Co-option and Irreducibility in Regulatory Networks for Cellular Pattern Development

Ranjitha A. Dhanasekaran
Department of Computer Science
Utah State University, Logan, USA
Phone: (435) 797-2432, email: ranjitha@cc.usu.edu

Gregory J. Podgorski,
Department of Biology and Center for Integrated BioSystems
Utah State University, Logan, USA
Phone: (435) 797-3712, email: podgorski@biology.usu.edu

Nicholas S. Flann
Department of Computer Science
Utah State University, Logan, USA
Phone: (435) 797-2432, email: nick.flann@usu.edu

Abstract* – We used a computational approach to examine three questions at the intersection of developmental biology and evolution: 1) What is the space available for evolutionary exploration for genetic regulatory networks (GRNs) able to solve developmental patterning problems? 2) If different GRNs exist that can solve a particular pattern, are there differences between them that might lead to the selection of one over another? 3) What are the possibilities for co-opting GRN subcircuits or even entire GRNs evolved to solve one pattern for use in the solution of another pattern? We used a Monte Carlo strategy to search for simulated GRNs composed of nodes (proteins) and edges (regulatory interactions between proteins) capable of solving one of three striped cellular patterning problems. These GRNs were subjected to a knockout procedure akin to gene knock-outs in genetic research. Knockout was continued until all individual network components of the reduced GRN were shown to be essential for function. This GRN was termed irreducible. We found many different unique irreducible GRNs that were able to solve each patterning problem. Since any functional GRN must include an irreducible GRN as a core or subgraph, the space for evolutionary exploration of pattern-forming GRNs is large. Irreducible GRNs that solve a particular pattern differed widely in their robustness—the ability to solve a target pattern under different initial conditions. These differences may offer a target for selection to winnow out less robust GRNs from the set of GRNs found in nature. Finally, subgraph isomorphism analysis revealed great potential for co-option during evolution. Some irreducible GRNs appear in their entirety within larger GRNs that solve different patterning problems. At much higher frequency, subcycles are shared widely among irreducible GRNs, including those that solve different patterns. Irreducible GRNs may form the core elements of GRNs found in biological systems and provide insight into their evolution.

Keywords* – Genetic regulatory network, GRN, subgraph isomorphism, co-option, evolutionary dynamics, modularity, development, pattern formation, subcircuit, self-organization.

I. INTRODUCTION

Genetic regulatory networks (GRNs) that control development are being deciphered through experimental approaches propelled by advances in genomics and systems biology [1]. These GRNs for development are complex and robust, generating reproducible outputs over a broad range of initial conditions [2]. Common themes in network architecture and regulatory logic are beginning to emerge (see [3], [4], [5], [1]), including the use of evolutionarily conserved regulatory modules (kernels) and the existence of smaller regulatory circuits that have been repeatedly co-opted for different ends in development [1].

Extant GRNs in nature may not be the only ones capable of controlling a particular aspect of development. Instead, GRNs in nature may represent the one solution that was stumbled upon in the evolutionary history of lineage. Once the solution was "discovered," it may have been frozen in place. The high degree of conservation of many GRN architectural elements in animal development suggests this possibility—but only if there are other solutions open for random evolutionary processes to discover.

We are interested in learning the dimensions of the developmental GRN solution space for evolutionary exploration. We approached this question by searching for GRNs that would solve three different patterns that resemble the striped patterns of pair-rule gene expression seen in the development of *Drosophila* and other insects (see Fig. 1). Building on our earlier work [6], we used a Monte Carlo strategy to identify networks that solve the three striped patterns. A genetic knockout algorithm, similar in logic to the use of targeted gene knockouts to understand biological GRNs, was employed to remove network components one-by-one until an irreducible network capable of solving the pattern was discovered.

Our results suggest that the solution space for evolutionary exploration is large. We find that each of the patterns can be solved by many different irreducible GRNs. We also determined that there are significant differences in the robustness of the networks. These robustness differences may generate a strong selective pressure capable of winnowing out less robust GRNs. Finally, we show that there are many possibilities for evolutionary co-option of GRN subcircuits or even entire GRNs to solve new patterning problems.

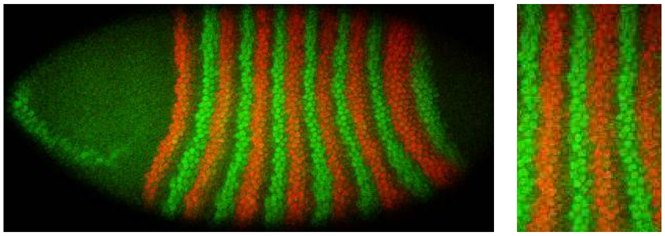


Fig. 1. Striped gene expression patterns in early *Drosophila* development. Odd-skipped (green) and even-skipped (red) expression domains at cleavage cycle 14. The left panel shows the nearly complete embryo; the right panel is a closer view where each point is a cell nucleus. Image courtesy of Dr. J. Reinitz. Mt. Sinai School of Medicine.

II. APPROACH

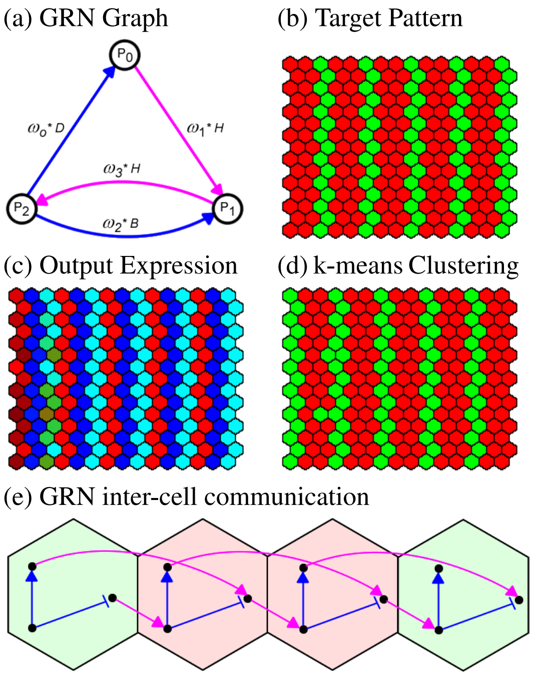
Our approach was to model GRNs composed of nodes (proteins) connected by a variety of edge types (regulatory interactions between proteins) that simulate many of the regulatory interactions that occur within and between embryonic cells. In contrast to more explicit GRN models that consider the interaction of transcription factors with cis-acting control sequences of regulatory genes ([7], [8]), we chose a more abstract level of modeling that only considered interactions between the regulatory gene products ([9],[3]).

