48 parameters in the model. The inner circle represents the minimum value in each parameter's range, and the outer circle represents the maximum value. Thus one parameter set is represented by a single irregular polygon with a vertex on each spoke of the wheel (a). By plotting many parameter sets on top of each other, one can get an idea of whether one or another parameter's values are clustered in one part of its range (b). For parameters where we explored values across several orders of magnitude, we both picked random values and plotted those values using a log scale.

Chapter 3
Robustness, flexibility, and the role of lateral inhibition in the neurogenic network.

Note This chapter is a replica of a paper of the same title published in Current Biology in 2002 (Meir *et al.*, 2002).

Background

In this paper we use a computer model to explore the properties of the neurogenic network, originally characterized in *D. melanogaster*. This is but one example of the many networks of cross-regulatory genes at work in complex organisms. Other familiar examples include the networks of segment polarity genes, of cell cycle genes, of circadian clock genes, and so on. Each of these seems to have remained more or less intact through long periods of evolutionary time and across many phyla. Many of these networks have acquired novel functions in different organs and species. Why is that? Is there something special about each of these networks that causes natural selection to preserve them through deep time and co-opt them to perform new functions? Or is it simply that evolution doesn't fix what first evolved if it isn't broken?

Only recently can we begin asking these network-level questions. A tremendous and ongoing experimental effort has generated an unprecedented catalog of the components and connections of many networks, while advances in computers give us the power to explore complex systems in ways unimaginable a decade ago. Recently we found that a model of the segment polarity network in *Drosophila* embryos exhibits two very interesting properties – robustness to changes in kinetic parameters, and robustness to changes in its initial conditions (i.e. due to prior expression patterns of upstream genes) (von Dassow *et al.*, 2000). This example is tantalizing, but it takes several such case

studies to support general conclusions. We therefore investigate in this paper whether a second well-studied network, the neurogenic and proneural genes (hereafter simply "the neurogenic network") has similar properties.

We use the "canonical" structure of this network as it operates during *Drosophila* development, especially in determining neuroblasts (NB) in embryos and sensory organ precursor (SOP) cells in imaginal disks (reviewed in (Simpson, 1997)). Components of this network are used in a wide range of patterning processes, from vertebrate retinal development (Perron and Harris, 2000) to nematode anchor cell specification (Wilkinson *et al.*, 1994), with apparently similar interactions among the core genes. Neurogenesis is often cited as a classic case of lateral inhibition (Doe and Goodman, 1985). This hypothesis postulates that a small group of neighboring cells all start out competent to assume a particular fate. A stochastic fluctuation or external cue slightly favors one cell over the others. The favored cell then suppresses (laterally inhibits) its neighbors to keep them from also assuming that fate.

Here we first ask whether the neurogenic network, as the literature currently portrays it, can perform lateral inhibition. Simple models encapsulating the barest essential facts of the neurogenic network (the Dl / N interaction), perhaps unsurprisingly, succeed under certain conditions in differentiating cells with initially similar levels of Dl and N (e.g. (Collier *et al.*, 1996)). The real pathway between Notch activation and Delta expression, however, has layers of feedback and modulation, including switch and homeostat-like sub-circuits. These ought to (and do) make it more difficult to accomplish lateral inhibition. Thus it is a non-trivial question under what conditions, if any, the neurogenic network can accomplish lateral inhibition, and if it can, how do we explain the inclusion

of design features that seem to work against the apparent biological role of the circuit? This is where we begin our exploration.

Abstracting the neurogenic network, and three tasks it performs.

We focused our investigations of the neurogenic network on the genes that select neural cells in *Drosophila* embryos and imaginal disks. Figure 3-1 shows our summary of the core genes, their products, and their interactions. In crafting Fig. 3-1 we approach the model-building process as a biochemist approaches *in vitro* reconstitution; by adding to the system piece by piece, we hope to figure out how each design feature contributes to the function of the essential core network. We rationalize our choice of this diagram in the Online Supplement, with a synopsis as follows (capitalized/lower case denotes proteins/mRNAs respectively):

Delta (Dl) is a ligand for the receptor Notch (N). When Dl activates N, a cleaved-off cytoplasmic piece of N binds to the transcription factor Suppressor of Hairless (Su(H)) and that heterodimer activates *Enhancer of split* (*E(spl)*) complex genes. The proneural genes *achaete* (*ac*) and *scute* (*sc*) encode transcription factors that actually specify neural fate. Both Ac and Sc are autoactivating and cross-activating: they promote their own, and each other's, transcription. Thus the proneural genes constitute a bistable switch at the heart of the neurogenic network. They also activate transcription of *E(spl)* and *Dl*. E(spl) in turn represses transcription of *ac* and *sc*. Thus the loop works as follows: something activates *ac* and/or *sc* in the neural-competent cluster. They upregulate *Dl*, whose product activates N in neighboring cells, which, through Su(H), activates *E(spl)*. E(spl) represses *ac* and *sc* in those neighboring cells. To achieve a neural fate, a cell must upregulate *ac* and *sc* enough that their autoactivation overwhelms E(spl)-mediated repression due to neighboring cells signaling through N.

We constructed three different models of the network in Fig. 3-1, which we call 'augmented', 'standard', and 'reduced'. The standard network includes all components and interactions shown in Fig. 3-1 except for *cis*-negative regulation of N activity by Dl, and E(spl) auto-repression (Fig. 3-1 without red or blue connections). Experimental evidence for each of the latter interactions exists (see Supplement) but the literature has not given them much attention. Neither did we initially, but our results below with the augmented network (which adds the red connections) indicate that these may indeed be important. Our reduced network eliminates intracellular negative feedback from Ac and/or Sc to suppress ac and sc transcription (blue connections replacing red and green connections and their E(spl) hub). Such a simplified network could have functioned in a precursor to the *Drosophila* network since the similar process of anchor cell specification in the worm *C. elegans* appears to take place without E(spl)-like genes or function (X. Karp and I. Greenwald, pers. comm) (though the *C. elegans* version is likely derived).

We explored three spatial patterning functions with each network (Fig. 3-2). The first was lateral inhibition leading to a separation in states between two neighboring competent cells, as in a related network in *C. elegans* (Wilkinson *et al.*, 1994). We started two cells with different concentrations of ac and sc products, and tested whether, within 300 minutes, the initially higher cell achieved a high concentration while the other turned off Ac (see legend of Fig. 3-2). The second test asked the central cell in a group of seven to "win", inspired by SOP-forming clusters in an epithelial sheet (Skeath and Carroll, 1992). The third test asked a double line of cells to refine to a single line, very roughly mimicking the pattern seen at the margin of wing and leg imaginal disks (de Celis *et al.*, 1996). We made and ran all models using our Ingeneue software (Meir *et al.*, in press; von Dassow *et al.*, 2000) a general purpose program for modeling genetic networks.

These tasks are intended to be stereotypes that abstract many biological phenomena, rather than accurate mimics of any one case, and we designed them to insure the network must use lateral inhibition in order to succeed. All files needed to recreate our results are available at www.ingeneue.org.

All three work!

Our first question was whether the three versions of the network in Fig. 3-1 can perform lateral inhibition. A network model consists not only of its topology (Fig. 3-1), but also dozens of parameters specifying the strength, rate, and functional form of *each* connection. The reduced, standard, and augmented versions of the network have 53, 63, and 69 such parameters, respectively. Asking whether the network can do lateral inhibition amounts to asking whether there exist values for these parameters where the network will pass any of the three tests in Fig. 3-2. We have practically no idea what appropriate values for these parameters would be, so we instantiated networks by sampling values randomly from within broad but biologically realistic ranges (Meir *et al.*, in press; von Dassow *et al.*, 2000). Any set of randomly-chosen parameter values that allows the model to pass some functional test we call a "solution".

All three models exhibited lateral inhibition in the two-cell test, but the standard model had a much lower frequency of solutions than the augmented model, while the reduced model had the highest solution frequency of the three (Table 3-1). By this naive measure, the reduced model seems the most robust of the three. However if p is the probability of picking a "good" value for each parameter independently, we expect that the solution frequency for models with equivalent robustness per parameter would scale as p^n, where n is the number of parameters. p is highest for the augmented model (Table 3-1). That measure makes the augmented network most robust. All three networks could also

mediate lateral inhibition between a central neural cell and 6 surrounding cells (seven-cell pattern, Fig. 3-2), with similar solution frequencies as for the two-cell pattern (Table 3-1). As with the segment polarity network (Meir *et al.*, in press), cooperativity (Hill coefficients > 1) in transcriptional activation and repression is essential for the network to function (Table 3-1). In many neurogenic tissues, when the winning cell is removed early in the patterning process, a cell originally destined for a non-neural fate replaces it. We took parameter sets from our two-cell test in the augmented network and removed the presumptive winner at a point where concentrations of proneural genes in the loser where beginning to drop towards zero. We found parameters where the presumptive loser could recover at that point and assume a neural fate. We also found the same result using a three-cell test (the largest number where all cells can still touch in our hexagonal grid).

There are two differences between our standard and augmented models – *E(spl)* auto-inhibition, and Dl inhibition of N activation (red lines in Fig. 3-1). Both are supported by experiments, and both contribute to improving the solution frequency (Table 3-1), but *E(spl)* auto-inhibition contributes more than *cis*-inhibition of N by Dl. We hypothesized that the reduced model has a higher solution frequency than the other two because it lacks activation of *E(spl)* by Ac and Sc (which implements a homeostat – see discussion). Indeed, eliminating Ac and Sc activation of *E(spl)* in the augmented model raised the frequency of solutions to almost the same as the reduced model (Table 3-1).

To see whether the network is robust to noise in initial conditions, we took solutions from each model and asked whether they could pass the seven-cell test from initial conditions as in Fig. 3-2, but with Gaussian noise added to the initial concentrations of ac, Ac, sc, and Sc in each of the focal cells. For all three networks we found solutions that were robust to this noise (Table 3-2). Similarly, both the augmented and reduced networks can

select a single winner with very little initial bias in the pre-pattern by amplifying small amounts of noise (Table 3-2). Solutions for the reduced network were more likely to tolerate noise than solutions for the others. Since all networks can pass our tests the results in this and the previous paragraph provide no evidence for which network is more likely to be an accurate representation of what exists in the fly. But they do point to the relative importance of the different links in conferring robustness on the network, and raise a puzzle about the role of *E(spl)* to which we return below.

To summarize, realistic dynamical models of the neurogenic network can, in fact, carry out classic lateral inhibition over an extraordinarily large portion of its parameter space, even when, unlike earlier models (e.g. Collier et al. (Collier *et al.*, 1996)), we include the intermediate players between N activation and Dl expression. As we showed previously for the segment polarity network, these results predict that the lateral inhibition function of the core neurogenic network is very robust to changes in parameters; this network is a lateral-inhibition module, whereas the segment polarity network is a boundary-stabilizing module.

Parameters are not restricted in their values

Figure 3-3 (see also Supplement) shows histograms of the values of selected parameters from all solutions we found while randomly searching parameter space using the two-cell test on the augmented neurogenic network. Although some parameters tend to cluster in part of their range, there do not appear to be any absolute restrictions. That is, solutions permeate the entire huge box within which we randomly chose values (with one trivial exception of Ac half-life, explained in the Supplement). This apparent lack of restrictions could mask compensation between pairs or groups of parameters (e.g. parameter A could be high as long as parameter B were low and vice-versa). We tested for but found no

strong pair-wise cross-correlations, so any compensation must be relatively weak, or, more likely, involve more than two parameters.

The augmented network succeeds almost everywhere inside at least one large neighborhood in parameter space.

