

Dynamic Structure Discovery and Repair for 3D Cell Assemblages

Giordano B. S. Ferreira¹, Max Smiley³, Matthias Scheutz¹ and Mike Levin²

¹Department of Computer Science, Tufts University, Medford MA, USA

²Department of Biology, Tufts University, Medford MA, USA

³Microsoft Research, Seattle WA, USA

Abstract

Many organisms can regenerate their bodies, but it is currently unclear how they accomplish this feat. In this paper, we introduce a cell-to-cell communication mechanism that allows a 3D arrangement of cells to discover its structure and maintain it in the light of random cell death, even at very high death rates. We report results from simulations of an agent-based model that demonstrate the effectiveness of the proposed approach for Planarian worm-like shapes, but the proposed model is general and applies to any shape.

Introduction

Biological organisms have the ability to regenerate themselves (Birnbaum and Alvarado, 2008), i.e., they are able to detect and reproduce damaged cells that make up their morphological structure. In some cases, whole body parts (e.g., limbs, tail, etc.) can be regenerated and the question arises how this information is encoded and where it is stored (Friston et al., 2015; Pezzulo and Levin, 2015)? While current orthodoxy would still point to genetic encodings and thus morphological information being stored in and recovered from gene expressions, there is converging evidence that this might not be so, at least not in all cases (see the next section). Some of the evidence (reviewed in Lobo et al. (2014)) comes from studies where morphological changes performed on organisms were regenerated after they were lesioned (e.g., damage to deer antlers can result in ectopic growths at the same spot of injury and these growth persist through several subsequent shedding and regenerations of the deer's antlers (Bubenik, 1990)). Since there were no opportunities for genes to encode those initial morphological changes, the information must have been stored elsewhere. But if morphological information is not stored genetically, what other mechanisms could be accountable for representing the morphological structure of an organism?

In this paper, we propose a dynamic messaging mechanism that while not yet mapped on biological substrate can functionally explain how morphological information can be obtained, stored, and used to repair structural damages to organisms. Specifically, different from genetic encodings

where information is local to each cell, statically encoded in the gene and thus retrievable only locally, the proposed mechanism is distributed, dynamic, and integrates information across cells. Hence, it is able to detect when cells are missing in a structure and start a regrowth process that generates exactly and only the missing parts. We will demonstrate the operation of the mechanism using an agent-based model of cell-to-cell communication and prototypical 3D organismal shape of a flatworm and show that for various rates of random cell destruction (e.g., due to radiation) the organism is able to maintain its structure. In concluding, we discuss next steps for further simulations and validations of the employed principle.

Background and Previous Work

One of the major questions facing biology and biomedicine is how groups of cells cooperate to build and maintain complex anatomical structures. In many animals, this occurs over long time-spans, counteracting aging, carcinogenic transformation, and tissue abrasion. An understanding of the information structures and algorithms that keep cells orchestrated towards maintaining a large-scale body-plan would be very important for regenerative medicine, aging research, and degenerative disease, as well as having basic implications for understanding pattern regulation in evolution (Ingber and Levin, 2007). regenerative biology is the Planarian flatworm – a complex bilaterian organism that regenerates amputated pieces, and continuously maintains its bodyplan despite significant turnover and remodeling (Oviedo et al., 2003). While progress is being made with models of gene regulation (Lobo and Levin, 2015), we still seek testable models of cellular communication that explain pattern memory (Tosenberger et al., 2015). Deriving generative, fully-specified models of pattern regulation in this kind of model species is an essential goal for converting molecular-genetic insights into actionable strategies for manipulating growth and form in regenerative medicine, birth defects, cancer, and synthetic bioengineering (Doursat et al., 2013).

The problem of structural maintenance has been ap-

proached by the artificial life community using genetic algorithms, agent-based models and cellular automata to model the behavior of how a single cell could multiply and generate a whole tissue, and after some time, this tissue could maintain its shape against some external or internal perturbations. Andersen et al. (2009), for example, used a genetic algorithm to evolve a gene regulatory network which controls the behaviors of cells. The authors put specific shapes that they want to create in their fitness function, thus the GA could find a network (i.e., genotype), which, starting using a single cell, generates that specific shape (i.e., phenotype). They concluded that different networks can lead to the same phenotype, and more interestingly, the shape become capable of healing wounds even though this process was not encoded in the fitness function.

In Gerlee et al. (2011), the authors use a genetic algorithm to evolve a 3-dimensional cellular automaton that creates and maintains a mono-layer tissue structure. First, the authors wanted to show that a cellular automaton could evolve from a network containing just one cell, to a 2-dimensional structure similar to how epithelial cells are organized in most of the organs of the body. Further, the authors put some external and internal perturbations to verify if the model was capable of returning to its original structure. Similar to our proposed model, each cell is a discrete agent which interacts with its neighborhood, and in their paper, depending on its concentration of oxygen and a generic growth factor. However, in that model the network will grow indefinitely until it reaches a pre-defined area.