A. GRN Space Representation

In this framework, GRNs can be described as a graph, where each node represents a protein's expression level and each edge represents a regulatory interaction between proteins. A protein is influenced when its production or degradation is controlled as a function of another protein's expression levels. Since production and degradation are defined as rates of change, the GRN is naturally modeled as a set of coupled differential equations ([8], [9], [10], [3]). Fig. 2(a) shows an example of a 3 protein, 4 edge GRN represented as a graph and Fig. 2(f) shows the same GRN as a set of coupled differential equations.

Table I illustrates the edges that represent protein interactions considered in this study. An edge j from node P_1 to node P_0 contributes to the rate of change of P_0 as a weighted expression of P_1. Each edge j has a weight ω_j that is the strength of the influence of j ($0.0 \leq \omega_j \leq 1.0$) on the change in P_0. Each cell σ maintains its own set of protein expression values $P_i(\sigma)$. Edges implement intracellular interactions when P_0 and P_1 are in the same cell and intercellular interactions when P_1 is from other cells.

In our model, pattern formation is over a 2D sheet of regular hexagonal cells (see Fig. 2), each operating the same GRN. Within an individual cell, five simple intracellular protein interactions are modeled (see Table I): ***A*** and ***B*** implement direct up and down regulation [10], ***C*** and ***D*** control to the same or opposite of an expression value [11], or ***E*** which implements a simple quadratic regulation used in reaction-diffusion models [12] and [13]. Limiting functions $f(x) = \frac{x^2}{(1+x^2)}$ and $g(x) = 1 - f(x)$ are employed in ***C*** and ***D*** to model saturation effects in protein production and degradation [11].



(f) Coupled differential equations

$$\begin{aligned}
\frac{dP_0(\sigma)}{dt} &= \omega_0(g(P_2(\sigma)) - P_0(\sigma)) \\
\frac{dP_1(\sigma)}{dt} &= \omega_1(f(P_0(n_W(\sigma))) - P_1(\sigma)) - \omega_2 P_2(\sigma) \\
\frac{dP_2(\sigma)}{dt} &= \omega_3(f(P_1(n_W(\sigma)) - P_2(\sigma))
\end{aligned}$$

Fig. 2. Example of an irreducible GRN (a) found by our system for solving the *2-skip-1* target pattern (b). The letters on the edges of (a) indicate edge types shown in Table I. The output protein expression levels are shown in (c). Panel (d) shows the result of k-means clustering of the output expressions levels from (c). The GRN mapped onto a strip of cells is show in (e) and its coupled differential equations are shown (f).

Over the sheet of cells, intercellular proteins influence each other through both long-range and short-range signaling. Edge ***F*** implements long-range signaling through diffusion [8]. Edges ***G***-***J*** implement short-range signaling, where a cell can sense protein expression levels in directly neighboring cells across contacting membranes as in [11], [2] and [3]. Edges ***G***-***J*** enable a cell to signal to a specific geometric neighbor cell using n_S, n_W, n_N, n_E, which return the cell directly neighboring to the south, west, north and east respectively. Such directional signalling is used in the embryo to build internal segment borders [2] and relies on sensing anterior-posterior, dorsal-ventral and right-left polarity.

In [3] the definition of the cardinal directions was unambiguous because a sheet of square cells was modeled. In this work with hexagonal cells, each cell has two neighbors in to the north and two to the south. To avoid ambiguity, north-south signalling occurs only with the single cell to the left. This formalism enables the 90 degree rotation of both patterns and GRNs to be well defined.

A GRN may have p proteins and e edges, where $p \leq e \leq$

TABLE I

THE POSSIBLE EDGES FOR A GRN

Label	Description	Definition
A	P_0 direct expression by P_1	$\frac{dP_0(\sigma)}{dt} = \omega_j P_1(\sigma)$
B	P_0 direct degradation by P_1	$\frac{dP_0(\sigma)}{dt} = -\omega_j P_1(\sigma)$
C	P_0 driven to same as P_1	$\frac{dP_0(\sigma)}{dt} = \omega_j (f(P_1(\sigma)) - P_0(\sigma))$
D	P_0 driven to opposite of P_1	$\frac{dP_0(\sigma)}{dt} = \omega_j (g(P_1(\sigma)) - P_0(\sigma))$
E	P_0 quadratic regulation by P_1	$\frac{dP_0(\sigma)}{dt} = \omega_j (P_1(\sigma)^2 - \psi_j)$
F	P_0 Diffusion with zero boundary conditions	$\frac{dP_0(\sigma)}{dt} = \omega_j \frac{\partial^2 P_1(\sigma)}{\partial x^2} = \nabla P_1(\sigma)$
G:J	P_0 driven to same as geometric neighbor value of P_1; with $i \in N, W, S, E$	$\frac{dP_0(\sigma)}{dt} = \omega_j (f(P_1(n_i(\sigma))) - P_0(\sigma))$