Although only a single neurogenic network parameter showed absolute restrictions in its range, many parameters tended to cluster in a subset of the full range. Using the augmented network, we restricted each parameter to the "best" order-of-magnitude sub-range with respect to the two-cell solutions (see Fig. 3-3 legend). We then conducted another random parameter search inside these new restricted ranges. The success rate increased almost a thousand-fold inside these (still very broad) ranges (Table 3-1). Using the same restricted range found with the two-cell pattern, the success rate with the seven-cell test also increased more than a hundred-fold. We found a similar result with our model of the segment polarity network (von Dassow et al, in review). Thus both the neurogenic and segment polarity networks contain at least one large neighborhood of parameter space (10-fold wide in all directions) where solutions are very common. Furthermore, this neighborhood is shared between at least two of the patterns the neurogenic network makes.

The neurogenic network can act as a general inhibitor that outside activators must overcome.

One explanation of proneural cluster function invokes the classic Delta/Notch lateral inhibition switch. Experiments in grasshopper embryos (Doe and Goodman, 1985) wherein ablation of the presumptive NB results in its replacement by another cell in the cluster supports this explanation. A defining feature of this switch is that Dl is expressed at high concentrations only in proneural cells (as indeed occurs in wing discs (Haenlin *et*

al., 1994)) and is down-regulated in cells that lose Ac/Sc expression. In all the models above, Dl is expressed only in the proneural cluster, consistent with the wing disc paradigm, and remains expressed forever only in the winner. In some solutions, however, Dl expression attenuates very slowly, persisting in all cells of the cluster long after only the winner expresses Ac/Sc. This latter behavior is more consistent with experimental reports of Dl expression in embryos: Dl concentrations differ little among the NB, its neighbors in the cluster, and cells outside the cluster (Kooh *et al.*, 1993), and *Dl* mRNA concentrations begin to differ only after NB selection is complete (Haenlin *et al.*, 1990). These observations support a different view of neural specification where neurogenic genes inhibit neural fate equally in all cells, pre-patterning genes overcome this general inhibition only in the winning cell, with lateral inhibition playing no strong role (reviewed in (Simpson, 1997)). Supporting this view, Seugnet et al. (Seugnet *et al.*, 1997) replaced endogenous Dl with ubiquitously-expressed Dl and found a fairly normal pattern of NBs in embryos, although 20% of clusters formed an extra NB.

Seugnet et al.'s experiments are so compelling that we think it useful to explore the conflict between the two apparently opposite views just outlined. Many of the solutions in our seven-cell test exhibited similar levels of dl (mRNA) and Dl (protein) in all seven cells over most of their run, but all had at least some period of time in which dl concentration was at least twice as high, and often much higher, in the presumptive neural cell than its neighbors. If lateral inhibition and proneural regulation of *Dl* are indeed involved in neural specification, our models indicate that for at least some brief period of time Dl or some related protein (such as different isoforms of N (Wesley and Saez, 2000)) must have higher concentrations in the winner. Dl dynamics are complex (Haenlin *et al.*, 1990), and difficult to measure accurately due to the concentration of both *Dl* protein and mRNA in membranes ((Haenlin *et al.*, 1990);(Kooh *et al.*, 1993);Meir and

D. Lehman, unpub.), so it is still unclear whether there are in fact heterogeneous concentrations within proneural clusters. But if differences in *Dl* levels are important to the mechanism, how can the process still succeed in most cases despite constitutive *Dl* expression (Seugnet *et al.*, 1997)?

We added a constitutive input (with tunable parameters determining that input rate) to the *Dl* promoter in our augmented model to produce the broad distribution of Dl observed in early embryos. We found solutions in which this model still passed our seven-cell test (Table 3-1), in one of which the final Dl concentrations differed by less than 40% between the winner and all other cells. When we removed the upregulation of Dl by Ac and Sc in one of these solutions, leaving the constitutive expression, the network passed the test for lower initial Ac concentrations, but all cells became neuronal with higher initial Ac concentrations. This suggests a resolution of conflicting experimental results. In cases where the pre-pattern is well-tuned, the core neurogenic network, without lateral inhibition, can select a single winner through cell autonomous processes. But if there are small changes in the pre-pattern, or developmental noise, lateral inhibition enables the network to continue to function where it would otherwise fail.

The model neurogenic network with a single set of parameter values can form several different patterns.

We might expect that the neurogenic network, or some ancestral version, evolved originally to perform just one patterning behavior. Over time, the evolutionary process co-opted it to perform novel functions. This path would seem easier to travel if the network with a single set of parameter values could pass each of our tests simply by starting from different pre-patterns (initial conditions). We found this to be true for the augmented network. Many of the two-cell solutions also enable the network to pass the

seven-cell or line tests; a large fraction could form all three patterns (Table 3-3). Many seven-cell solutions similarly enabled two-cell, line, or all three patterns. This reveals a fundamental relationship between robustness and evolvability (see Fig. 3-4 caption) as anticipated by Kirschner and Gerhart (Kirschner and Gerhart, 1998).

The shape of the "working region" of parameter space.

Clearly, simple but realistic representations of the core neurogenic network can accomplish classic lateral inhibition. Given the large number of parameters in each model, random sampling within a huge "box" in parameter space produces solutions with extraordinarily high frequency, indicating the neurogenic network is robust to changes in parameters. From a biological perspective, however, this statistical measure is a crude characterization of parameter space. We could get such results from a single broad basin of good parameters, a skinny crevasse that snakes around parameter space, many small disconnected neighborhoods, or anything in between. These differences would have important biological consequences, both for robustness of the network as it exists today and for the ability of evolution to navigate parameter space. We explored these shapes in three ways. We first used a traditional sensitivity analysis to show that most parameters can vary by over an order of magnitude around the typical solution (see Supplement). We now propose two new measures to quantify genetic network robustness in more biologically-meaningful terms, and reveal the shape of the functional territory.

In nature, different "solutions" are blended through recombination during sexual reproduction. We tested robustness by "recombining" randomly-found solutions to create offspring parameter sets. Our 'recombination' means randomly selecting each offspring parameter value from one or the other of two randomly chosen successful parental parameter sets (see Chapter 4). Success of offspring sets at passing our tests generally

correlates with the frequencies of finding solutions through a random search of parameter space (Table 3-1). As with random sampling, the recombination success rate was much higher among solutions found within the restricted parameter ranges. These results confirm the robustness of the networks and hint that the restricted parameter ranges contain a region where almost all recombinations would be successful.

To test that latter prediction, we devised another measure of robustness which we term "mutational expansion" by adding an evolutionary component on top of recombination. We imagine that a single founder solution set of parameters expands by simulated mutations into a population of individuals who all recombine with each other randomly to produce successive generations. As more generations go by, more mutations arise, and the spread of values for each parameter in the population will grow larger. We want to find out how large a spread in parameter values the population can tolerate before the within-population recombination success rate drops too low (below 90% "mating" success) (see Supplement).

We tried this procedure on approximately 25 randomly selected 'founder' parameter sets found within the full parameter ranges ("widely selected sets"), and 25 found within the restricted parameter ranges ("narrowly selected sets") (cartooned in Fig. 3-5a). After completing the mutational expansion procedure on each, we measured the average ratio between the highest and lowest value for each parameter within the population. This measure will be highest when the volume in parameter space covered by the population is wide in all dimensions (along all parameter axes). Using this measure, we found that parameter sets from the full parameter ranges generally expanded into populations that spanned much smaller volumes than those found in the restricted range (Fig 3-5b, c;

cartooned in Fig. 3-5d). The largest volumes spanned over an order of magnitude on average for each parameter, a truly enormous range considering the difficulty of the test.

How do these populations (parameter space volumes) relate to each other? To answer this we recombined parameter sets from each population with those from the other populations. The mating success between narrowly selected populations often approached the mating success within a single population (Fig 3-5d caption). The mating success between a narrowly selected population and the widely selected populations was quite a bit lower, and the success among widely selected populations was even lower, though still higher than the success rate for completely random parameter selections. This leads us to a new view of the functional territory for this network in parameter space; it has the shape of a high-dimensional octopus (Fig 3-5e), with at least one large central area (the body) where solutions are very common, and long tentacles of good solutions snaking away from this area within which solutions are also common, but separated by regions where solutions are rare or non-existent.

Discussion

The neurogenic network's robustness leads to evolutionary flexibility

Most efforts so far to explore realistic models of gene networks show robustness to parameter variation e.g. (Barkai and Leibler, 1997; Laub and Loomis, 1998; Tyson *et al.*, 1996; von Dassow *et al.*, 2000). The core neurogenic network also exhibits enormous robustness to variation in parameters as well as robustness to noise in initial conditions. Furthermore, the neurogenic network model reveals a fundamental relationship between robustness and evolvability. We tested the model's ability to perform three distinct functions. Some parameters sets enable one or another function; however the existence

of parameter sets enabling all three functions means that once the network is tuned to do one of those tasks, say, the two-cell choice task, the parameters are free to vary, only *because* of the network's robustness, into the subspace in which the model could also make the seven-cell and line patterns. This would not be the case absent robustness for all three functions; thus in this case, robustness confers flexibility (Fig. 3-4).

We don't claim that this model accounts for every context in which these genes are deployed by developing embryos. In particular, we don't believe the model can account for more complex pattern-formation tasks, such as the production of regularly-spaced bristles, where mutations in components of this network change the spacing. To produce robust longer range patterns, we think this lateral inhibition model must be joined with either a mechanism for long-range cell-cell communication (such as Dl diffusion (Klueg *et al.*, 1998) (Qi *et al.*, 1999)) and/or a mechanism by which N/Dl signaling could feed back on a spatially-varying pre-pattern.

Constitutive Dl production combined with weak lateral inhibition can resolve the controversy about the role of lateral inhibition in neurogenesis

As discussed above, the experimental literature includes both support for, and refutation of, an important role for lateral inhibition in neural determination. Our results can account for both sets of experiments. If the pre-pattern which initiates neuroblast selection is well-tuned, the pre-pattern plus a constant level of inhibition could select the winner, absent lateral inhibition. But lateral inhibition buffers the patterning against perturbations in the initial pre-patterning (e.g. due to genetic or environmental variation, or "developmental noise"). Seugnet and colleagues (Seugnet *et al.*, 1997) reported that with only constant production of *Dl*, 80% of proneural clusters developed normally, but 20% produced an extra NB. We interpret these experiments to say that the pre-pattern is well-tuned in most proneural clusters, but in 20% either a poorly-tuned pre-pattern or

noise causes errors in the absence of lateral inhibition. This is a testable idea. One could remove lateral inhibition as Seugnet et al (Seugnet *et al.*, 1997) did. We would then predict that the embryo would be much more sensitive to hyper- and hypomorphs in pre-patterning genes such as *extramachrochaete* and *hairy*. We would also predict they would be more sensitive to mutations in genes within the network itself, such as missing or extra copies of *Dl* or *N*. We make the latter prediction because those mutations should change the threshold to which the pre-pattern is tuned. In the absence of lateral inhibition, a pre-pattern that was well-tuned to the former threshold could not also be well-tuned to the new threshold.

The *E(spl)* homeostat might increase the network's robustness to noise, as in human-engineered circuits.

From our results above and others not described here, we deduce that *E(spl)* greatly *reduces* the percentage of random parameter sets that enable lateral inhibition. We believe this is because *E(spl)* acts as a homeostat. As the expression level of the proneural genes (*ac* and *sc*) rise, their products activate *E(spl)*. E(spl) then downregulates the proneural genes. As with the thermostat in a house, this negative feedback loop tends to keep the proneural genes at an intermediate level rather than allowing them to switch to either a high or low state. Both E(spl) auto-inhibition, and to a lesser extent *cis*-Dl inhibition of N activation, help overcome this homeostat. On the face of it, this seems a strange design. The *ac/sc* network itself is a bistable switch that tends to go in the direction it is pushed and remain there. The switch and homeostat mechanisms are exact opposites. We found that removing the homeostat (the reduced model) makes it easier to find parameters sets which pass our tests (which all involve throwing the switch). Why incorporate counter-acting mechanisms in the same circuit?