Basanta et al. (2008), also used a 3D cellular automaton to model the interaction among neighbors cells and used a genetic algorithm to find a good genotype to perform the procedure. In this work, a 3D shape is created based on the cellular automaton's rule coded in the genes of all cells, and at some point, some genotypes achieve a state of homeostasis. After that, lesions were performed on the shape, and some genotypes were able to regenerate their structures. The authors verified that the organisms which perform best in this "wound recovery" were the ones which had a specific direction by which the cells evolved in the tissue creation.

Overall, past approaches (to the extent that we could find) used some kind of genetic encoding to define how interactions among cells should take place. Thus, cells behaviors depend on their neighborhood and are encoded in the genotype. Our proposed model, on the other hand, does not rely on any genetic encoding, because the behavior of the cells depends on the messages they receive. The critical advantage is that our model does not have any local storage of shape data nor does it rely on it; rather, it can dynamically learn and maintain new morphologies using the same underlying mechanism.

The Communication Model

We start by first presenting the idea of the proposed cell-to-cell communication mechanism, followed by the detailed agent-based model implementing it based on the now customary ODD (Overview, Design concepts, Details) protocol (Grimm et al., 2010).

Discovery and Regeneration

The purpose of the agent-based model is to investigate possible structure monitoring and regeneration process for 3D cell structures, possibly resembling organismal bodies such as the Planarian flatworm. Specifically, we intend to propose mechanisms for such 3D cell structures to *dynamically discover* their morphology and then maintain it indefinitely in the light of random damages happening to parts of it such the damages that occur as part of natural aging. The basic idea is that cells can send messages to other cells or forward messages they receive from other cells which contain information about the path they traveled. This information can then be checked as a packet travels through the body's cells and if a cell along the way is missing, it must have been damaged and thus needs to be repaired. To illustrate how this works, consider the 2D arrangement of cells in Fig. 1.

The packet originating at the bottom right cell during "discovery" (where packets are randomly generated and only those are kept whose paths actually reflect paths that can be taken) consists of three segments of variable length – (0,4), (1,2), and (0,2) – where the first number in each pair indicates a direction (with 0 being West, 1 being SW, 2 being SE, and so on) and the second indicates the distance measured in cells (4 means four cells across). Thus, this packet structure can specify arbitrary paths with up to two directional changes to cover cells in a 2D arrangement.

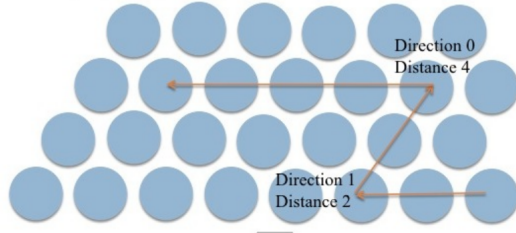
If that packet were now to retrace its path back to its origin in a lesioned structure and thus could not find the fourth cell in a row as predicted by its (0,4) segment, this detection failure could be used by the cell where the packet got stuck to grow a new cell in the missing position (as the old cell residing there must have died). The regrowth now allows the packet to complete the first segment of its path and the same process of regeneration repeats itself for the second and third segments up until all missing cells along the path have been recovered. Note that not all missing cells were regenerated, only those discovered by the particular packet along its path. For the other missing cells to be regenerated, additional packets with paths going through them would be needed.

The 3D Spatial Agent-Based Model

The proposed ABM model has just one type of agent representing the cells of the organism. Each agent has some attributes that describe them at a given time. Each agent i is defined by an unique identity number i_{id} and its location on the organism's body $\langle i_x, i_y, i_z \rangle$.

Example packet generated during Discovery stage

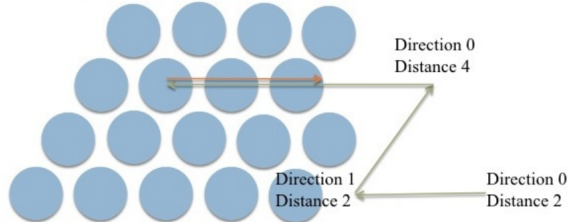
Packet: (0,4), (1,2), (0,2)