$d(p^2 - p)$, where d is the number of edge types ($d = 10$). In this work, GRN graphs are mostly sparse, with $e \ll d(p^2 - p)$, in contrast to the fully connected networks of [10] and [3] where $e = d(p^2 - p)$. There are $d^e p^e p^{2p(e-p)}$ possible GRNs, where d is the number of edge types. In this study $d = 10$ (see Table I). Fig. 2(a) shows an example GRN with 3 proteins and 4 edges discovered by the knockout search method described in the following section. To determine the coupled differential equations of a GRN, the equations of each edge are composed, as illustrated in Fig. 2(f). To solve a GRN implemented in a sheet of q cells, each cell's protein values are first set from a uniform random distribution $[0.5, 1.0]$, then the $p \times q$ differential equations are numerically solved using the Runge-Kutta method with $dt = 0.05$ until a fixed point is achieved (where the average update error $\leq 10^{-8}$ per cell). Fig. 2(c) illustrates the output expression pattern formed when the differential equations in Fig. 2(f) are solved over a sheet of 15×15 cells. In this panel, the color of each cell is determined by mapping P_0 to the red level, P_1 to green and P_2 to blue.

B. Pattern Formation and GRN Evaluation

In this work we study the space of GRNs that can solve a small set of striped patterning problems similar to the patterns observed in early *Drosophila* development. Fig. 1 shows an example from *Drosophila* development and Fig. 2(b) shows one of our model target patterns, referred to as a *2-skip-1*. Other patterns we studied are *1-skip-1* and *2-skip-2*.

To quantify how well a GRN solves a particular target pattern, the protein expression levels that result from solving the coupled differential equations must be mapped to distinct cell types using a combinatorial code. Then the assigned cell types are matched against the cell types given in the target pattern, and a count made of the mismatched cells.

Combinatorial codes partition the p-dimensional protein spectrum into distinct regions, each corresponding to an assigned cell type [14]. Combinatorial codes enable each cell's fate to be established autonomously and are defined as combinations of easily distinguishable thresholds of protein values. A similar logic is used in the readout of the Hox gene combinatorial code to establish segment identity [15].

In this work, k-means clustering is used to discover a combinatorial code that best partitions the cells into k-types, where k is the number of types in the target pattern. Each partition in k-means clustering is represented as a centroid $\bar{P}_j, 1 \leq j \leq k$ such that the intra-cluster distance is minimized and the inter-cluster distance maximized. A combinatorial code is extracted from a set of centers by greedily picking proteins whose values best discriminate the centers by using an information gain measure. Only proteins whose values are well separated are picked. If such a set of proteins cannot be identified, a combinatorial code cannot be found and the pattern evaluator fails. A successful example is given in Fig. 2(d), with $k = 2$ since there are two types in the target pattern. Two clusters are easily discriminated by defining the combinatorial code: IF $(P_0(\sigma) > 0.5$ and $P_1(\sigma) < 0.5)$ THEN σ is type one (red cells in Fig. 2(d)) ELSE σ is type two (green cells).

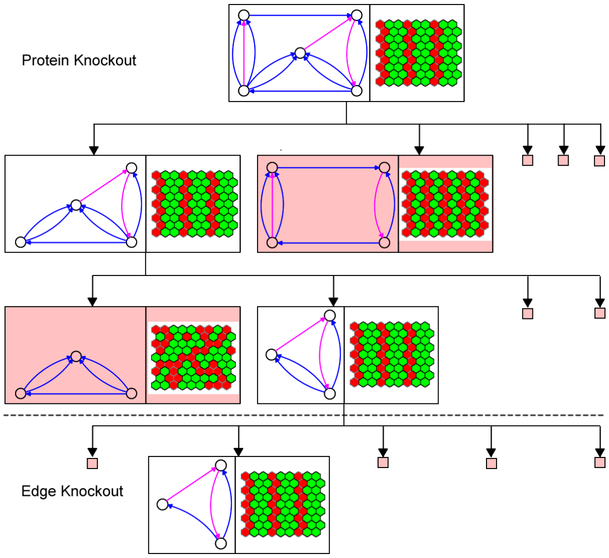


Fig. 3. An example tree created by running the knockout method show in Table II on a *2-Skip-1* GRN discovered by Monte Carlo Search. Each GRN explored is shown with an example expression pattern. Nodes with light red backgrounds represent failure, where the pattern cell mismatch percent exceeds the 10% threshold.

Each cell's assigned type is compared with the type given in the target pattern and a tally is made of misplaced cells. If the target pattern has alternative rotations (or phases), all rotations are tried and the best match is used as the mismatch percent. Comparing Fig. 2(b) with Fig. 2(d) gives a mismatch of 2 cells in 225 or 0.89%. To give an accurate evaluation, the GRN is solved five times under different random initial protein conditions and the mean mismatch percent returned. Five random repeats were used in this study to minimize computation time while serving as a reasonable estimate of the true mismatch value. In this study a threshold value of 10% misplaced cells is considered an acceptable solution since the output patterns scoring below this threshold are clearly recognizable target patterns.

C. Irreducible GRNs

An irreducible GRN is defined as a GRN where the removal of any one component (protein or edge) results in loss of function. In this study, loss of function is the inability of the GRN