It is, of course, possible this is simply a vestige of the network's evolutionary history, with no design rationale. But electrical engineering suggests one possible advantage. An op-amp is a famous circuit that amplifies the difference between two inputs. Good op-amps can amplify a voltage difference more than a million-fold. Usually, though, engineers add a negative feedback circuit (that is, a homeostat). This greatly attenuates the gain but makes the amplifier much more stable – noise generated internally inside the op-amp will not affect the output signal. Reducing function to gain stability is common in other electrical circuits as well. These electrical circuits do not make good direct analogies to genetic networks, but the concept of adding negative feedback to increase stability might still apply. Perhaps the *E(spl)* homeostat reduces the network's sensitivity to developmental "noise" such as stochastic changes in transcription or translation rates, in the pre-pattern, or in the concentrations of modulators such as Da and Emc.

A related "design" benefit might be that the *E(spl)* homeostat prevents the network from switching individual cells on or off before the pre-pattern has a chance to decree the winner. A simple bistable switch consisting of *ac* and *sc* alone could not help but be thrown in one direction or the other by noise (as apparently takes place in *C. elegans* anchor cell specification). Adding *E(spl)* leads to a new, neither-on-nor-off steady state, which could enable the proneural switch to procrastinate until some extrinsic cue forces the system to choose one or the other switched state.

Conclusions

Our results highlight several evolutionarily interesting properties of genetic networks. Like several other recent simulated networks, this model network, inspired by an evolved design, exhibits robustness that would be the envy of any human designer. Perhaps this is a generic feature of genetic organization, but perhaps it reflects a co-evolution between

evolved networks, biologists, and theorists: modular, robust networks are the easiest to get at experimentally, thus the best-understood, and thus the best fodder for models. In any case, the model discussed here shows that this robustness can lead to an evolutionary flexibility where a network originally tuned to one function can mutate, without losing that original function, towards the ability to perform additional functions. Furthermore, were evolution to discover a relatively non-robust solution (at the tip of an arm in Fig. 3-5e), the network could migrate down the octopus arm to more robust regions. Conversely, the network could migrate back up one of these arms if the characteristics of the central region were no longer selected for. This could lead to reproductive isolation, the precondition for speciation.

Table 3-1. Robustness of neurogenic network models to parameter variation. The second column shows the proportion of randomly selected parameter sets which successfully formed the test pattern. Success rates are based on finding hundreds (reduced and augmented networks) or dozens (standard network) of successful parameter sets. In parentheses is the n^{th} root of this success rate, where n is the number of free parameters in the model. Note that small differences in the n^{th} root lead to large differences in solution frequency because of the high number of parameters. The third column shows percent of successful offspring from 10,000 recombinations of 100–200 randomly found successful parameter sets (standard two-cell used only 16 parental sets).

Network / Task (number of free parameters)	**Success rate for random parameters, f; (f ^1/n)**	**Recombination success**
Augmented / 2-cell; (69)	1 / 3,800; (0.89)	5%
Augmented / 7-cell; (69)	1 / 8,000; (0.88)	5%
Standard / 2-cell; (63)	1 / 113,000; (0.83)	1%
Standard / 7-cell; (63)	1 / 153,000; (0.83)	not done
Reduced / 2-cell; (53)	1 / 570; (0.89)	9%
Reduced / 7-cell; (53)	1 / 2,900; (0.86)	4%
Augmented / 2-cell: Restricted range (69)	1 / 4; (0.98)	58%
Augmented / 7-cell; Restricted range (69)	1 / 34; (0.95)	45%
Variations on augmented network		
Augmented w/o Dl cis-inhibition of N (67)	1 / 7,200; (0.88)	
Augmented w/o E(spl) auto-inhibition (67)	1 / 43,000; (0.86)	
Augmented w/o AC, SC activation of e(spl) (65)	1 / 600; (0.91)	
Augmented with no cooperativity in transcriptional activation/repression (54)	0 / 210,000	
Augmented w/ constant Dl input in all cells (71)	1 / 157,000 (0.84)	

Table 3-2. Robustness of neurogenic network models to variation in initial conditions (pre-pattern). The first 3 rows show the number of solutions to each network which formed the correct final pattern from at least 75% of noisy seven-cell pre-patterns. We tested 100 pre-patterns against 40 solutions per model (only 10 for Standard model). Each pre-pattern was as described in Fig. 3-2, but with the initial ac and sc product concentrations in the focal cells randomly perturbed by a value drawn from a Gaussian distribution with a standard deviation equal to 40% of the initial difference between high and low cells. The last 2 rows show results for the same 40 solutions per model, but starting from initial concentrations drawn from a Gaussian distribution with the same mean for all cells, and a 20% standard deviation (we used only 50 different pre-patterns here). In both cases, we ensured that the middle cell had a higher initial concentration of each product than all surrounding cells (thus there is still effectively a weak pre-pattern in the second case), but the difference could be arbitrarily small.

Network	**Number of solutions found robust to noise**
Augmented w/ prepattern	5 / 40
Standard w/ prepattern	1 / 10
Reduced w/ prepattern	12 / 40
Augmented w/o prepattern	1 / 40
Reduced w/o prepattern	4 / 40

Table 3-3. Proportion of solutions to one test that can pass other tests in the augmented network.

Original Pattern	two-cell	seven-cell	line	All three
two-cell	1.0	0.25	0.14	0.12
seven-cell	0.8	1.0	0.33	0.25

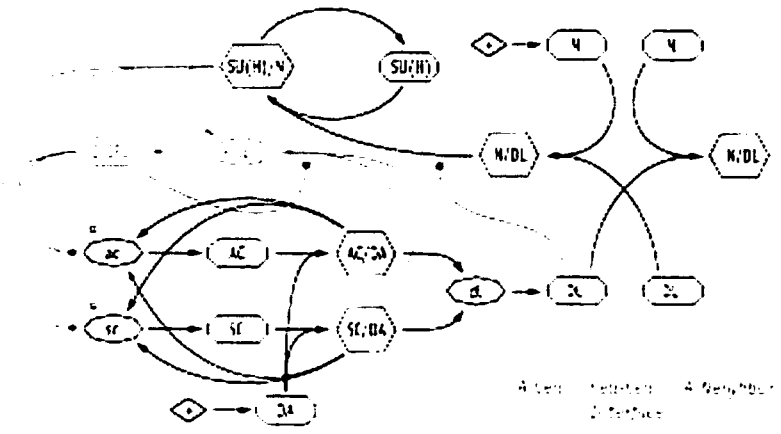


Figure 3-1. Models of the neurogenic network used in neural specification in *D. melanogaster*. A graphical summary of the experimental literature on neural specification, which our model represents mathematically. Ovals are mRNAs, rectangles are proteins, hexagons are protein complexes. Diamonds containing + signs indicate constitutive expression of N and DA where we have not diagrammed, but have modeled, the intermediate mRNA concentrations. Though not diagrammed, all gene products decay with parameter-specified decay rates except Su(H), which we modeled as cycling between two states with no production or decay. Our 'augmented' network model includes the black, green, and red connections. Our 'standard' network model includes the black and green connections. Our 'reduced' network model includes the black and blue connections such that SU(H)/N directly inhibits production of ac and sc with all E(spl) products absent in that version.

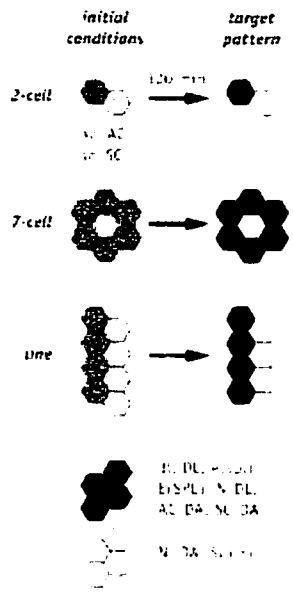


Figure 3-2. Two-cell, seven-cell, and line patterns used to test our models. Each test starts some cells with twice the concentrations of the mRNA and protein of both proneural genes as its neighbors (brighter colors show higher concentrations). We surround the field of competent cells with other cells (not drawn) containing no initial ac/sc products, hence no ac/sc ever. All other model concentrations start either off (black) or at a uniform high level (blue) across all cells. We run the model for 300 minutes and measure whether the initially higher cell(s) achieve(s) a high concentration of AC (greater than 20% of the highest steady-state concentration possible), and the initially lower cell(s) turn off (below 2% of maximum steady state). This could happen without lateral inhibition if there were a threshold of initial ac and sc concentrations above which the ac/sc switch would turn on and below which it would turn off in isolated cells. To ensure the separation is due to lateral inhibition rather than mere thresholding, we *always* run a second test with the same two-fold initial difference between cells, but where the lower concentration cells in the second test have the same initial concentrations of *ac* and *sc* products as the higher cell in the first test (the one exception was in testing the effect of noise in initial conditions where we did not use a second test). We call a model with its parameter values a 'solution' *only* if both pairs achieve the correct pattern at 300 minutes, and if additionally both pairs maintain that state for 1000 minutes. We conducted identical tests using much smaller initial concentration differences (down to 1%) and found qualitatively similar results, albeit with commensurately lower solution frequencies.

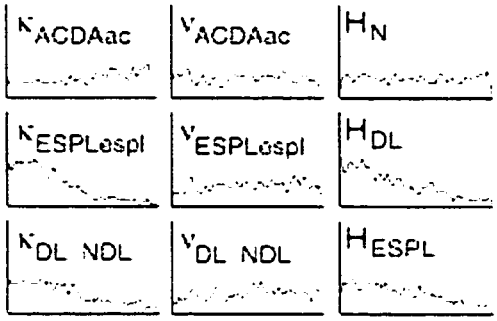


Figure 3-3. Selected histograms of successful parameter values from solutions to the augmented neurogenic network using the 2-cell lateral inhibition test. Each histogram shows the values of just one of the 69 parameters in the model. H's are half-lives, κ's are half-maximal activations, and ν's are cooperativity (Hill) coefficients. The horizontal axes span the range within which we sample each parameter (ranges: H = 1 – 1000 minutes; κ = 0.001 – 1 proportion of max concentration; ν = 1 – 10). We sample H's and κ's on a log scale. We selected these histograms to show the variety of clustering among different parameters (see full suite in Supplement). We found the "best" range for each parameter by choosing values from the smallest sub-range covering at least an order of magnitude and containing 50% of the values used in solutions. We also forced all transcriptional activation and repression in these sets to be at least slightly cooperative.

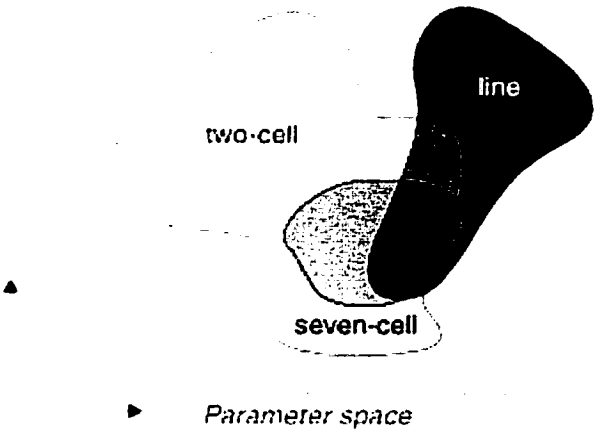


Figure 3-4. Robustness to parameter variation confers evolutionary flexibility in the pattern produced by the network. The cartoon shows a hypothetical cross-section through the parameter space with outlines for regions where the network can form different patterns. The regions overlap approximately as shown. If the neurogenic network was originally tuned to make any one of the three patterns, evolution could modify parameter values, wandering through neighborhoods of working territory, until the network could make the other patterns as well. This is only possible due to the network's robustness to parameter variation; if the pools were too small they would not overlap and natural selection would have to leap from one function to another across non-functional zones.