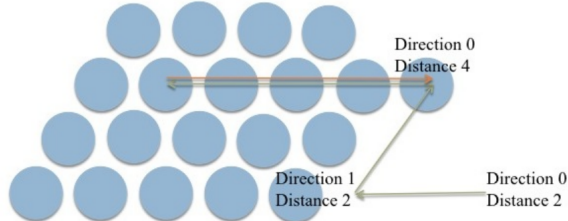
Damage to Structure

Example packet backtracking during Repair stage

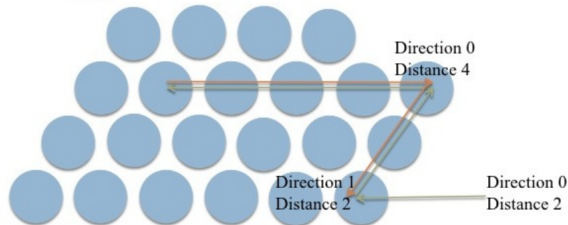
Packet: (0,4), (1,2), (0,2)



Packet: (0,4), (1,2), (0,2)



Packet: (1,2), (0,2)



Packet: (0,2)

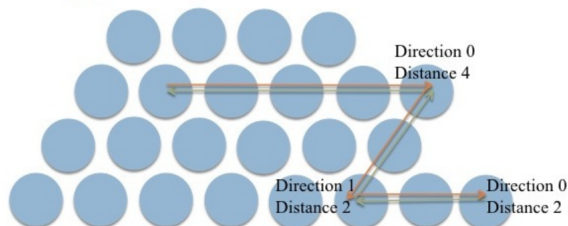


Figure 1: Example of cell discovery, damage detection, and repair.

The specific shape of the evaluation organism, a Planarian worm, is a 3D structure called rhombic dodecahedral honey-

comb. One can imagine each cell as an hexagon with three other hexagons stacked above it and three other hexagons behind. Therefore, each cell is a rhombic dodecahedron hence it has at most 12 neighbor cells which is stored in a list $i_{Neighbors}$.

Cells hold and send packets to its neighbors. A packet β contains a list of vectors β_V , of distance and direction that describe the path that a packet has traveled across the cell network. The packets are organized in temporal order, with the most recent vector at the top of the list. Thus, each cell contains one list containing the packets received from its neighbors during a cycle $i_{ReceivedPackets}$ and a list of packets the cell is holding $i_{HeldPackets}$.

Each vector v has an integer $v_{Distance}$ representing the number of cells the vector traveled through, $v_{Direction}$ representing one of the twelve directions in which the vector traveled, and v_{Mode} which stores whether the packet is charting its path and adding to v or backtracking and taking data from v .

At each cycle, each cell generates $PacketFreq$ packets and sends them to adjacent cells in random directions. When a cell receives a packet, it increments its top vector's distance. For each packet, received in a given cycle, the cell will either (1) send the packet along the same direction as its top vector's direction, (2) send the packet in a new direction, or (3) hold the packet. In order for a cell to hold a packet, this packet must have at least $MinVectorsToHold$, and the top vector must have a distance of at least $MinTopLen$. If the packet is not held, there is a $BendProb$ probability that the packet will be sent in a new direction. This new direction should be different from the opposite direction of the top vector's direction.

When a packet reaches a cell, the cell verifies the number of bends until that moment. If this value is equal to $MinBends$, then the packet will backtrack, regenerating dead cells during this process.

The model runs as a discrete-time simulation for a defined number of cycles, $RunCycles$. And at each cycle, the cells perform only two processes: sensing and acting. In the first they receive packets from their neighbors and decide if the packets will be held or sent (see Algorithm 1). Moreover, if a packet is backtracking, and the neighbor is dead, then that neighbor is regenerated during sensing process. The acting process is just the cells sending packets to their neighbors (see Algorithm 2).

In the proposed model, each cell creates packets to send to its neighbors, the only interaction between agents. This local interaction creates an emergent behavior of structure maintenance where cells along the travel path are restored. As long as some packets will eventually hit each dead cell, the system is guaranteed to keep the structure intact.

Regarding the stochastic procedure, there are two cases where they occur. First when a cell needs to decide the direction of a packet (a new packet or a received packet that

Algorithm 1 Pseudo code of the sensing process performed by the cells.