TABLE II

ALGORITHM DEFINING THE KNOCKOUT PROCEDURE

```
**Algorithm** Knockout($P$,$E$,$T$)
**Input:**$P = \{p_1, p_2, \ldots, p_k\}$, GRN proteins
        $E = \{e_1, e_2, \ldots, e_q\}$, GRN edges
        $T$ = Target cell pattern
**Output:** $(\widetilde{P}, \widetilde{E}), \widetilde{P} \subseteq P, \widetilde{E} \subseteq E$ (a Knocked-Out GRN)
**Begin**
    //Protein Knockout
    **for** each $p_i \in P$
        $P' \leftarrow P - \{p_i\}$
        $\widehat{E} \leftarrow \{e_j \in E | e_j$ connects to $p_i\}$
        $E' \leftarrow E - \widehat{E}$
        **if** Mismatch($T$,Solve($P'$,$E'$)) $\leq 10\%$
        **then return** Knockout($P'$,$E'$,$T$)
    //Edge Knockout
    **for** each $e_i \in E$
        $E' \leftarrow E - \{e_i\}$
        $\widehat{P} \leftarrow \{p_j \in P | p_j$ is isolated w.r.t $E'$ $\}$
        $P' \leftarrow P - \widehat{P}$
        **if** Mismatch($T$,Solve($P'$,$E'$)) $\leq 10\%$
        **then return** Knockout($P'$,$E'$,$T$)
    **return** $(P, E)$
**End**.
```

to produce a target pattern meeting the 10% mismatch threshold. To identify irreducible GRNs a knockout procedure, similar to those used in a genetic approach to analyze biological GRNs, was employed to reduce the size of GRNs found by a Monte Carlo search of the GRN space (see [6] for details). This knockout procedure starts with a GRN that accurately solves one of the target patterns and identifies a smaller subgraph of the original GRN that still adequately solves that same pattern. Each protein and edge successfully deleted is termed an extraneous component.

Consider the knockout tree in Fig. 3. The parent GRN has 5 proteins and 12 edges representing the proteins and the interactions among them, respectively. The knockout procedure is a two step process. First, the proteins are removed (knocked-out) one-by-one, then the edges are knocked-out one-by-one. At every node in the knockout tree, every component of the parent GRN is knocked-out one-by-one, and each resulting child GRN is measured for its fitness with respect to the target pattern. If a child GRN's mismatch percent is less than or equal to the threshold value, then it becomes the parent GRN and is submitted to another round of the knockout procedure. Only the first successful child GRN is pursued for further knockout. Knockouts are repeated until none of the child GRNs meets the threshold criteria. In this case the parent GRN is considered to be the irreducible GRN. In this example shown in Fig. 3, 2 proteins and 8 edges were found to be extraneous, resulting in an irreducible GRN with only 3 proteins and 4 edges. During the knockout of this GRN, which produces a *2-skip-1* pattern, we noted that one of the failure GRNs accurately produced a *1-skip-1* pattern. This indicates a close relationship among GRNs for similar stripped patterns.

The knockout algorithm is formally defined as a greedy depth-first search and is given in Table II. Two sub-routines are called: Solve(P, E) which initializes the proteins and then uses the Runge-Kutta method until a fixed point is reached, and Mismatch(T, S) which matches the target pattern T against the pattern created by the GRN and its combinatorial code. The algorithm terminates when the attempt to delete each protein and each edge fails to produce the solution pattern. The input to the knockout algorithm is a GRN discovered by a Monte Carlo search that solves a striped pattern. The output is defined as an irreducible GRN that is both minimal in size and adequately solves the target pattern T.

Since the knockout algorithm is greedy and only considers single component deletions, it may not find the globally minimum GRN, but may discover a locally minima. Such a local minima GRN may include a subgraph that if deleted as a unit, would result in a still functioning GRN. A subgraph in a local minima GRN may for example include cycles of up and down regulation (edge types ***A*** and ***B***) or north and south signalling (edge types ***G*** and ***I***), that will disrupt the pattern formation process if only one component is deleted, but maintain the pattern when both are deleted together. To identify globally irreducible networks, an exhaustive search deleting all possible subsets of components would have to be performed. Such an optimal search was not done for this study due to computational limitations and the demonstrated effectiveness of the single-step knockout procedure in identifying small GRNs.

D. Co-option among Irreducible GRNs

Co-option is the use of an element of one GRN by another GRN, often for a distinct function. This work considers two kinds of co-optable network elements: modules and subcircuits. We define a module as a fully functional GRN that appears as part of another larger GRN. A subcircuit is defined as a GRN cycle that occurs in two or more GRNs. In contrast to modules, sub-circuits may or may not function as GRNs that produce a recognizable pattern.

D.1 Modules: Subgraph Isomorphic GRNs

Modules are discovered by computing subgraph isomorphisms between all pairs of GRN graphs. A graph G_1 is a subgraph of G_2 if under some one-to-one mapping between the nodes of G_1 and a subset of the nodes of G_2, the in and out edges of all nodes in G_1 are always a subset of those of G_2. This problem is known to be P-space Complete in general. In this study, we are interested in determining subgraph isomorphism between directed graphs with distinct edge labels, which is known to be NP-Complete. The relatively small size of the irreducible GRN graphs (see Fig. 6 for some examples) makes it feasible to use an exact algorithm whose performance was

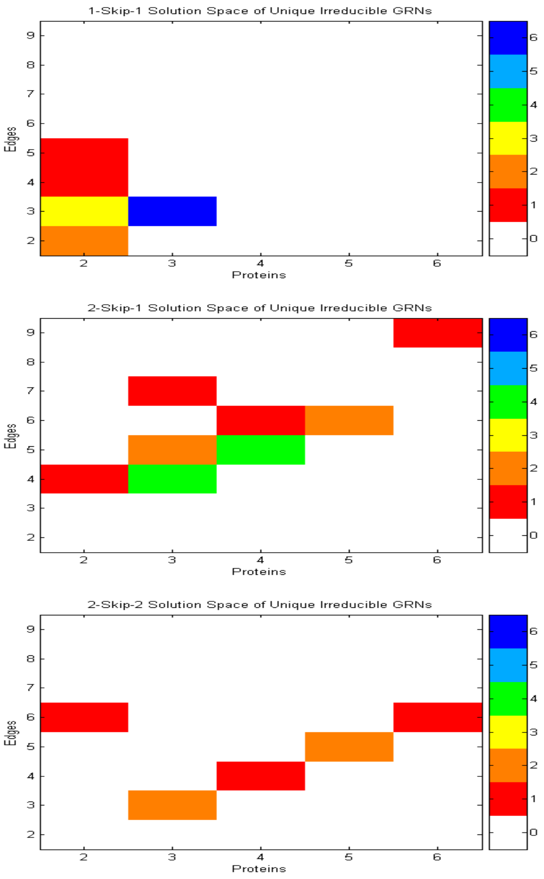


Fig. 4. The distribution of unique irreducible GRN's for the three target patterns. The scale maps each color to the count of GRNs.

greatly improved through the use of edge-hashing [16]. All irreducible GRN graphs were compared with all other GRNs for all three patterns.