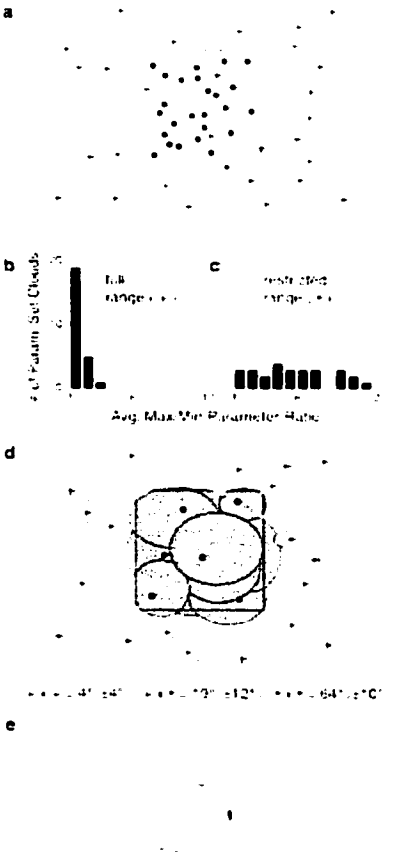


Figure 3-5. The parameter space is shaped like an octopus with multiple possible network functions overlapping. (a) A cartoon of randomly selected solutions from the full ('+') and restricted ('•') parameter regions. After mutational expansion, each of these grows into a population of parameter sets confined within a local "solution volume" (shaded domains in d). The ratio of max/min values for each parameter across all the sets in a population have average values shown in (b) and (c). These ratios indicate that most parameter sets from the restricted range occupy larger solution spaces than those from the full range (cartooned in d). The text under (d) shows the average mating success between parameter sets found within the full range (+ x +), within the restricted range (• x •), and between one parameter set from the restricted range with parameter sets found in the full range (+ x •). This leads to a view of solution space as an octopus, with the solutions found in the full parameter space out in the arms, and a large central area around the restricted range.

Chapter 4
More details on lateral inhibition in the neurogenic network and comparisons with other networks.

Note This chapter is a replica of the supplement to the paper in Chapter 3 (Meir *et al.*, 2002). It is also available online at www.ingeneue.org.

Genes involved in the neurogenic network

The first step in making a model of a genetic network is to construct a "wiring diagram" of the genes and gene products involved and connections between these representing their interactions. This paper concerns the neurogenic network and its role in selecting single neuroblasts (NB) and sensory organ precursors (SOP) from proneural clusters in *Drosophila* embryos and their imaginal disks. The next few paragraphs outline the central loop of the network used in this process, as understood currently in the literature. There follow discussions of some of the important genes outside this loop which influence the positions of the SOP's. The core genes and their interactions are all summarized in the diagram in Figure 3-1. Many other reviews exist of various pieces of the neurogenic network. Here we attempt a thorough summary of current literature to justify all the connections in Figure 3-1 and some of the modeling choices we make in the main text. Readers unfamiliar with the neurogenic network may find it more profitable to skip this section and simply refer to Figure 3-1.

Two genes, *achaete* (*ac*) and *scute* (*sc*), appear to be the primary determinants of neural fate. Cells expressing high levels of AC and SC proteins become neural while others adopt an epidermal fate (Campuzano and Modolell, 1992). Ac and Sc both belong to a family of transcription factors which contain a conserved basic helix-loop-helix region (bHLH). The family members use the bHLH domains to dimerize with one another

(Alifragis *et al.*, 1997; Gigliani *et al.*, 1996), becoming transcriptionally active only as dimers. Ac and Sc both bind to the bHLH cofactor Daughterless (Da) and, as heterodimers with Da, each of these can activate both its own and each other's transcription (Cabrera and Alonso, 1991; Martinez and Modolell, 1991; Skeath and Carroll, 1991; Van Doren *et al.*, 1992).

Both Ac and Sc, as heterodimers with Da, directly activate transcription of the *Delta* (*Dl*) gene (Haenlin *et al.*, 1994; Hinz *et al.*, 1994; Kunisch *et al.*, 1994). Delta protein is a transmembrane ligand which can bind to and activate its receptor protein Notch (N) in neighboring cells (Fehon *et al.*, 1990). There is some evidence that Dl can also bind to N within the membrane of the same cell, though it is not clear whether this intracellular binding can activate N (Fehon *et al.*, 1990). What seems clearer is that intracellular interactions between Dl and N can inhibit N activity in that cell, even in the presence of Dl in neighboring cells (de Celis and Bray, 1997; Heitzler and Simpson, 1993); also see (Klein *et al.*, 1997) for evidence that another N ligand, Serrate, does the same. Recently, both (Klueg *et al.*, 1998) and (Qi *et al.*, 1999) showed that Dl is cleaved at the membrane by the protein Kuzbanian, forming extracellular Dl fragments that are biologically active and exist *in vivo*. These results make it likely that in at least some cases a cleavage product of the Dl protein diffuses and may be able to act at a distance from the cells that expressed it. But as it has yet to be shown that Dl diffusion plays a role in the processes explored in this paper, so we omit Dl diffusion between cells in the model we present here. We include diffusion of both Dl and N between different faces of the same cell.

Notch (N) is a transmembrane receptor molecule produced in nearly constant quantities across broad fields of cells (Fehon *et al.*, 1991). Regulation of *N* production *per se* does not seem to play a role in the neurogenic patterning mechanism (Seugnet *et al.*, 1997),

although in some patterns involving this network the concentration of N is regulated (Huppert *et al.*, 1997). In our model we assume constitutive expression of N in all cells. When N is activated by binding to its ligand Dl, it has the effect of reducing the production of *ac* and *sc*. The members of the pathway between N and *ac/sc* are well known, but the way they interact is still somewhat confusing. There is now fairly good evidence that upon binding Dl, an intracellular portion of N is cleaved and transported to the nucleus (Struhl and Adachi, 1998). This cleaved portion of N may be transcriptionally active on its own, but at least some of Notch's activity is due to interactions with the Suppressor of Hairless (Su(H)) protein, a known transcription factor (Fortini and Artavanis-Tsakonas, 1994; Hsieh *et al.*, 1996). When activated by the cleaved fraction of N, Su(H) in turn upregulates the transcription of some genes in the *Enhancer of split complex* (*E(spl)*), a group of 7 bHLH transcription factors that act as repressors of, among other genes, *ac* and *sc* transcription (Bailey and Posakony, 1995; Eastman *et al.*, 1997; Jimenez and Ish-Horowicz, 1997; Lecourtois and Schweisguth, 1995). This repression of *ac* and *sc* by E(spl) completes the loop.

In the models here, we chose to ignore the growing body of work on the details of N activation and cleavage and model it as a simple dimerization between N and Dl, after which the dimer enters the cytoplasm and interacts with Su(H). This is functionally equivalent to N being cleaved upon contact with Dl, with an intracellular portion becoming active (NDL in Fig. 3-1) and the extracellular portion of N, together with the Dl molecule it bound, degrading with no further activity.

Su(H) is normally found in the nucleus, and is distributed nearly uniformly among cells in each proneural cluster (Gho *et al.*, 1996). Recent work on both peripheral neurogenesis and eye development shows that Su(H) is normally a transcriptional repressor, but

becomes a transcriptional activator when bound by an activated N fragment in the nucleus (Furriols and Bray, 2001; Li and Baker, 2001). Loss of function *Su(H)* mutants do not express *E(spl)* genes, though, even in the presence of activated Notch, and adding *Su(H)* alone produces a slight increase in *E(spl)* transcription (Eastman *et al.*, 1997). Similarly, removing the *Su(H)* repressor Hairless, which binds to and inactivates Su(H), leads to an increase in *E(spl)* production (Brou *et al.*, 1994). Thus in our models here, we have only included the activator function of Su(H) in combination with N, and left out the repression function. Further, for simplicity, our model assumes a never changing amount of SU(H) protein, some of which is complexed with N. This is equivalent to, but simpler by omitting needless differential equations, than having a constant transcription rate for SU(H) mRNA, a constant rate of decay for that mRNA, constant translation of the mRNA into SU(H) protein, and equal decay rates for SU(H) protein alone or complexed with N. In the future, as these interactions are better elucidated, we plan to add a more detailed representation of Su(H) function.

Su(H) binds to the promoter region of several of the *E(spl)* complex genes and, in the presence of activated Notch, Su(H) upregulates transcription of these genes (Bailey and Posakony, 1995; Eastman *et al.*, 1997; Lecourtois and Schweisguth, 1995). In addition to Su(H), these *E(spl)* genes are also activated by Ac and Sc (Hinz *et al.*, 1994; Kramatschek and Campos-Ortega, 1994; Singson *et al.*, 1994). E(spl) proteins are bHLH transcription factors which in turn inhibit the activity of Ac/Da and Sc/Da, both in *E(spl)* promoters and in the *ac/sc* complex promoters (Oellers *et al.*, 1994). Thus E(spl) antagonizes its own production as well as antagonizing production of *ac/sc* complex members. The latter interaction completes the cycle from activated N to downregulation of the proneural genes *ac* and *sc*. For lack of evidence otherwise, we assume that *ac* and *sc* regulation is qualitatively (but not quantitatively) identical.

A few other genes appear to lie within this loop of interactions but their function is not as well understood. There are several other bHLH activator genes such as *lethal of scute* and *atonal* that belong to the same family as *ac* and *sc* and have similar effects when over-expressed. It does not appear that these genes play as much of a role in neural selection as *ac* and *sc*, however, so we do not consider them here. Several genes related to those in the *E(spl)* complex, collectively called the *bearded* family, are activated by *ac* and *sc* (Singson *et al.*, 1994) and as with *E(spl)* genes may act as a repressor of E(spl) and proneural activity (Lai *et al.*, 2000). Lai and Posakony (Lai and Posakony, 1997) argue that *bearded* genes repress translation rather than transcription. The role deduced for *bearded* so far, though, appears to be the same as the way in which we modeled E(spl), so we leave *bearded* out of this first version of our model. That is, we are omitting a known redundancy and we will therefore underestimate the robustness of the whole network.

The loop of genetic interactions outlined above is thought to be the basis of the neurogenic network's pattern formation mechanism as follows. Initially, *ac* and *sc* are turned on in all cells of a proneural cluster by upstream genes. Ac and Sc upregulate production of *Dl*. Dl activates N in neighboring cells and represses N activity in its own cell. The activated N in neighboring cells, through the Su(H) and E(spl) pathway, turns off *ac* and *sc*. If one cell accumulates more *ac* or *sc* than its neighbors, either by random chance or through the action of pre-patterning genes, that cell will produce more Dl than its neighbors, causing more N to be activated in neighboring cells. This in turn will cause the *ac* and *sc* production in neighboring cells to decrease, increasing the difference in *ac* and *sc* concentrations in the focal cell compared to its neighbors. Once begun, this process snowballs to high *ac* and *sc* concentrations in one focal cell, and very low concentrations in all of that cells neighbors, thus selecting one cell out from the group.

