Computational Evaluation of Planarian Regeneration Models

Marianna Budnikova, Jeffrey W. Habig, and Tim Andersen
Computer Science, Boise State University, Boise, ID, USA

Abstract—We have combined a cell-based modeling platform with a database of formally encoded morphological experiments and their outcomes in order to automate the search for computational models of cellular networks involved in maintenance of morphology and to validate such models against existing experimental outcomes. To achieve automated validation of virtual cellular models, we have developed an algorithm for converting a formal graph representation of planarian morphologies into a polygon representation which can be overlayed with a cell-based model. We show how the overlay algorithm can be used as a basis for the fitness function to guide automated searches for models of cellular development.

Keywords: planaria, regeneration, tissue modeling

1. Introduction

For the last 200 years scientists have been trying to understand and explain the processes behind the regeneration of planaria worms [1], [2]. A large amount of experimental data has been accumulated on planarians over the years, and a plethora of planarian regeneration models have been proposed [2] [3] [4]. However, the proposed models are able to explain only a few experimental outcomes. The studies performed on planarians could greatly advance the understanding of planarian regeneration if they were united in one central database and combined with a computational search engine. Recently, the Levin lab introduced a graph-based formalism for a database to store planarian experiments. [5]. Flexible graph notation allows organisms to be described in terms of nodes connected by links at specific angles. Due to the flexibility of the graph formalism, the PlanformDB database allows searching the experiments performed on planaria not only by keywords, but also by the worm’s shape as the key. PlanformDB can be used to create and validate new regeneration models if combined with an automated generate-and-test tool [6]. A great example of such a tool is the cell-based modeling platform used by the Andersen lab to discover models of planarian regeneration by performing evolutionary search [7].

We are working on combining the PlanformDB and the CellSim evolutionary search engine to develop an automated system for searching and validating computational models of development. The automation calls for mechanisms to compare individuals produced in the simulation platform to morphologies taken from the database. This comparison method will serve as a fitness function to guide the evolutionary search for planarian regeneration models. In this work we present a novel fitness function that compares individuals produced in CellSim to PlanformDB morphologies by creating a graph to cell overlay.

2. Graph Edit and Overlay Fitness Functions

During the evolutionary search, the CellSim platform simulates an experiment taken from PlanformDB and returns the cellular outcome of the experiment. In our experiments, the cellular morphology was approximated starting with a flat sheet of 420 autonomous cells, where each cell is capable of growing, dividing, dying and regulating metabolic and genetic networks in response to changes in their local environment (Fig. 1). The cellular morphology returned by the simulation platform is compared to the target graph from the PlanformDB database, yielding a fitness value that can be used to guide the evolutionary search.

We have been using a graph matching algorithm to calculate the fitness of the regeneration model. With this approach, a cellular morphology generated by the simulation platform is converted into graph representation and compared against the target graph using the graph edit distance technique [8]. The graph edit distance evaluator is a very flexible fitness function, since the graph formalism can be used in simulation platforms other than CellSim. However, the evaluator is not optimal for the purposes of the evolutionary search due to the coarse grained nature of how a cell region type is determined during the conversion of a cell-based morphology into graph. The assignment of a cell to a region is a very complex question, since many factors determine whether a cell belongs to a particular region. The component gathering algorithm used to convert a cellular morphology to graph assigns a cell to a region type based on the maximum concentration of the indicator molecule. This approach may not work well when a cell has several indicator molecules and thus the potential of becoming the sought-for region type. For example, consider two cells, where the head indicator concentration of the first cell is 0.1, the second cell 0.9, and both cells have the trunk indicator concentration of 1.0. Suppose the sought-for region type is head. The graph edit distance algorithm will assign both of the cells to the trunk region resulting in equal fitness, even though the second cell is closer to becoming the head type.

We would like to be able to reward cells that are close to becoming the sought-for type. In the example above, the second cell with 0.9 concentration of the head indicator should be rewarded more than the cell with 0.1 concentration. To approach this problem, we have designed and implemented an overlay evaluator as an alternative to the graph edit distance evaluator. The overlay works by taking the converted graph representation of the target morphology and pasting it onto the cellular outcome from the simulation platform. Once the overlay is created, each cell in the cellular morphology can be examined individually against the closest graph region.
In contrast to graph edit distance where a cell is labeled as one region, a cell in the overlay evaluator is examined in respect to many of the cell’s features. Therefore, the overlay evaluator can reward a cell that is close to becoming the expected region type. The current graph formalism as presented in the Planform database is not conducive to being used as an overlay. For our overlay implementation we chose to cover every cell in the simulation platform outcome by the overlaying structure. This overlaying structure was obtained by converting the region nodes of the graph morphology into polygons (Fig. 2). The algorithm starts by converting the graph into polygons. Then, the best fit of the graph polygons to a cellular morphology is calculated by scaling the graph so the graph’s height and width match those of a cellular morphology. For each cell in the cell-based morphology, the closest polygon region of the overlayed polygon graph is found. A subfitness value between 0.0 and 1.0 is calculated for each cell measuring how far the cell is from becoming the expected region. The final fitness of the model in the overlay difference evaluator is calculated by dividing the sum of the subfitness values for every cell by the number of cells.

3. Experiments and Conclusion

We have run several evolutionary searches with various parameters. To get the starting population for the search, different knockouts were performed on a hand-crafted gradient regeneration model, such as removing the promoters of head and tail in the genome and the head regeneration equations in the metabolic description of the model. Table 1 shows the kinds of solutions that the GA found. Morphology with ID 1 was found using the graph edit distance as a fitness function, and individual with ID 2 was found with the overlay evaluator.

Table 1: Graph Edit Distance and Overlay Evaluator Fitness Values for Different Morphologies

<table>
<thead>
<tr>
<th>ID</th>
<th>Morphology</th>
<th>Graph Edit Distance</th>
<th>Overlay</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><img src="image1" alt="Morphology Image" /></td>
<td>0.983</td>
<td>0.817</td>
</tr>
<tr>
<td>2</td>
<td><img src="image2" alt="Morphology Image" /></td>
<td>0.587</td>
<td>0.978</td>
</tr>
</tbody>
</table>

The first morphology had only one cell of head and tail regions regenerated. Graph edit distance fitness function converted those cells into full regions, and assigned a high fitness to this morphology. This fitness assignment is not intuitive, since one would expect a head or a tail region to have more than one cell to be considered a head or a tail. The overlay evaluator matched our intuition and assigned a lower fitness of 0.817 to this individual. The second morphology found by the GA search closely resembled the target model, except that it had a single head generated in the tail. The graph edit distance assigned this morphology a very low fitness value, since the algorithm considered the single head cell as an extra added region. When we examined this cell, it had 32.0 concentration of head indicator, and 14.0 concentration of tail indicator, which means it was slowly evolving into tail region. From these experiments, it can be seen that the overlay evaluator produced more intuitive fitness values from evolutionary search prospective than the graph edit distance evaluator. In future work, we plan to run more complex searches where we knock down the equations responsible for entire region regeneration.

References