D.2 Subcircuits: Common Sub-cycles between GRNs

Subcircuits are computed by extracting all cycles from the GRN graphs and performing pairwise comparisons over all cycles. A cycle in a GRN graph of p proteins and e edges is a feedback loop comprising of between 2 and e edges and is characterized by the order of the edge types involved. To determine common cycles between graph G_1 and G_2, we first extract all cycles from each graph, then identify an intersection under all possible rotations of each cycle. The complexity of this computation is $O(p^2e^3)$, which is feasible for the small GRN graphs studied here.

III. RESULTS

This study first discovered a distinct set of irreducible GRNs for each of the target patterns and considered their distribution with respect to size (number of proteins and edges). Next, the robustness of the irreducible GRNs was measured and analyzed with respect to network architecture. Finally, the common subgraphs and sub-cycles were identified between the irreducible GRNs.

TABLE III

MONTE CARLO RESULTS

Pattern	Number of GRNs generated in the Monte Carlo Search	Number of GRNs meeting the threshold	Chance of finding a GRN meeting threshold
1-skip-1	3,387,073	440	1/7,698
2-skip-2	4,422,289	205	1/21,572
2-skip-1	3,773,131	113	1/33,390

A. Identification of Irreducible GRNs

An extensive Monte Carlo search of the GRN space was performed using 30 high performance workstations running in parallel for 10 days. Each random GRN was generated by first uniformly sampling the number of proteins $2 \leq p \leq 8$ and edges $p \leq e \leq 13$, creating a random connected graph, assigning random edge types (from Table I) then setting the initial edge strengths randomly from the range $0.8 \leq \omega_j \leq 1.0$. This search identified 758 GRNs with mismatch percent below the 10% threshold over the target patterns (see Table III). A random subset of GRNs was selected for each target pattern to provide an approximately uniform distribution of GRNs over the space of network sizes $2 \leq p \leq 8$ and $p \leq e \leq 13$. Each GRN solution was then processed by the knockout procedure shown in Table II to identify its irreducible GRN. Finally, isomorphic GRNs were eliminated from the set of irreducible GRNs using an edge hashing scheme [16]. This produced a total of 13 *1-skip-1* GRNs, 16 *2-skip-1* GRNs and 7 *2-skip-2* GRNs. The distribution of irreducible GRNs is shown in Fig. 4.

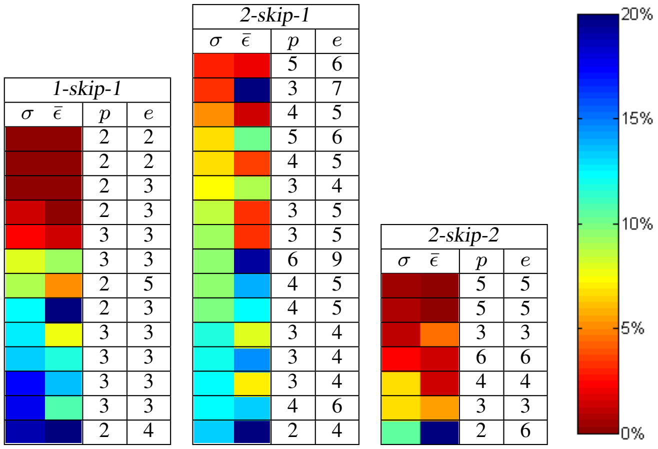
B. Robustness of Irreducible GRNs

Robustness measures the ability of a GRN to consistently produce a high quality pattern under varying initial concentrations of proteins and varying strengths of the interactions between them (ω_j in Table I). All the irreducible GRNs demonstrate robustness under varying initial concentrations.

To measure the robustness of the GRNs we employed the same methodology used by von Dassow et. al [2] where they evaluated the robustness of a single network that creates segment polarity in *Drosophila*. The robustness of each irreducible GRN was determined by setting each edge strength ω_j, $(1 \leq j \leq e)$ in turn to a uniform random number in the range $0.1 \leq \omega_j \leq 1.0$ and then computing the mismatch percent of the resulting pattern with respect to its target pattern. The process was repeated 40 times for each GRN, then the mean and standard deviation was computed. The results are illustrated in Fig. 5. The standard deviation of the mismatch percent quantifies the robustness of each individual GRN with low standard deviation implying high robustness. Examples of GRNs with high, median and low robustness for each target pattern are shown in Fig. 6.