Both *daughterless* (*da*) and *extramachrochaete* (*emc*) are bHLH proteins that form heterodimers with Ac and Sc. Da / Ac and Da / Sc heterodimers are transcriptional activators as outlined above. Da homodimers are also transcriptionaly active and upregulate the production of *ac* and *sc*, though in a much weaker way than Da / Ac and Da / Sc heterodimers (Cabrera and Alonso, 1991). Emc lacks a DNA binding region, so it represses Ac and Sc activity by binding to them to form an inactive complex (Martinez *et al.*, 1993; Van Doren *et al.*, 1991; Van Doren *et al.*, 1992). *emc* (Van Doren *et al.*, 1992) does not appear to be regulated by any of the genes in neurogenic loop. *da* is expressed ubiquitously and at fairly uniform levels throughout the tissues involved in neurogenesis (Cronmiller and Cummings, 1993; Vaessin *et al.*, 1994) and we also cannot find any evidence that it is regulated by any of the genes in the neurogenic loop. *emc*, on the other hand, is expressed in a pattern complementary to that of the proneural clusters (Van Doren *et al.*, 1992). Although neither of these is transcriptionally regulated by genes in the central loop, the concentrations of both proteins will be regulated by dimerization with Ac and Sc. We chose to include *da*, but not *emc*, in our model, because we believed *da* should tend to make the model less stable while *emc* should tend to make it more stable. We therefore think our study underestimates the robustness of the neurogenic network. It would be interesting to include *emc* in a future model.

Several other genes outside this central neurogenic loop appear to play important roles in the patterning of neural cells. Among these, the genes *hairy* (Van Doren *et al.*, 1994) and *Hairless* (Bang *et al.*, 1995) both appear to be important in laying down the pre-pattern for neural selection. Other less well studied genes, such as members of the *iroquois* complex (Leyns *et al.*, 1996), *msh* (D'Alessio and Frasch, 1996), and *vnd* (Skeath *et al.*, 1994), are also involved in this pre-patterning or in interactions with the neurogenic

genes in other patterning processes. These latter genes will not be discussed here, but the results from our study should at least partially extrapolate to them. When, in our model, we prescribe initial conditions that boost expression of Ac and Sc in one or a few cells we assume this bias comes from the prior action of genes like *hairy*, *Hairless*, *iroquois*, *msh*, and *vnd*.

Parameters are not restricted in their values

Figure 4-1 shows histograms of the parameter values from all solutions we found while randomly searching parameter space using the 2-cell test on the augmented neurogenic network. In this search, we picked all parameters except cooperativities and activation strengths from ranges spanning at least 2, and often 3 or 4 orders of magnitude (the latter two spanned a single order of magnitude). For instance, the half-life of each component could range from 1 to 1000 minutes. Although some parameters tend to cluster in part of their range, there do not appear to be any absolute restrictions. The parameter histograms for the segment polarity network looks similar to Fig. 4-1, with no absolute restrictions (von Dassow *et al.*, 2000). This lack of restrictions could be due to compensation between pairs or groups of parameters. In other words parameter A can be high as long as parameter B is low and vice-versa. However, we failed to find any strong pair-wise cross-correlations for either network. We enforced some weak correlations during our parameter search to avoid physically unrealistic corners of parameter space (where the ODE integrations ran extremely slowly because of unrealistically high dimerization rates). Excluding those, the highest cross-correlations we found were on the order of $r = 0.3$. So any compensation must be relatively weak, or, more likely, involve groups of more than two parameters.

The neurogenic and segment polarity networks are both astonishingly insensitive to parameter variation.

We did traditional parameter sensitivity analyses on 25 solutions from each of our three neurogenic network models forming the two-cell pattern. For comparison, we also did this sensitivity analysis on 25 solutions from the segment polarity network. In each solution, we varied one parameter at a time through 40 steps across its full range (not restricted as in the previous section) while keeping all other parameter values fixed. At each parameter value we calculated a quantitative score representing how close the network came to passing the test, and picked a threshold for this score which separated good solutions (by our subjective judgement) from bad ones. In our previous paper on the segment polarity network, we showed that parameters could usually be varied quite far without changing the pattern (von Dassow *et al.*, 2000). The full neurogenic network shows a slightly higher sensitivity to parameter values. We counted the number of parameters that were restricted to less than an order of magnitude variation across all solutions for each network (excluding parameters where our original ranges were one order of magnitude or less). Whereas only 19% of parameters in the segment polarity network were limited to less than one order of magnitude variation, 33% of the full neurogenic network parameters were so restricted (Fig 4-2a). Conversely, 44% of the segment polarity parameters could vary across the whole range we gave for them without losing the pattern, while only 31% of the neurogenic network parameters could so vary (Fig 4-2b). The reduced neurogenic network showed similar sensitivities to the augmented neurogenic network. The standard neurogenic network was more sensitive to parameter variation than either of the others, and significantly more sensitive than the reduced network. These sensitivities are consistent with the success rates of randomly chosen parameter sets.

The above results indicate that the model neurogenic network may be slightly more sensitive to parameter variations than the model segment polarity network. But there are more significant differences among certain classes of parameters. The biggest difference is in parameters governing dimerizations. There are many more of these in the neurogenic network, and 50% of these parameters could not be varied by an order of magnitude around their original value (Fig 4-2a). By contrast, the parameters associated with the one dimerization step in the segment polarity network were only so limited 5% of the time. The segment polarity network, on the other hand, seems more sensitive to parameters governing transport processes, whereas the neurogenic network is almost completely insensitive to variations of these parameters (which in the neurogenic network only includes intracellular diffusion of Dl and N among neighboring faces of each cell). Other classes of parameters such as half-lives have similar sensitivities in both networks.

Recombination and mutational expansion algorithms

Sexual mating produces offspring with one allele from each parent. The phenotype can be an equal blend of the effects of each parent allele (Fig. 4-3a), can be determined by only the dominant allele (Fig. 4-3b), or can lie between these extremes. In meiotic recombination, the gamete receives only one parental allele (Fig. 4-3b). As shown in Figure 4-3, meiotic recombination produces offspring that are more dissimilar to the parents than a blended mating, and we thus chose this form of mating for our "recombination" test of robustness.

Mutational expansion adds an evolutionary component to the recombination test. We start with a single successful parameter set and make several offspring parameter sets by introducing +/- 10% magnitude random "mutations" in a randomly selected 10% of the parameters. These changes in value might correspond to single amino acid changes that have small effects on the binding rates of the proteins, etc. We then ask whether each of

these offspring parameter sets can pass our chosen test. We keep the ones that are successful as a new group of parent parameter sets. We then recombine this generation of parents to form a new group of offspring, again introducing mutations into the parameter values of each offspring as above. After discarding the unsuccessful offspring, we keep up to 200 of the successful offspring to form the new group of parents. We aim to produce 800 successful offspring each round, from which we randomly decide which of the successful ones to keep as the next parents with a bias favoring those whose parents had parameter values that were most distant from each other. We introduce this bias that favors a surrogate for "hybrid vigor" to prevent our simulated population of parameter sets from contracting toward a small-diameter set (within which recombination would be successful trivially). We continue this process until less than 90% of random "matings" of the 200 parents result in successful offspring parameter sets. At that point, we measure the ratio between the highest and lowest value for each parameter, and take the average of the ratios across all parameters as our measure of the "size" of the neighborhood in parameter space within which random "matings" almost always produce viable offspring networks. Stopping the process when fewer than 90% of matings fail (rather than 92%), or keeping the size of our population of networks hovering near 200 (rather than 400 or 10,000), etc., are details, decided by the computational capacity we have access to, that do not much matter, we believe. The following flow chart gives an outline of the mutational expansion algorithm we used to make a cloud of parameter sets that expands around one successful parameter set.

1. Make 8 offspring parameter sets from the original parameter set, where each parameter in each offspring has a 10% chance of having a random multiplicative change (mutation) of up to ±10% in its value relative to the parent value.
2. Test each offspring parameter set for its ability to make the pattern in question and keep only those that succeed. This creates a new pool of parent parameter sets.

3. From each new parent, create *n* offspring by mating that parent with another, randomly chosen parent. *n* initially equals 8. "Mating" means randomly selecting half the parameter values from one parent and the other half from the other (Fig 4-3b). Also mutate each offspring's parameter values as in (1).

4. Test each of the new offspring for its ability to make the pattern in question and keep only those that succeed.

5. If there are more than 200, decide randomly which to keep, but with a bias favoring those whose parents were farthest apart in parameter space. Distance is a scaled Euclidean distance between the parameter vectors of each parent, where each parameter's value is scaled between 0 and 1 based on the ranges we used when picking that parameter's value randomly.

6. If there were less than 800 successful offspring, increase *n* by 1. If there were more than 800 successful offspring, decrease *n* by 1.

7. Repeat 3–6 until the percentage of offspring that are successful falls below 90%.

A proposal for how to conduct a saturating search for robust neighborhoods in parameter space.

The parameter space in gene network models is huge (i.e. of large dimension). No realistic depiction of gene networks can possibly escape this difficulty. With 69 parameters, it would be just barely possible to explore all combinations of a high and a low value for each parameter on one of our labs current computers within the life-time of the universe. Therefore in order to explore gene network models at all we need either to be lucky enough to work with extraordinarily robust networks, as we have been lucky so far, or we need sophisticated methods for navigating parameter space. The following mutational expansion method we propose for finding the robustness to parameter variation is a prototype for a technique which should be practically computable. We start

with one successful parameter set and use mutational expansion to find a cloud of parameter sets around this first one. When we have expanded the cloud as far as possible, we randomly find a new successful parameter set. We then cross this new parameter set with all the parameter sets in the first cloud. If the new one crosses successfully with a large proportion of the cloud, we add it to the cloud. If it cannot cross successfully with most parameter sets in the cloud, we assume that it lies outside the cloud. In the latter case, we use mutational expansion to find a cloud of parameter sets around this new set. We now have two distinct clouds of successful parameters in the space. We continue picking new random successful parameters, mating them with each of the existing clouds, and forming a new cloud around those which cannot successfully mate with any existing clouds. This is analogous to the Gram-Schmidt orthonormalization procedure.

If there are a small number of "good" regions in the parameter space then the probability that a new random parameter set will *not* belong to an already existing cloud will decrease over time. By extrapolating the curve of this probability over number of random parameter sets sampled, we can make a good guess at how many distinct successful neighborhoods are contained in the parameter space, and perhaps find the majority of the largest ones.

Based on the limited number of mutational expansion computations we did, the procedure outlined above would certainly take vast computational resources, but potentially an amount that a reasonable budget could afford. The process outlined here would be easiest in the case of a network for which solutions were easily found by random search, as for our models of the segment polarity and neurogenic networks, but would be a valuable assessment no matter how one discovered the founder sets. The end result would be a description of how many qualitatively different ways a network can

form a particular pattern and how far apart these different strategies are from each other. This is interesting both evolutionarily and practically. From an evolutionary perspective, these neighborhoods define the adaptive landscape within which natural selection functions. From a more practical perspective, measuring the values of parameters in a genetic network is hard and almost never done. It is unlikely that anyone will ever measure all 69 parameters in our full neurogenic network. But to find the neighborhood in parameter space where the real network sits might be possible after measuring only a few parameter values. The above analysis tells the experimentalist which parameters are most important to measure in order to distinguish between the different potential mechanisms.

We finish this discussion with an appeal to mathematicians for help. The abstract concept we advance above is to find (by *some* computation, practical in high dimensional parameter space) boundaries of the largest possible neighborhood surrounding a single "founder" parameter set within which network function would be heritable through random recombinatorial "mating", then to use some measure of the volume/shape of that neighborhood as a biologically meaningful measure of network robustness (in the vicinity of the founder parameter set). The computational algorithm for finding this neighborhood need have no biological realism. We indeed used a computational algorithm whose steps mimic evolution to "feel out" the neighborhood because we suspect *we* can not do better than to emulate natural selection. It has been navigating such parameter spaces for billions of years and has likely found an efficient way to wend its way. But other algorithms than we used might be much faster on computers.