Sense(*i*)

```

for all packet $\beta \in i.ReceivedPackets$  do
  top  $\leftarrow \beta.TopVector$ 
  if top.Mode == Charting then
    top.Distance  $\leftarrow top.Distance + 1$ 
    if  $\beta.Bends \geq MinVectorsToHold$ 
    and top.Distance  $\geq MinTopLen$  and
    isAlive(i.Neighbors[top.Direction]) then
      i.HeldPackets.add( $\beta$ )
    else
      if random()  $\leq BendProb$  then
         $\beta.addVec(getNewDirection(top.Direction))$ 
      else
         $\beta.addVec(top.Direction)$ 
      end if
      i.SendingPackets.add( $\beta$ )
    end if
  else
    top.Distance  $\leftarrow top.Distance - 1$ 
    if top.Distance  $\leq 0$  then
      i.ReceivedPackets.pop()
    end if
    if top  $\neq nil$  then
      if isAlive(i.Neighbors[reverse(top.Direction)])
      then
        regenerateCell(i, reverse(top.Direction))
      end if
      i.SendingPackets.add( $\beta$ )
    end if
  end if
end for
if i.HeldPackets.size()  $\geq MinBends$  then
  for all packet $\beta \in i.HeldPackets$  do
     $\beta.Mode \leftarrow Backtracking$ 
    i.SendingPackets.add( $\beta$ )
  end for
end if

```

Algorithm 2 Pseudo code of the acting process performed by the cells.

Act(*i*)

```

for all packet $\beta \in i.SendingPackets$  do
  top  $\leftarrow \beta.TopVector$ 
  if top.Mode == Backtracking then
    top.Direction  $\leftarrow reverse(top.Direction)$ 
  end if
  if isAlive(i.Neighbors[top.Direction]) then
    sendPacket(i, top.Direction, packet)
  end if
end for
i.SendingPackets.clear()

```

needs to change direction). The second stochastic procedure is the random death of cells which will be explained in the next section.

Simulation Experiments

The goal of the experimental evaluation was to see whether the proposed cell-to-cell communication mechanism would be sufficient to maintain the structure of an organism over time in light of random cell death. The model was implemented in our Java-based agent-based *SimWorld* simulation environment (Scheutz and Harris, 2011).¹ For all simulations runs, we consider a prototypical 3D Planarian-like structure with a fixed shape of 8 layers containing 339 cells each, resulting in 2712 cells total (the top-most layer of cells of the employed shape is depicted Figure 2).



Figure 2: Shape of the topmost layer of the worm containing 339 cells.

To simulate the process of structural deterioration (e.g., due to a toxic or radioactive environment, or the natural aging and death of cells), we fixed a particular cycle in the simulation when this process would start to occur (*DeathTime*=80). At the moment that a cell dies, all held packets are lost and consequently it cannot transmit other packets that reach it later. To verify whether enough of the structure of the organism's body was still intact, we fixed the *Threshold* as 90% of alive cells for the entire simulation, i.e., for the organism to be considered "intact" at least 90% of its cells must be alive at any given cycle.

The function *isAlive* verifies if a specific cell is alive and can transmit packages. If a cell tries to send a packet which has its top vector in the *Backtracking* mode, and the neighbor cell supposed to receive this packet is not alive, then the alive cell calls the function *regenerateCell* which "revives" the dead neighbor. The function *getNewDirection* randomly chooses a new direction distinct from the direction passed as a parameter and also distinct from the reverse of this direction, to assure that the packet would not return from the cell it comes. Thus, *reverse* is a function that given the direction to one side of the dodecahedron, returns the direction to the reverse side of this polyhedron. Finally, the func-

¹SimWorld is a versatile environment with support for graphical and batch runs of models. It is easy to program and easy to extend, and it provides an interactive graphical interface user interface for inspecting agent behavior (and novel mechanisms for playing a simulation forward and backward, which supports the modeller in detecting interesting emergent behaviors). SimWorld has been under development in our lab for over a decade.

tion *sendPackets* adds the packet to the *ReceivedPackets* list of the cell that exists on the direction of the top vector.

Simulation runs can terminate in two different cases: for every cycle during the simulation, the organism must have at least *Threshold* percent alive cells otherwise the simulation stops. The second condition is when the simulation reaches the pre-determined limit of 500 cycles (if the organism can maintain its structure through 420 cycles, then we assume it can do so indefinitely, at least in approximation).

To explore the parameter space of the model, we first varied the probability of a cell dying on a given cycle (*DeathProb*) in order to simulate the death of cells as time passes. For our experiments, $DeathProb \in \{0.0, 0.01, 0.02, 0.03, 0.04\}$. For example, with a 2% death rate per cell per cycle, every cell will on average die every 50 cycles or 10 times in the course of the 500 cycle simulation. Since there are 2712 cells in the body, over 54 cells will die on average at any given cycle which is significant structural damage that accrues over time if not repaired quickly.

We also varied the number of new “packets” a cell produces on each cycle (*PacketFreq*). In our experiments, we varied $PacketFreq \in \{1, 4, 7, 10, 13, 16, 19, 22, 25, 28, 31\}$.