B.1 Modules: Subgraph Isomorphic GRNs

The results showing the sub-graph isomorphic relation between each pair of GRNs are given in Fig. 7. Each circle in



1-skip-1 p	e
2	2
2	2
2	3
2	3
3	3
3	3
2	5
2	3
3	3
3	3
3	3
3	3
2	4

2-skip-1 p	e
5	6
3	7
4	5
5	6
4	5
3	4
3	5
3	5
6	9
4	5
4	5
3	4
3	4
3	4
4	6
2	4

2-skip-2 p	e
5	5
5	5
3	3
6	6
4	4
3	3
2	6

Fig. 5. The robustness of all identified irreducible GRN solutions for the 3 striped patterns. In each table, σ is the standard deviation of mismatch percent, $\bar{\epsilon}$ is the average mismatch percent, p is the number of proteins and e number of edges. The rows are sorted by σ so the most robust network is the at the top and the least robust network at the bottom. The key for the colors used to represent σ and $\bar{\epsilon}$ is on the right.

the figure represents a unique GRN, with the clockwise order of each pattern corresponding to the order of robustness shown in Fig. 5. Our results show that modules among irreducible GRNs are rare. There are 630 possible uses of the GRNs as modules (when the graph in Fig. 7 is fully connected), but only 9 (1.4%) are employed. Interestingly, only one module was shared among GRNs that solve the same target pattern. In pattern *1-skip-1*, a $p = 2, e = 3$ solution GRNs was included as a subgraph of a $p = 2, e = 5$ solution GRN. In this case, the two extraneous components in the larger GRN form a north-south regulatory cycle that disrupts the pattern when only one edge is eliminated. All the other 35 irreducible GRNs appear to be globally minimal. This result supports the effectiveness of the greedy knockout procedure for the discovery of irreducible GRNs.

There is a partial ordering relationship among the GRNs solving the three striped patterns. The *1-skip-1* GRNs can be used as modules within GRNs that solve the *2-skip-1* and *2-skip-2* patterns. The *2-skip-2* GRNs also appear as modules within the *2-skip-1* GRNs. We never detected a case in which a *2-skip-1* GRN was used as a module within any other pattern's GRN. This suggests that the *2-skip-1* pattern is more difficult to solve, perhaps because of its asymmetry.

B.2 Subcircuits: Common Sub-cycles between GRNs

The results showing the common sub-cycle relation between each pair of GRNs is shown in Fig. 8. A sub-cycle is feedback loop which occurs in two or more GRNs. A sub-cycle, which functions as a subcircuit in biological GRNs, may or may not operate as an independent pattern-solving GRN. There are many more subcircuits among the GRNs than modules. We identified 56 (9%) out of a potential 630 pairs of GRNs that could share a subcircuit. An analysis of these 56 common subcircuits using edge hashing yields ten unique subcircuits, six 2-edge cycles and four 3-edge cycles. In contrast to modules, we found subcircuits that were shared between GRNs that solve the same pattern. The *2-skip-1* GRNs had 28 shared subcircuits, the *1-skip-1* GRNs had 14, while the *2-skip-2* GRNs shared none.

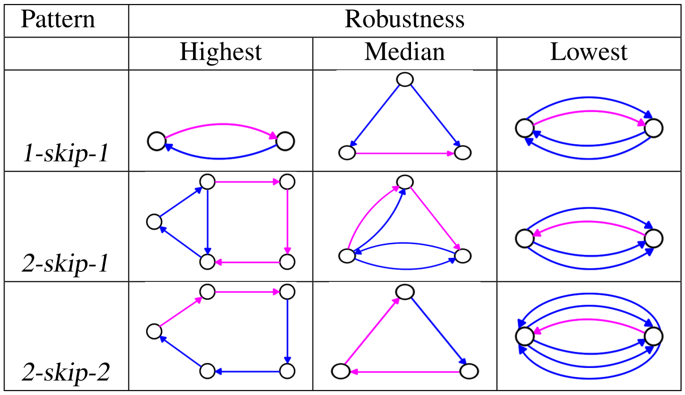


Fig. 6. Examples of irreducible GRNs discovered for the three target patterns. Edges are color coded based on whether they implement intracellular control (blue) or intercellular signalling (purple).

IV. DISCUSSION

Three questions drove this work: What is the space available for evolutionary exploration for GRNs able to solve developmental patterning problems? If different GRNs exist that can solve a particular pattern, are there differences between them that might lead to the selection of one over another? What are the possibilities for co-opting GRN subcircuits or even entire GRNs evolved to solve one pattern for use in the solution of another pattern?

We searched for all GRNs within the range of 2 protein-2 edge to 8 protein-13 edge GRNs that could solve one of three cellular patterning problems. GRNs of a given complexity (number of proteins and number of edges) were subjected to a mutational knockout analysis. Proteins and the connections between them (edges) were removed one-by-one to learn if they were extraneous or essential for GRN function. Knockout of extraneous components was continued until a GRN was discovered in which every individual component was essential for network function. Even though many of these GRNs maybe local minima they were considered to be irreducible in this study. These simple, functional GRNs are potentially available for evolutionary discovery.

A. The GRN Space Available for Evolutionary Exploration

This work has identified a set of minimal and irreducible GRNs that can solve a set of striped cell patterning problems using a Monte Carlo exploration followed by a knockout procedure. Knowing the number and structure of irreducible GRNs helps define the potential space available for evolutionary exploration since any GRN that can solve a pattern T must

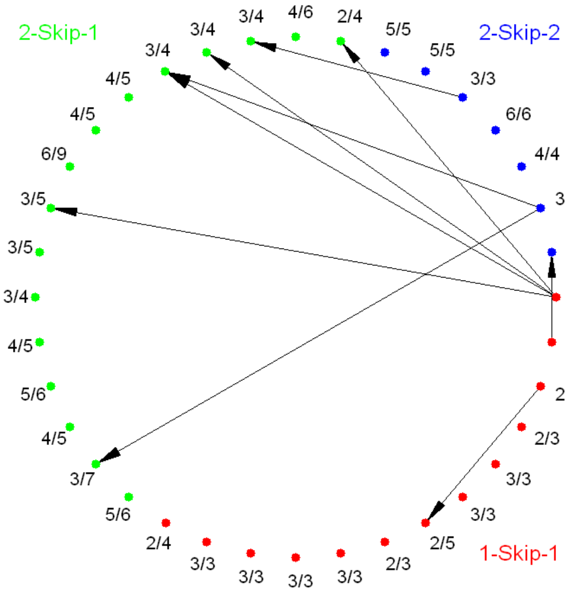


Fig. 7. Subgraph isomorphism (module) results. Each node in the circle represents a single irreducible GRN. Red is *1-skip-1*, green is *2-skip-1* and blue is *2-skip-2*. The node labels correspond to those in Fig. 5. An edge from GRN i to j means that i is a sub-graph of j.