In addition to the challenge of conceiving a good computational algorithm, there is a conceptually deeper problem whose solution eludes us so far. We have suggested above

the rough outlines of one possible way toward meeting the following challenge in topology/dynamical systems: to assess the robustness of a dynamical system model of a genetic network by characterizing the shapes, sizes, and arrangements of the loci of robust function in parameter space within which not only does the model work, but it works as well for almost all the possible "offspring" parameter sets. This characterization of a zone of robust function bears some resemblance to the standard concept of convexity. The concept of *structural stability* is almost relevant here, but not quite because whether the "right" pattern forms does not hinge on the signs of eigenvalue real parts at critical points of the dynamical system. This is because we accept time-oscillatory solutions - limit cycles or strange attractors - as "successful" patterns provided the attractors have sufficiently small amplitude. What we really need is a formal way to define -- from a network's connections -- a *topology* on the parameter vector space whose open neighborhoods delineate who can breed with whom to produce "viable" offspring (networks). Different points in the parameter space will cause the dynamical system to make different ensembles of spatial patterns (i.e. different attractors). Some points will cause the system to have no attractors at all. Thus we suspect that, in order to define the topology we seek, we must first pick a particular pattern (attractor), and understand that different patterns induce different topologies on parameter space. In any case, what we seek are ideas for defining a proper topology on parameter space whose open neighborhoods correspond formally to our casual use of the term "neighborhoods" in the main text.

The equations for the augmented model

The equations in the models presented here use the same general formulation and non-dimensionalization as in Dassow et al (2000) (von Dassow *et al.*, 2000), and further

described in (Meir *et al.*, in press). Using those forms, here are the equations for the 'augmented' model discussed in the paper.

We represent transcriptional activation by a single activator using a sigmoidal function

$$\Phi(X,\kappa_X,\nu_X) = \left(\frac{X^{\nu_X}}{\kappa^{\nu_X} + X^{\nu_X}}\right)$$

where X represents the activator concentration, κ is the activator concentration at which transcription proceeds at half its maximal rate, and ν is the degree of non-linearity in the activation function (which we refer to as cooperativity). The concentrations of each element in the model are scaled so that 1 is the maximum achievable concentration, and likewise the production terms are scaled so 1 is their maximum value. We then represent repression with a backwards sigmoid function:

$$\Psi(X,\kappa_X,\nu_X) = 1 - \Phi(X,\kappa_X,\nu_X)$$

Where production of an mRNA species is controlled by more than one activator, we use a probabilistic formulation to combine these activators (Meir *et al.*, in press):

$$\Gamma(X_1,X_2,X_3,\ldots) = 1 - (1 - \frac{\alpha_1}{Max(\alpha_1,\alpha_2,\ldots)}\Phi(X_1,\kappa_{X1},\nu_{X1}))(1 - \frac{\alpha_2}{Max(\alpha_1,\alpha_2,\ldots)}\Phi(X_2,\kappa_{X2},\nu_{X2}))(\ldots)$$

$$= 1 - \prod_i (1 - \frac{\alpha_i}{Max(\alpha)}\Phi(X_i,\kappa_{Xi},\nu_{Xi}))$$

Using these formulas, here are the full equations for the 'augmented' model discussed in the paper. We use the above formulas as a shorthand, with the assumption that all parameters in the formulas above are included (for instance, $\Gamma(X,Y)$ would have a κ, ν, and α for each of the species X and Y). The subscript i indicates the cell number,

and the subscript j indicates the face of the cell for membrane bound components (each cell has 6 faces).

$$\frac{dn_i}{d\tau} = \frac{T_0}{H_n}(\Phi(Const, \kappa_{C,n}, \nu_{C,n}) - n_i)$$

$$\frac{dN_{i,j}}{d\tau} = T_0\begin{bmatrix} \frac{1}{H_N}\left(\frac{n_i}{6} - N_{i,j}\right) - C_{NDL}X_{DL}N_{i,j}DL_{opp}\Psi(DL, \kappa_{DL\_NDL}, \nu_{DL\_NDL}) \\ -r_N(2N_{i,j} - N_{i,j-1} - N_{i,j+1}) \end{bmatrix}$$

$$\frac{ddl_i}{d\tau} = \frac{T_0}{H_{dl}}(\Gamma(ACDA, SCDA) - dl_i)$$

$$\frac{dDL_{i,j}}{d\tau} = T_0\begin{bmatrix} \frac{1}{H_{DL}}\left(\frac{dl_i}{6} - DL_{i,j}\right) - C_{NDL}X_N DL_{i,j}N_{opp}\Psi(DL, \kappa_{DL\_NDL}, \nu_{DL\_NDL}) \\ -r_{DL}(2DL_{i,j} - DL_{i,j-1} - DL_{i,j+1}) \end{bmatrix}$$

$$\frac{dNDL_{i,j}}{d\tau} = T_0\begin{bmatrix} \frac{-NDL_{i,j}}{H_{NDL}} + C_{NDL}X_{DL}N_{i,j}DL_{opp}\Psi(DL, \kappa_{DL\_NDL}, \nu_{DL\_NDL}) \\ -C_{NDL\_SUH}X_{SUH}NDL_iSUH_i \end{bmatrix}$$

$$\frac{dac_i}{d\tau} = \frac{T_0}{H_{ac}}(\Gamma(ACDA, SCDA)\Psi(ESPL, \kappa_{ESPL,ac}, \nu_{ESPL,ac}) - ac_i)$$

$$\frac{dAC_i}{d\tau} = T_0\left[\frac{1}{H_{AC}}(ac_i - AC_i) - C_{ACDA}X_{DA}AC_iDA_i\right]$$

$$\frac{dsc_i}{d\tau} = \frac{T_0}{H_{sc}}(\Gamma(ACDA, SCDA)\Psi(ESPL, \kappa_{ESPL,sc}, \nu_{ESPL,sc}) - sc_i)$$

$$\frac{dSC_i}{d\tau} = T_0\left[\frac{1}{H_{SC}}(sc_i - SC_i) - C_{SCDA}X_{DA}SC_iDA_i\right]$$

$$\frac{dda_i}{d\tau} = \frac{T_0}{H_{da}}(\Phi(C, \kappa_{C,da}, \nu_{C,da}) - da_i)$$

$$\frac{dDA_i}{d\tau} = T_0\left[\frac{1}{H_{DA}}(da_i - DA_i) - C_{ACDA}X_{AC}AC_iDA_i - C_{SCDA}X_{SC}SC_iDA_i\right]$$

$$\frac{despl_i}{d\tau} = \frac{T_0}{H_{espl}}\begin{bmatrix} \Gamma\left[SUH\_N, \left[\Gamma(ACDA, SCDA)\Psi(ESPL, \kappa_{ESPL,espl}, \nu_{ESPL,espl})\right]\right] \\ -espl_i \end{bmatrix}$$

$$\frac{dESPL_i}{d\tau} = \frac{T_0}{H_{ESPL}}(espl_i - ESPL_i)$$

$$\frac{dACDA_i}{d\tau} = T_0\left[\frac{-ACDA_i}{H_{ACDA}} + C_{ACDA}X_{DA}AC_iDA_i\right]$$

$$\frac{dSCDA_i}{d\tau} = T_0\left[\frac{-SCDA_i}{H_{SCDA}} + C_{SCDA}X_{DA}SC_iDA_i\right]$$

$$\frac{dSUH_i}{d\tau} = T_0\left[\frac{SUH\_N}{H_{SUH\_N}} - C_{NDL\_SUH}X_{NDL}NDL_iSUH_i\right]$$

$$\frac{dSUH\_N_i}{d\tau} = T_0\left[-\frac{SUH\_N}{H_{SUH\_N}} + C_{NDL\_SUH}X_{SUH}NDL_iSUH_i\right]$$

$$\frac{dConst_i}{d\tau} = 0$$

Figure 4-1. Histograms of successful parameter values from solutions to the augmented neurogenic network with the 2-cell lateral inhibition test. Each histogram shows the values of a single one of the 69 parameters in the model. The horizontal axes span the range within which we picked each parameter and is plotted on a log-scale for all parameters except cooperativity (Hill) coefficients (v) and relative activator strengths (α). The vertical axes show the number of successful parameter sets where that parameter fell within each bin along the horizontal axis. We made these plots from several hundred parameter sets. No parameter has an absolute restriction on its value except the half-life of AC. The latter restriction arises because our test involves checking for AC concentrations close to 0 within 300 minutes – thus we found no parameter sets with AC half-life > 200 minutes. Many other parameters, however, concentrate their values within certain ranges. We made our "restricted" parameter ranges based on those tendencies.

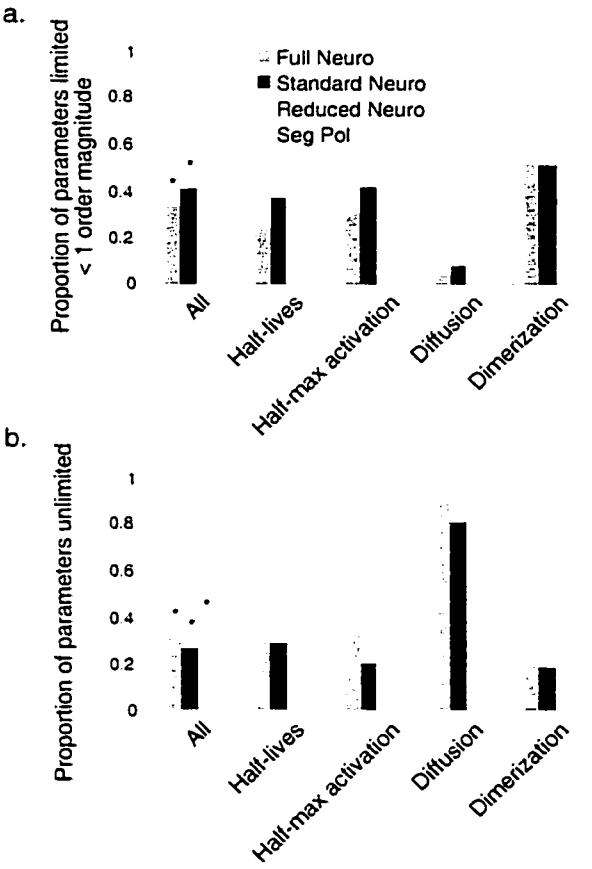


Figure 4-2. Sensitivity analyses for parameters in each network. We randomly selected 25 successful parameter sets that passed the 2-cell test in each of the 3 neurogenic networks, and 25 successful parameter sets from the segment polarity network. For each of these, we varied each parameter individually across its full range, while holding all other parameters constant. (a) shows the proportion of parameters whose values could not vary by an order of magnitude around the original value without losing the ability to form the pattern. (b) shows the proportion of parameters who could successfully form the pattern across the whole of their range. The colors of the bars correspond to the different networks as indicated in the legend. In both graphs, we show results from all parameters together first, then show results from four distinct groups of parameters. * in the "All" column indicates proportions significantly different ($p < 0.05$) from the segment polarity network. We did not perform statistics on the other columns.