In order to control the variety of navigation patterns of packets, we varied the minimum size of the vector of bends before a packet can backtrack (*MinBends*); the minimum length the top vector of a packet should be to be able to bend, by adding a vector in new direction (*MinTopLen*); the probability that a packet will bend, given the top vector length is at least *MinTopLen* (*BendProb*). For our experiments, $MinBends \in \{1, 3, 5, 7\}$, $MinTopLen \in \{1, 3, 5, 7\}$ and $BendProb \in \{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$.

As our model has stochastic processes, we need to explore this parameter space using different random number generator seeds. Thus, for each point in the parameter space, we ran 8 different simulations resulting in a total of 47520 simulations. The dependent variable was the number of cycles the simulation ran with more than *Threshold* alive cells.

Results and Analyses

From 50688 different data points that have $DeathProb > 0.0$, 28961 points maintained a rate of 90% of cells alive during the whole simulation, i.e., 500 cycles. More specifically, 11801 points were with $rd = 0.01$, 9685 with $rd = 0.02$, 5802 with $rd = 0.03$ and 1673 with $rd = 0.04$ as shown in Figure 3. These results show that there exists a parameter space in which our model can repair death cells and maintain the individual’s structure indefinitely. The mean number of cycles with number of alive cells above 90% for all simulations was 388,521.

In order to compare the main effects of each independent variable on the cycles above threshold, we performed an ANOVA with *PacketFreq*, *MinBends*, *MinTopLen*, *BendProb* and *DeathProb* as independent variables and the *CyclesAboveMin* as dependent variable. The ANOVA shows

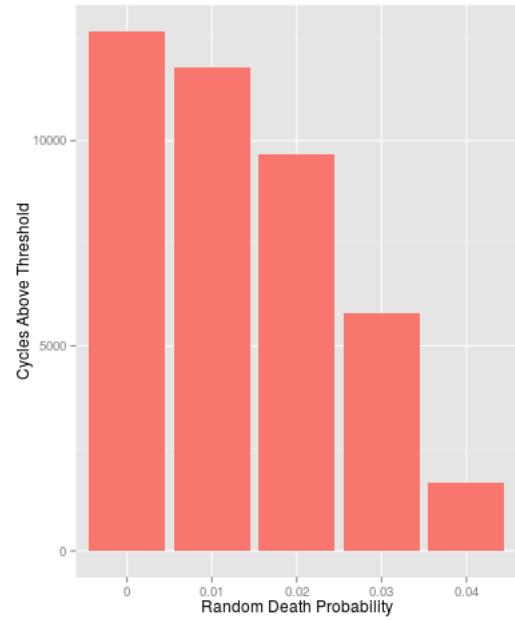


Figure 3: Histogram of points which maintain the structure of the worm after 500 cycles.

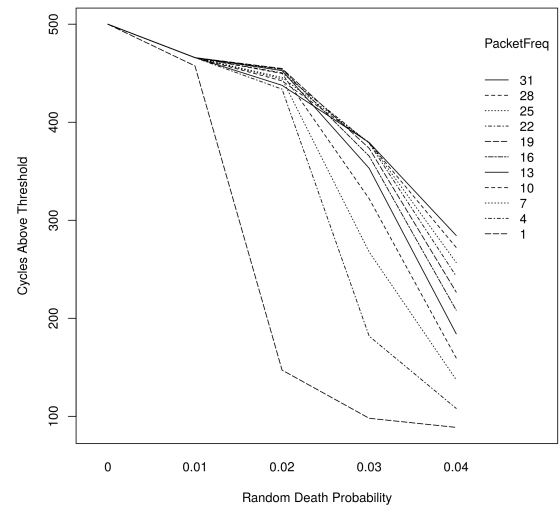


Figure 4: Interaction between RandomDeath and CyclesAboveMin for each number of produced packets per cycle.

significant main effects for all independent variable other than *BendProb*. Significant two-way, three-way and four-way interactions among variables other than *BendProb* were also found. These results confirm our hypothesis that variance in the packet vector is not relevant for the process

of structure maintenance.

As expected, there was a significant negative effect between *DeathProb* and *CyclesAboveMin* as shown in Figure 4. We also found a positive correlation with *PacketFreq* and *CyclesAboveMin* as shown in Figure 5. Increasing the value of *PacketFreq* means more variations of possible packets are explored, and from a certain point on increasing this value will generate more redundant packets than novel ones, maintaining an asymptote. The point by which the performance does not change depends on the probability of cell death, because as we increase the probability of cells deteriorating, more packets are not redundant; more specifically, more packets are necessary to maintain the structure of the organism.