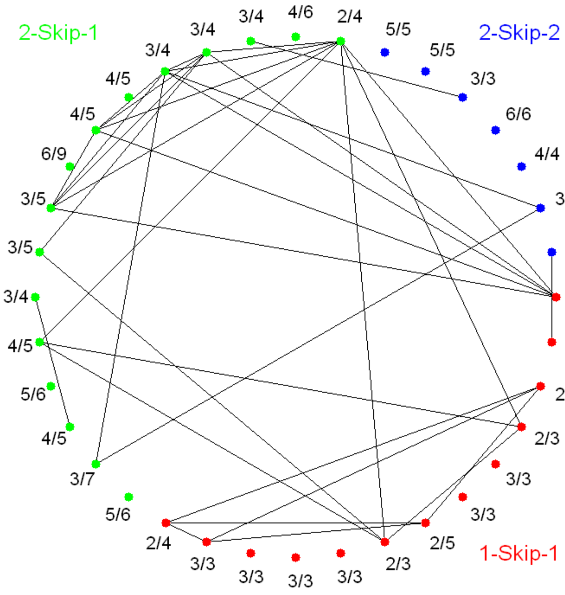


Fig. 8. Graph sub-cycle (subcircuit) results. Each node in the circle represents a single irreducible GRN. Red is *1-skip-1*, green is *2-skip-1* and blue is *2-skip-2*. The node labels correspond to those in Fig. 5. Two GRNs are connected if they share a common cycle.

include as a subgraph or core element one of the available irreducible GRNs for T, with additional components being extraneous.

There is evidence [10], [15] for the evolution of biological GRNs that exceed the minimum required complexity (i.e., GRNs capable of reduction in this work). One explanation is that extraneous components provide robustness when GRNs are subjected to continuous mutation [10]. Identification of a set of irreducible GRNs and the knockout procedure enables the quantification of such extraneous components and the testing of this explanation. Such complex networks may have evolved through the initial discovery of simple ancestral GRNs followed by their embellishment with extraneous components.

The GRN space was dense for all three patterns we investigated, with chances of randomly finding a solution ranging from 1 in 7,698 for *1-skip-1* to 1 in 33,390 for *2-skip-1*. Among these 758 discovered GRNs, there were 16 unique, irreducible GRNs that solved the *1-skip-1* pattern, 13 for the *2-skip-1* pattern and 7 for the *2-skip-2* pattern (Fig. 5).

Fig. 4 shows the set of minimal complexity solution GRNs as a density plot for each target pattern: the count of irreducible GRNs as a function of the number of proteins and edges. Some observations included: 1) *1-skip-1* is the simplest problem with the lowest required network complexity; 2) There was a significantly broader distribution of network complexity for solutions of the *2-skip-1* and *2-skip-2* patterns; and 3) For symmetrical patterns *1-skip-1* and *2-skip-2* most solutions occurred when $p = e$ and the network is a single cycle. In contrast, the asymmetric pattern *2-skip-1* required solutions where $e = p + \Delta e, \Delta e \geq 1$ with the extra edges forming new cycles (see *2-skip-1* row in Fig. 6).

B. Selection for Robustness

Given that there are many possible GRNs open for discovery, are there differences between these core elements that would favor the ultimate selection of one over another? One feature of GRN operation that may provide a selective advantage is robustness [17], [18]. We examined the robustness of each irreducible GRN over randomly generated and widely varying protein and edge-strength values. Large differences were seen in the robustness of GRN pattern formation as a function of network complexity (Fig. 5). For irreducible GRNs that solve *2-skip-1* and *2-skip-2* patterns, there is a trend for networks of intermediate complexity to be more robust than those at either extreme of complexity. In addition, it appears that GRN graphs with low in- and out-degree and sparse cycles had higher robustness, which can be seen in Fig. 6. Note that more densely connected networks (where $e \approx d(p^2 - p)$) appear to have poor robustness (consider the three least robust networks in Fig. 6), perhaps because the two proteins are influenced by multiple overlapping cycles. A significant conclusion drawn from this part of the investigation is that if robustness is a target of selection, then of the many networks initially discovered by evolution, only those with high-robustness core GRNs are likely to remain after long-term selection.

C. There Are Many Opportunities for Co-Option

An important aspect of GRN architecture and its evolution is suggested from analysis of GRNs that control animal development. This feature is the existence of subcircuits and modules that appear frequently in GRNs employed in the development of divergent lineages, such as *Drosophila*, sea urchin, and the chordate *Ciona* [15]. The appearance of these simpler sub-elements opens the possibility for evolution by co-option of existing network modules and subcircuits. By combining previously evolved network features, new patterning problems can be solved. Co-option of existing subcircuits and modules also may provide a mechanism to escape traps of local fitness maxima in the GRN fitness landscape.

We examined whether and to what extent the irreducible GRNs discovered here shared architectural features. Such shared features, especially if they appear frequently, may illuminate potential evolutionary trajectories through co-option in the evolution of pattern forming GRNs. We discovered extensive overlap in many architectural features in GRNs that solved the three different target patterns.

Network elements have been referred to by a variety of terms that include kernels, subcircuits, switches, and modules [1]. We searched for sharing of two types of network elements, referred here as modules and subcircuits. In this work modules were defined as a subgraph within a larger GRN that is capable of operating on its own to solve a pattern problem. A subcircuit is a conserved (duplicated in at least two irreducible GRNs) network feedback loop that may or may not operate as an independent GRN.