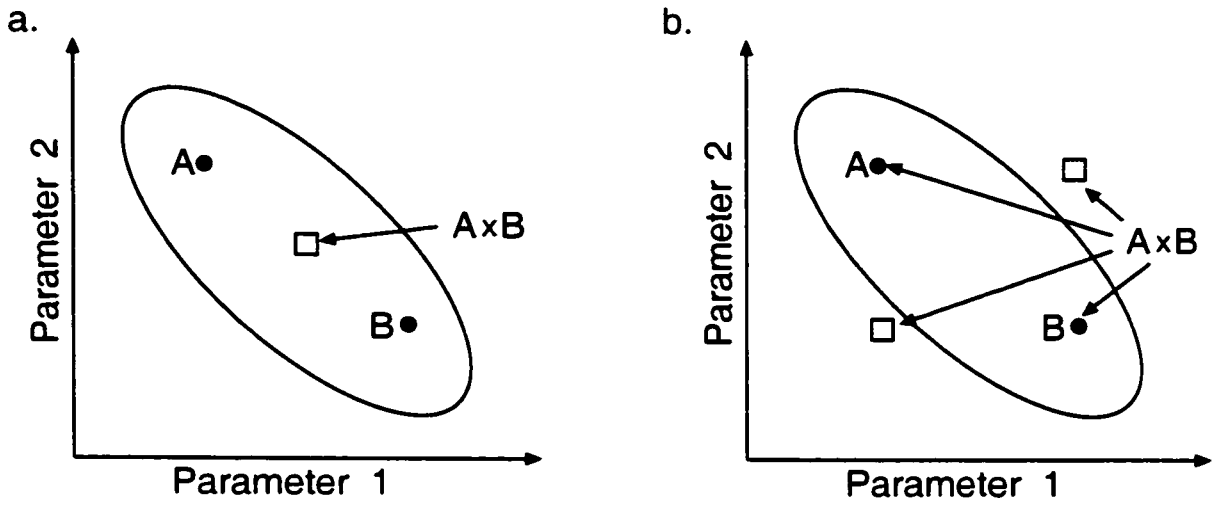


Figure 4-3. Two forms of crosses between parameter sets. Each parent contributes one parameter "allele" to the offspring. In sexual mating, the offspring contains both alleles and the parameter value used by the offspring could be either one of those values (one allele is "dominant") or could be intermediate between the two. (a) shows mating parameter sets A and B, each containing two parameters, where the offspring receives the average of its parents alleles. In producing gametes through meiosis, recombination between parent chromosomes produces a new chromosome that receives only one of the parent alleles, randomly chosen from the two parent alleles (b). We chose the latter mechanism because, as shown in the diagrams, it is a stronger test of robustness - a group of parameter sets that are robust via recombination are likely also to be robust via diploid mating.

Literature Cited

Alifragis, P., Poortinga, G., Parkhurst, S. M. and Delidakis, C. (1997). A network of interacting transcriptional regulators involved in Drosophila neural fate specification revealed by the yeast two-hybrid system. *Proc Natl Acad Sci U S A* **94,** 13099-104.

Aza-Blanc, P., Ramirez-Weber, F. A., Laget, M. P., Schwartz, C. and Kornberg, T. B. (1997). Proteolysis that is inhibited by hedgehog targets Cubitus interruptus protein to the nucleus and converts it to a repressor. *Cell* **89,** 1043-53.

Bailey, A. M. and Posakony, J. W. (1995). Suppressor of hairless directly activates transcription of enhancer of split complex genes in response to Notch receptor activity. *Genes Dev* **9,** 2609-22.

Bang, A. G., Bailey, A. M. and Posakony, J. W. (1995). Hairless promotes stable commitment to the sensory organ precursor cell fate by negatively regulating the activity of the Notch signaling pathway. *Dev Biol* **172,** 479-94.

Barkai, N. and Leibler, S. (1997). Robustness in simple biochemical networks [see comments]. *Nature* **387,** 913-7.

Barkai, N. and Leibler, S. (2000). Circadian clocks limited by noise. *Nature* **403,** 267-8.

Bray, D. (1995). Protein molecules as computational elements in living cells [published erratum appears in Nature 1995 Nov 23;378(6555):419]. *Nature* **376,** 307-12.

Bray, D., Levin, M. D. and Morton-Firth, C. J. (1998). Receptor clustering as a cellular mechanism to control sensitivity [see comments]. *Nature* **393,** 85-8.

Brou, C., Logeat, F., Lecourtois, M., Vandekerckhove, J., Kourilsky, P., Schweisguth, F. and Israel, A. (1994). Inhibition of the DNA-binding activity of Drosophila suppressor of hairless and of its human homolog, KBF2/RBP-J kappa, by direct protein- protein interaction with Drosophila hairless. *Genes Dev* **8,** 2491-503.

Cabrera, C. V. and Alonso, M. C. (1991). Transcriptional activation by heterodimers of the achaete-scute and daughterless gene products of Drosophila. *Embo J* **10,** 2965-73.

Campuzano, S. and Modolell, J. (1992). Patterning of the Drosophila nervous system: the achaete-scute gene complex. *Trends Genet* **8,** 202-8.

Collier, J. R., Monk, N. A., Maini, P. K. and Lewis, J. H. (1996). Pattern formation by lateral inhibition with feedback: a mathematical model of delta-notch intercellular signalling. *J Theor Biol* **183,** 429-46.

Cronmiller, C. and Cummings, C. A. (1993). The daughterless gene product in Drosophila is a nuclear protein that is broadly expressed throughout the organism during development. *Mech Dev* **42,** 159-69.

D'Alessio, M. and Frasch, M. (1996). msh may play a conserved role in dorsoventral patterning of the neuroectoderm and mesoderm. *Mech Dev* **58,** 217-31.

de Celis, J. F. and Bray, S. (1997). Feed-back mechanisms affecting Notch activation at the dorsoventral boundary in the Drosophila wing. *Development* **124,** 3241-51.

de Celis, J. F., Garcia-Bellido, A. and Bray, S. J. (1996). Activation and function of Notch at the dorsal-ventral boundary of the wing imaginal disc. *Development* **122,** 359-69.

Doe, C. Q. and Goodman, C. S. (1985). Early events in insect neurogenesis. II. The role of cell interactions and cell lineage in the determination of neuronal precursor cells. *Dev Biol* **111,** 206-19.

Eastman, D. S., Slee, R., Skoufos, E., Bangalore, L., Bray, S. and Delidakis, C. (1997). Synergy between suppressor of Hairless and Notch in regulation of Enhancer of split m gamma and m delta expression. *Mol Cell Biol* **17,** 5620-8.

Fehon, R. G., Johansen, K., Rebay, I. and Artavanis-Tsakonas, S. (1991). Complex cellular and subcellular regulation of notch expression during embryonic and imaginal development of Drosophila: implications for notch function. *J Cell Biol* **113,** 657-69.

Fehon, R. G., Kooh, P. J., Rebay, I., Regan, C. L., Xu, T., Muskavitch, M. A. and Artavanis-Tsakonas, S. (1990). Molecular interactions between the protein products of the neurogenic loci Notch and Delta, two EGF-homologous genes in Drosophila. *Cell* **61,** 523-34.

Fortini, M. E. and Artavanis-Tsakonas, S. (1994). The suppressor of hairless protein participates in notch receptor signaling. *Cell* **79,** 273-82.

Furriols, M. and Bray, S. (2001). A model Notch response element detects Suppressor of Hairless-dependent molecular switch. *Curr Biol* **11,** 60-4.

Gho, M., Lecourtois, M., Geraud, G., Posakony, J. W. and Schweisguth, F. (1996). Subcellular localization of Suppressor of Hairless in Drosophila sense organ cells during Notch signalling. *Development* **122,** 1673-82.

Gigliani, F., Longo, F., Gaddini, L. and Battaglia, P. A. (1996). Interactions among the bHLH domains of the proteins encoded by the Enhancer of split and achaete-scute gene complexes of Drosophila. *Mol Gen Genet* **251,** 628-34.

Haenlin, M., Kramatschek, B. and Campos-Ortega, J. A. (1990). The pattern of transcription of the neurogenic gene Delta of Drosophila melanogaster. *Development* **110,** 905-14.

Haenlin, M., Kunisch, M., Kramatschek, B. and Campos-Ortega, J. A. (1994). Genomic regions regulating early embryonic expression of the Drosophila neurogenic gene Delta. *Mech Dev* **47,** 99-110.

Hartwell, L. H., Hopfield, J. J., Leibler, S. and Murray, A. W. (1999). From molecular to modular cell biology. *Nature* **402,** C47-52.

Heitzler, P. and Simpson, P. (1993). Altered epidermal growth factor-like sequences provide evidence for a role of Notch as a receptor in cell fate decisions. *Development* **117,** 1113-23.

Hinz, U., Giebel, B. and Campos-Ortega, J. A. (1994). The basic-helix-loop-helix domain of Drosophila lethal of scute protein is sufficient for proneural function and activates neurogenic genes. *Cell* **76,** 77-87.

Hsieh, J. J., Henkel, T., Salmon, P., Robey, E., Peterson, M. G. and Hayward, S. D. (1996). Truncated mammalian Notch1 activates CBF1/RBPJk-repressed genes by a mechanism resembling that of Epstein-Barr virus EBNA2. *Mol Cell Biol* **16,** 952-9.

Huppert, S. S., Jacobsen, T. L. and Muskavitch, M. A. (1997). Feedback regulation is central to Delta-Notch signalling required for Drosophila wing vein morphogenesis. *Development* **124,** 3283-91.

Jimenez, G. and Ish-Horowicz, D. (1997). A chimeric enhancer-of-split transcriptional activator drives neural development and achaete-scute expression. *Mol Cell Biol* **17,** 4355-62.

Kauffman, S. A. (1993). The origins of order : self-organization and selection in evolution, pp. xviii, 709. New York: Oxford University Press.

Kirschner, M. and Gerhart, J. (1998). Evolvability. *Proc Natl Acad Sci U S A* **95,** 8420-7.

Klein, T., Brennan, K. and Arias, A. M. (1997). An intrinsic dominant negative activity of serrate that is modulated during wing development in Drosophila. *Dev Biol* **189,** 123-34.

Klueg, K. M., Parody, T. R. and Muskavitch, M. A. (1998). Complex proteolytic processing acts on Delta, a transmembrane ligand for Notch, during Drosophila development. *Mol Biol Cell* **9,** 1709-23.

Kooh, P. J., Fehon, R. G. and Muskavitch, M. A. (1993). Implications of dynamic patterns of Delta and Notch expression for cellular interactions during Drosophila development. *Development* **117,** 493-507.

Kramatschek, B. and Campos-Ortega, J. A. (1994). Neuroectodermal transcription of the Drosophila neurogenic genes E(spl) and HLH-m5 is regulated by proneural genes. *Development* **120,** 815-26.

Kunisch, M., Haenlin, M. and Campos-Ortega, J. A. (1994). Lateral inhibition mediated by the Drosophila neurogenic gene delta is enhanced by proneural proteins. *Proc Natl Acad Sci U S A* **91,** 10139-43.

Lai, E. C., Bodner, R., Kavaler, J., Freschi, G. and Posakony, J. W. (2000). Antagonism of notch signaling activity by members of a novel protein family encoded by the bearded and enhancer of split gene complexes. *Development* **127,** 291-306.

Lai, E. C. and Posakony, J. W. (1997). The Bearded box, a novel 3' UTR sequence motif, mediates negative post- transcriptional regulation of Bearded and Enhancer of split Complex gene expression. *Development* **124,** 4847-56.

Laub, M. T. and Loomis, W. F. (1998). A molecular network that produces spontaneous oscillations in excitable cells of Dictyostelium. *Mol Biol Cell* **9,** 3521-32.

Lecourtois, M. and Schweisguth, F. (1995). The neurogenic suppressor of hairless DNA-binding protein mediates the transcriptional activation of the enhancer of split complex genes triggered by Notch signaling. *Genes Dev* **9,** 2598-608.