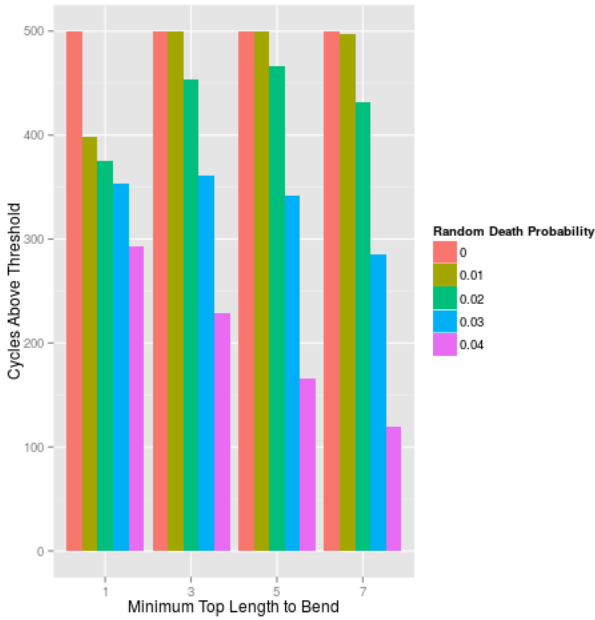


Figure 5: Interaction between *PacketFreq* and *CyclesAboveMin* for each probability of a random cell death.

Regarding the *MinBends* (see Figure 6), a moderate level (*MinBends* = 3) of minimum bends (before a packet can backtrack performed best. This value shows the optimization between the tradeoff of a longer packet covering a large area of the individual but also being more at risk of losses happening due to random cell deaths.

Figure 7 shows the interaction between *MinTopLen* and *CyclesAboveMin*. It is important to note that as *MinTopLen* increases, the length of the packet increases, therefore the packet must spend more time traversing before backtracking. Consequently, it increases the chance of a packet being lost to random death before it can repair another dead cell. Thus, for lower values of *RandomDeath*, it is good to have high values of *MinTopLen*. However, increasing *RandomDeath* confirms the tradeoff between better coverage

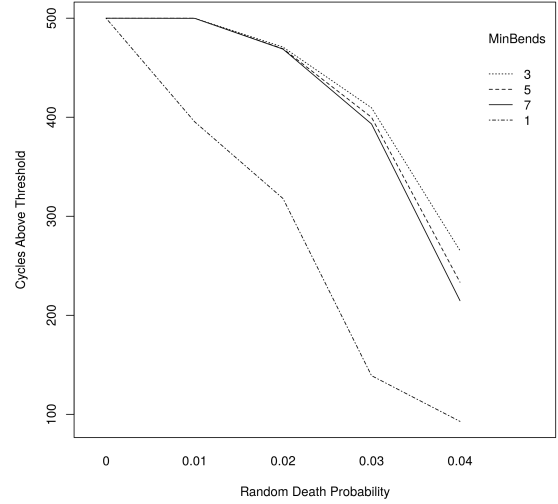


Figure 6: Interaction between *MinBends* and *CyclesAboveMin* for each probability of a random cell death.

and the risk of longer packets shifts, and for *RandomDeath* = 0.04, *MinTopLen* = 1 performs best.

The interaction between *MinTopLen* and *MinBends* shows an optimal combination with *MinTopLen* = 1 and *MinBends* = 3 (see Figure 8). Inverting the values of the two variables reduces the performance, even though these two combinations yield the same total packet length. The explanation for this is that the same length of packet can cover a wider space if it has more bends. This tradeoff is most pronounced when changing from a single bend to two bends.

Discussion

Our results show that organisms were able to maintain their structure using the proposed cell-to-cell communication mechanism for the right set of parameters: a high *PacketFrequency* > 22, a moderate *MinBends* = 3, a low *MinTopLen* = 1, and the value of *BendProb* not being relevant. Without modifying the algorithm, we hypothesize that it is possible to regenerate the worm from various more systematic cuts as well where a large part of the body is removed. For such a lesion to be healed, the packets residing in alive cells in the remaining body would have to be such that their collective paths would cover all excised cells which would then be regenerated during backtracking.

Space limitations allowed us to only discuss one particular structure but the proposed mechanisms are general enough to work for a very large set of structures. Whether a structure will be maintainable will effectively depend on both how cells die (e.g., randomly or because of lesions cutting of whole segments of the body) and how many bends packets can have which they will need to recover complex struc-