Four different modules exist within irreducible GRNs that solve the *2-skip-1* and *2-skip-2* patterns (Fig. 7). These modules are independent GRNs that solve the *1-skip-1* or *2-skip-2* patterns. In contrast, sharing of subcircuits is more extensive and complex (Fig. 8). Subcircuits are shared both within GRNs that solve a single patterning problem (e.g., see the extensive subcircuit sharing in many *2-skip-1* GRNs) and between GRNs that solve two different patterning problems (e.g., the extensive subcircuit sharing between the *1-skip-1* and *2-skip-1* GRNs.). In addition, some subcircuits that appear in the *1-skip-1* GRNs are used to solve more complex patterns. For example, a subcircuit within the 2-protein, 2 -edge *1-skip-1* GRN appears in 5 different GRNs able to solve the *2-skip-1* pattern.

The significance of these findings is that the possibilities for evolutionary co-option at the module and especially subcircuit levels are vast. Modules evolved for one purpose can be further evolved by the addition of new proteins or new interactions between existing proteins to form a new GRN for another purpose. Subcircuits utilized within one GRN can be duplicated and utilized by another GRN. An earlier modeling study has shown that gene duplication opens many possibilities for GRN evolution [19]. Significantly, subcircuits do not have to be independently functional to expand the potential of the evolutionary search space. The potential of co-option is the creation of an expanded GRN capable of solving a completely different patterning problem. The GRN evolved by co-option is a mixture of the old and the new.

Although the focus of this work was on pattern formation during development, the conclusions concerning the wide space of GRNs available for evolutionary discovery and the abundant opportunities for co-option to increase complexity are likely to apply to other fundamental biological processes, including the evolution of intracellular signal transduction cascades and metabolic networks.

V. ACKNOWLEDGEMENTS

The authors thank the anonymous reviewers for their excellent comments that helped clarify this paper. Thanks to the Computer Science Department, the Biology Department and the Center for Integrated BioSystems for support.

References

[1] E. H. Davidson and D. H. Erwin (2006) “Gene Regulatory Networks and the Evolution of Animal Body Plans,” *Science*, vol. 311(5762), pp. 796-800.

[2] G. von Dassow, E. Meir, E. M. Munro, and G. M. Odell (2000) “The segment polarity network is a robust developmental module,” *Nature*, vol. 406(6792), pp. 188-192.

[3] S. Keranen (2004) ”Simulation Study on Effects of Signalling Network Structure on the Developmental Increase in Complexity.” *J. Theor. Biol.* vol. 231 pp. 3-21.

[4] S. Istrail and E. H. Davidson (2005) “Logic functions of the genomic cis-regulatory code,” *Proc. Natl. Acad. Sci.* vol. 102(14), pp. 4954-4959.

[5] M. Levine and E. H. Davidson (2005) “Gene regulatory networks for development,” *Proc. Natl. Acad. Sci.* vol. 102(14), pp. 4936-4942.

[6] N. Flann, J. Hu, M. Bansal, V. Patel and G. Podgorski (2005) “Biological Development of Cell Patterns: Characterizing the Space of Cell Chemistry Genetic Regulatory Networks,” in *Lect. notes comput. sci.*, Berlin: Springer, vol. 3630, pp. 57-66.

[7] T. Quick, C.L. Nehaniv, K. Dautenhahn (2003) “Evolving Embodied Genetic Regulatory Network-driven Control Systems”. Banzhaf et.al., eds: Advances in Artificial Life, LNAI vol. 2801, 266-277.

[8] J. C. Bongard (2002) “Evolving Modular Genetic Regulatory Networks”. Fogel et.al., eds: Proceedings of the IEEE 2002 Congress on Evolutionary Computation, 1872-1877.

[9] J.T. Kim (2001) “Transsys: A Generic Formalism for Modelling Regulatory Networks in Morphogenesis”. Kelemen and Sosik, eds: Advances in Artificial Life, LNAI vol 2159, pp. 242-251.

[10] O. S. Orkun and S. Bonhoeffer (2006) “Evolution of complexity in signaling pathways,” *Proc. Natl. Acad. Sci.* vol. 103(44) pp. 16337-16342.

[11] J. R. Collier, N. A. M. Monk, P. K. Maini, and J. H. Lewis (1996) “Pattern Formation by Lateral Inhibition with Feedback: a Mathematical Model of Delta-Notch Intercellular Signalling,” *J. theor. Biol.*, vol. 183, pp. 429-446.

[12] S. Basu, Y. Gerchman, C. H. Collins, F. H. Arnold and R. Weiss (2005) “A synthetic multicellular system for programmed pattern formation,” *Nature* vol. 434, pp. 1130-1135.

[13] H. G. E. Hentschel, T. Glimm, J. A. Glazier and S. A. Newman (2005) “Dynamical mechanisms for skeletal pattern formation in the vertebrate limb,” *Proc. R. Soc. Lond. B 04PB0165.1*

[14] A. Ghazi and K. Vijay Raghavan (2000) “Developmental biology:Control by combinatorial codes,” *Nature*, vol. 408, pp. 419-420.

[15] E.H. Davidson (2006) *The Regulatory Genome: Gene Regulatory Networks in Development and Evolution*, First Edition, New York: Elsevier.

[16] P. C. Nguyen, T. Washio, K. Ohara and H. Motoda (2004) “Using a Hash-Based Method for Apriori-Based Graph Mining,” in Lect. notes comput. sci., vol. 3202. Berlin: Springer, pp. 349-361.

[17] S.A. Kaufman, (1993) “The Origins of Order: Self-Organization and Selection in Evolution,” Oxford University Press, New York.

[18] D. Roggen and D. Federici (2004) “Multi-cellular Development: Is There Scalability and Robustness to Gain?” in *Lect. notes comput. sci.*, Berlin: Springer, vol. 3242, pp. 391-400.

[19] A. Wagner (1994) “Evolution of gene networks by gene duplications: A mathematical model and its implications on genome organization”. PNAS 91: 4387-4391.