Leyns, L., Gomez-Skarmeta, J. L. and Dambly-Chaudiere, C. (1996). iroquois: a prepattern gene that controls the formation of bristles on the thorax of Drosophila. *Mech Dev* **59,** 63-72.

Li, Y. and Baker, N. E. (2001). Proneural enhancement by Notch overcomes Suppressor-of-Hairless repressor function in the developing Drosophila eye. *Curr Biol* **11,** 330-8.

Martinez, C. and Modolell, J. (1991). Cross-regulatory interactions between the proneural achaete and scute genes of Drosophila. *Science* **251,** 1485-7.

Martinez, C., Modolell, J. and Garrell, J. (1993). Regulation of the proneural gene achaete by helix-loop-helix proteins. *Mol Cell Biol* **13,** 3514-21.

McAdams, H. H. and Shapiro, L. (1995). Circuit simulation of genetic networks [see comments]. *Science* **269,** 650-6.

Meir, E., von Dassow, G., Munro, E. and Odell, G. M. (2002). Robustness, flexibility, and the role of lateral inhibition in the neurogenic network. *Curr Biol* **12,** 778-86.

Meir, E., von Dassow, G., Munro, E. and Odell, G. M. (in press). Ingeneue: a versatile tool for reconstituting genetic networks in silico. *J. Experimental Zoology* .

Oellers, N., Dehio, M. and Knust, E. (1994). bHLH proteins encoded by the Enhancer of split complex of Drosophila negatively interfere with transcriptional activation mediated by proneural genes. *Mol Gen Genet* **244,** 465-73.

Pai, L. M., Orsulic, S., Bejsovec, A. and Peifer, M. (1997). Negative regulation of Armadillo, a Wingless effector in Drosophila. *Development* **124,** 2255-66.

Perron, M. and Harris, W. A. (2000). Determination of vertebrate retinal progenitor cell fate by the Notch pathway and basic helix-loop-helix transcription factors. *Cell Mol Life Sci* **57,** 215-23.

Qi, H., Rand, M. D., Wu, X., Sestan, N., Wang, W., Rakic, P., Xu, T. and Artavanis-Tsakonas, S. (1999). Processing of the notch ligand delta by the metalloprotease Kuzbanian. *Science* **283,** 91-4.

Reinitz, J. and Sharp, D. H. (1995). Mechanism of eve stripe formation. *Mech Dev* **49,** 133-58.

Seugnet, L., Simpson, P. and Haenlin, M. (1997). Transcriptional regulation of Notch and Delta: requirement for neuroblast segregation in Drosophila. *Development* **124,** 2015-25.

Simpson, P. (1997). Notch signalling in development: on equivalence groups and asymmetric developmental potential. *Curr Opin Genet Dev* **7,** 537-42.

Singson, A., Leviten, M. W., Bang, A. G., Hua, X. H. and Posakony, J. W. (1994). Direct downstream targets of proneural activators in the imaginal disc include genes involved in lateral inhibitory signaling. *Genes Dev* **8,** 2058-71.

Skeath, J. B. and Carroll, S. B. (1991). Regulation of achaete-scute gene expression and sensory organ pattern formation in the Drosophila wing. *Genes Dev* **5,** 984-95.

Skeath, J. B. and Carroll, S. B. (1992). Regulation of proneural gene expression and cell fate during neuroblast segregation in the Drosophila embryo. *Development* **114,** 939-46.

Skeath, J. B., Panganiban, G. F. and Carroll, S. B. (1994). The ventral nervous system defective gene controls proneural gene expression at two distinct steps during neuroblast formation in Drosophila. *Development* **120,** 1517-24.

Struhl, G. and Adachi, A. (1998). Nuclear access and action of notch in vivo. *Cell* **93,** 649-60.

Thieffry, D., Huerta, A. M., Perez-Rueda, E. and Collado-Vides, J. (1998). From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in Escherichia coli. *Bioessays* **20,** 433-40.

Tyson, J. J., Novak, B., Odell, G. M., Chen, K. and Thron, C. D. (1996). Chemical kinetic theory: understanding cell-cycle regulation. *Trends Biochem Sci* **21,** 89-96.

Vaessin, H., Brand, M., Jan, L. Y. and Jan, Y. N. (1994). daughterless is essential for neuronal precursor differentiation but not for initiation of neuronal precursor formation in Drosophila embryo. *Development* **120,** 935-45.

Van Doren, M., Bailey, A. M., Esnayra, J., Ede, K. and Posakony, J. W. (1994). Negative regulation of proneural gene activity: hairy is a direct transcriptional repressor of achaete. *Genes Dev* **8,** 2729-42.

Van Doren, M., Ellis, H. M. and Posakony, J. W. (1991). The Drosophila extramacrochaetae protein antagonizes sequence-specific DNA binding by daughterless/achaete-scute protein complexes. *Development* **113,** 245-55.

Van Doren, M., Powell, P. A., Pasternak, D., Singson, A. and Posakony, J. W. (1992). Spatial regulation of proneural gene activity: auto- and cross- activation of achaete is antagonized by extramacrochaetae. *Genes Dev* **6,** 2592-605.

von Dassow, G., Meir, E., Munro, E. M. and Odell, G. M. (2000). The segment polarity network is a robust developmental module. *Nature* **406,** 188-92.

Weir, M. P., Edgar, B. A., Kornberg, T. and Schubiger, G. (1988). Spatial regulation of engrailed expression in the Drosophila embryo. *Genes Dev* **2,** 1194-203.

Wesley, C. S. and Saez, L. (2000). Analysis of notch lacking the carboxyl terminus identified in Drosophila embryos. *J Cell Biol* **149,** 683-96.

Wilkinson, H. A., Fitzgerald, K. and Greenwald, I. (1994). Reciprocal changes in expression of the receptor lin-12 and its ligand lag-2 prior to commitment in a C. elegans cell fate decision. *Cell* **79,** 1187-98.

Curriculum Vitae

SimBiotic Software
148 Grandview Ct
Ithaca NY 14850
Phone: (212) 658 9104
Email: meir@beakerware.org

Education

1988 - 1992 B.S., Biology. Cornell University.

Professional Positions

1998 - Owner, SimBiotic Software (formerly Beakerware), Educational software company.

1992 - 1993 Technician and Programmer, H. Steinitz Marine Biology Laboratory, Eilat, Israel, Sponsor: Dr. Amatzia Genin.

1990 - 1993 Technician and Undergraduate Research, Dept. of Ecology and Systematcis and Friday Harbor Laboratories, Advisor: Dr. Chuck Greene.

1988-1992 Programmer and Undergraduate Research, Dept. Neurobiology and Behavior, Cornell University, Advisor: Dr. Avis Cohen.

Fellowships and Honors

Howard Hughes Medical Institute Predoctoral Fellow

Honors, Undergraduate Thesis, Biology, Cornell University

Scientific Publications

In review Meir, E., S. J. Andelman. Do reserve selection heuristics work in a dynamic uncertain world.

In review Meir, E., S. J. Andelman. What data are needed to design effective reserve networks? Ecological Applications.

In press von Dassow, G. and E. Meir. Exploring modularity using dynamical models of gene networks, in Modularity in Development and Evolution, eds. G. Schlosser and G. P. Wagner, U. Chicago Press.

In press Andelman, S. J; E. Meir. Breadth is better than depth: Biodiversity data requirements for adequate reserve network design. Conservation Biology.

2002 Meir, E., G. von Dassow, E. Munro, G. M. Odell. Ingeneue: a versatile tool for reconstituting genetic networks in silico. J. Experimental Zoology.

2002 Meir, E., G. von Dassow, E. Munro, G. M. Odell. Robustness, flexibility, and the role of lateral inhibition in the neurogenic network. Current Biology 12: 778-786.

2001 Fagan, W. F., E. Meir, J. Predergast, A. Folarin, and P. Karieva. Characterizing population vulnerability for 758 species. Ecology Letters, 4: 132-138.

2001 Fagan, W.F., E. Meir, S. Carroll, and J. Wu. The ecology of urban landscapes: modeling housing starts as a density-dependent colonization process. Landscape Ecology, 16: 33-39.

2000 von Dassow, G, E. Meir, E Munro, G. M. Odell. 2000. The segment polarity network is a robust developmental module. Nature 406: 188-92.

2000 Meir, E. and W. F. Fagan. Will observation error and biases ruin the use of simple extinction models? Conservation Biology, 14: 148-154.

1999 Fagan, W. F., E. Meir and J. L. Moore. Variation thresholds for extinction and their implications for conservation strategies. The American Naturalist, 154: 510-520.

1998 Meir, E. Silicon teachers in undergraduate biology. Integrative Biology, 1: 99-107.

1998 Meir, E. Review of "An introduction to the mathematics of biology: With computer algebra models." Quarterly Review of Biology, 73: 257.

1998 Marvier, M. A., E. Meir, and P. Karieva. How do the design of monitoring and control strategies affect the chance of detecting and containing transgenic weeds?. In <a report to european community>.

1998 Meir, E. and P. Kareiva. Contributions of spatially explicit landscape models to conservation biology. In P. Fiedler and P. Kareiva (eds), Conservation Biology for the Next Decade, Chapman and Hall, NY.

1997 Skelly, D. K. and E. Meir. Rule-based models for evaluating mechanisms of distributional change. Conservation Biology, 11:531-538.

1996 J. E. Zamon, C. H. Greene, E. Meir, D. A. Demer, R. P. Hewitt, S. Sexton. Acoustic characterization of the three-dimensional prey field of foraging chinstrap penguins. Marine Ecology Progress Series, 131:1-10.

1995 Meir, E. Exploring ecological models with EcoBeaker. BioQuest Notes, 5: 13-15.

1994 Genin, A., C. Greene, L. Haury, P. Wiebe, G. Gal, S. Kaartvedt, E. Meir, C. Fey, J. Dawson. Zooplankton patch dynamics: daily gap formation over abrupt topography. Deep Sea Research I, 41:941-951.

1992 Meir, E. Three dimensional effects of langmuir circulation on the patchiness of pelagic animals in the ocean. Undergraduate Thesis, Ecology and Systematics, Cornell University.

1991 Meir, E and B. Haller. Climbing Fractal Mountains. MacTutor, May issue.

Software for Biologists

In beta Munro, E. M., W. Sunderland, A. Davis, S. Votaw, W. Moody, E. Meir. NerveWorks. SimBiotic Software, Ithaca, NY

2002 Meir, E., E. Steinberg, et al. EcoBeaker101. SimBiotic Software, Ithaca NY.

2002 Maruca, S., D. Stahl, J. Kingsolver, J. Herron, E. Meir. EvoBeaker Prototype. SimBiotic Software, Ithaca, NY

2001 Meir, E. et al. EcoBeakerHS. SimBiotic Software, Ithaca NY.

2000 Meir, E., G. von Dassow, E. Munro, G. M. Odell. Ingeneue. www.ingeneue.org.

1999 Meir, E. EcoBeaker (v2.0). BeakerWare, Ithaca NY.

1996 Meir, E. EcoBeaker (v1.0). Sinauer Associates, Sunderland, MA.

1995-2000 Meir, E. Biology Education Software FAQ. Published over internet at http://www.beakerware.org/biosoft/

1995 Meir, E. Matlab spatial statistics routines for biological acousticians. Freeware over internet.

1990 - 1995 Meir, E. Spike Studio (v 3.6). Data acquisition and analysis software for neurophysiologists. Freeware over internet (was used in dozens of laboratories).

1991 - 1994 Meir, E. I Ching Squared (v 2.1). Automated statistics for neurophysiologists. Freeware over internet.

1993 Meir, E. Phase D (v 1.0). Phase-shift diagram plotter. Freeware over internet.