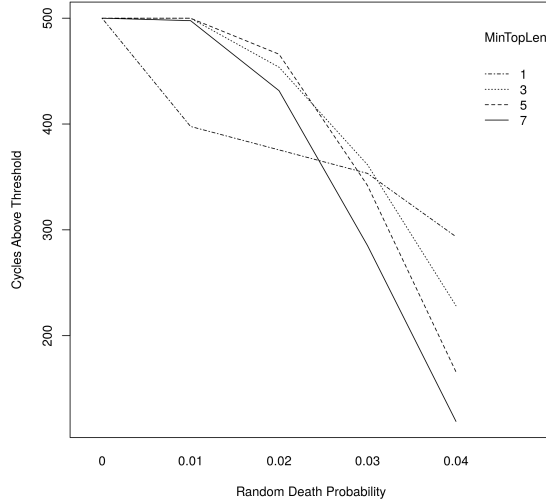


Figure 7: Interaction between MinTopLen and CyclesAboveMin for each probability of a random cell death.

tures that require many path segments to hit all component cells (e.g., to regenerate a cut-off arm packets need to travel through the upper arm, the lower arm, the wrist, the palm, and the various finger segments, thus requiring a larger number of segments in the packet).

Bi-directional cell communication *in vivo* takes place via several kinds of physical media (Edelstein et al., 2016), including chemical signals (diffusible molecules), physical forces (pressures and tensions), and bioelectric signaling (voltage gradients) (Levin, 2012). The latter is especially interesting because it enables many of the functions described in our model (Funk, 2013). Indeed, brains evolved by specializing such communication functions that were present from the dawn of multicellularity, and optimizing it for communication and message-passing functions in the central nervous system (Keijzer et al., 2013). The more ancient form, developmental bioelectricity (Bates, 2015) is a modality by which collections of cells communicate, store memory, and make group decisions about growth and form during embryogenesis and regeneration (Pezzulo and Levin, 2015). Using proteins such as ion channels and pumps, cells regulate their bioelectric dynamics (Levin, 2014; Mustard and Levin, 2014). However, using electrical synapses (gap junctions), cells can detect the presence and physiological state of neighbors (Palacios-Prado and Bukauskas, 2009). Communication via gap junctions has recently been shown to exert significant instructive control over growth and form during regeneration in planaria and other model systems (Emmons-Bell et al., 2015).

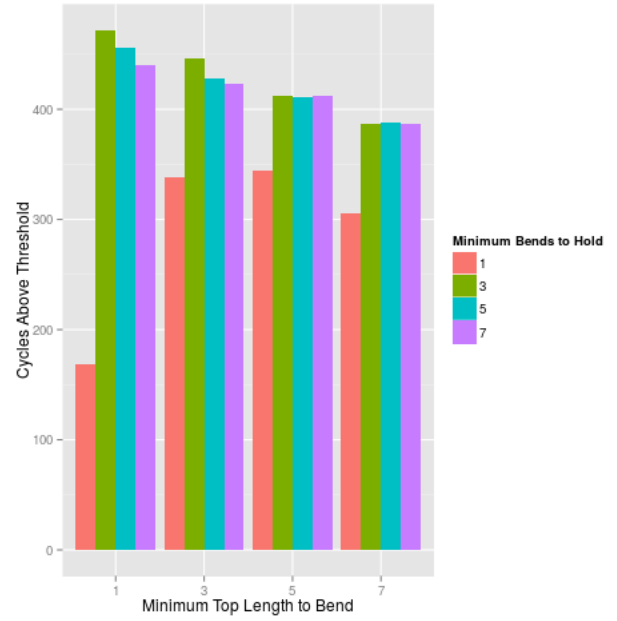


Figure 8: Interaction between MinTopLen and CyclesAboveMin for each minimum number of bends before backtracking.

Conclusion

In this paper we introduced the first agent-based model of structure discovery and repair which allows 3D cell structure to *discover their organization and repair it from damages occurring due to cell death*. We demonstrated the efficacy of the mechanism in large set of simulations of random cell death occurring at different rates in simulated body shaped as a Planarian. For even high cell death rates, we found parameters for the proposed cell-to-cell communication mechanisms that could maintain the structure indefinitely.

As a next step, we would like to verify how the model behaves with non-equally distributed cell death, i.e., where a cluster of adjacent cells dies at the same time due to, for example, the action of some toxin or an impact on a specific area of the organism. If for all dead cells there is a remaining packet held by an alive cell, then all cells can be regenerated.

In addition, we intend to investigate the regeneration from cuts that *in vivo* worms present. It is well known that the Planaria is capable of regeneration from cuts $\frac{1}{279^{th}}$ of the intact animal volume (Morgan, 1898). Our hypothesis is that there exists a parameter assignment by which our model is capable of regenerating structure from simultaneous death of a large area of cells and also from any number of cuts for the right set of communication parameters.

References

- Andersen, T., Newman, R., and Otter, T. (2009). Shape homeostasis in virtual embryos. *Artificial life*, 15(2):161–183.
- Basanta, D., Miodownik, M., and Baum, B. (2008). The evolution of robust development and homeostasis in artificial organisms. *PLoS Comput Biol*, 4(3):e1000030.
- Bates, E. (2015). Ion channels in development and cancer. *Annual review of cell and developmental biology*, 31:231–247.
- Birnbaum, K. D. and Alvarado, A. S. (2008). Slicing across kingdoms: regeneration in plants and animals. *Cell*, 132(4):697–710.
- Bubenik, G. A. (1990). The role of the nervous system in the growth of antlers. In *Horns, pronghorns, and antlers*, pages 339–358. Springer.
- Doursat, R., Sayama, H., and Michel, O. (2013). A review of morphogenetic engineering. *Natural Computing*, 12(4):517–535.
- Edelstein, L., Fuxe, K., Levin, M., Popescu, B. O., and Smythies, J. (2016). Telocytes in their context with other intercellular communication agents. In *Seminars in Cell & Developmental Biology*. Elsevier.
- Emmons-Bell, M., Durant, F., Hammelman, J., Bessonov, N., Volpert, V., Morokuma, J., Pinet, K., Adams, D. S., Pietak, A., Lobo, D., and Levin, M. (2015). Gap junctional blockade stochastically induces different species-specific head anatomies in genetically wild-type girardia dorotocephala flatworms. *International journal of molecular sciences*, 16(11):27865–27896.
- Friston, K., Levin, M., Sengupta, B., and Pezzulo, G. (2015). Knowing one's place: a free-energy approach to pattern regulation. *Journal of The Royal Society Interface*, 12(105):20141383.
- Funk, R. H. (2013). Ion gradients in tissue and organ biology. *Biol Syst Open Access*, 2(105):2.
- Gerlee, P., Basanta, D., and Anderson, A. R. (2011). Evolving homeostatic tissue using genetic algorithms. *Progress in biophysics and molecular biology*, 106(2):414–425.
- Grimm, V., Berger, U., DeAngelis, D. L., Polhill, J. G., Giske, J., and Railsback, S. F. (2010). The odd protocol: a review and first update. *Ecological modelling*, 221(23):2760–2768.
- Ingber, D. E. and Levin, M. (2007). What lies at the interface of regenerative medicine and developmental biology? *Development*, 134(14):2541–2547.
- Keijzer, F., Van Duijn, M., and Lyon, P. (2013). What nervous systems do: early evolution, input-output, and the skin brain thesis. *Adaptive Behavior*, page 1059712312465330.
- Levin, M. (2012). Morphogenetic fields in embryogenesis, regeneration, and cancer: non-local control of complex patterning. *Biosystems*, 109(3):243–261.
- Levin, M. (2014). Endogenous bioelectrical networks store non-genetic patterning information during development and regeneration. *The Journal of physiology*, 592(11):2295–2305.
- Lobo, D. and Levin, M. (2015). Inferring regulatory networks from experimental morphological phenotypes: A computational method reverse-engineers planarian regeneration. *PLoS Comput Biol*, 11(6):e1004295.
- Lobo, D., Solano, M., Bubenik, G. A., and Levin, M. (2014). A linear-encoding model explains the variability of the target morphology in regeneration. *Journal of The Royal Society Interface*, 11(92):20130918.
- Morgan, T. H. (1898). Experimental studies of the regeneration of planaria maculata. *Development Genes and Evolution*, 7(2):364–397.
- Mustard, J. and Levin, M. (2014). Bioelectrical mechanisms for programming growth and form: taming physiological networks for soft body robotics. *Soft Robotics*, 1(3):169–191.
- Oviedo, N. J., Newmark, P. A., and Sánchez Alvarado, A. (2003). Allometric scaling and proportion regulation in the freshwater planarian schmidtea mediterranea. *Developmental dynamics*, 226(2):326–333.
- Palacios-Prado, N. and Bukauskas, F. F. (2009). Heterotypic gap junction channels as voltage-sensitive valves for intercellular signaling. *Proceedings of the National Academy of Sciences*, 106(35):14855–14860.
- Pezzulo, G. and Levin, M. (2015). Re-membering the body: Applications of computational neuroscience to the top-down control of regeneration of limbs and other complex organs. *Integrative Biology*, 7(12):1487–1517.
- Scheutz, M. and Harris, J. (2011). An overview of the simworld agent-based grid experimentation system. In Werner, D., Kurowski, K., and Schott, B., editors, *Large-Scale Computing Techniques for Complex System Simulations*. Wiley.
- Tosenberger, A., Bessonov, N., Levin, M., Reinberg, N., Volpert, V., and Morozova, N. (2015). A conceptual model of morphogenesis and regeneration. *Acta bio-theoretica*, 63(3):283–294.