

Alloreognition in Hydractinia
Complex Adaptive Systems
CS 591, Section 7
Assignment 3
Due: May 12, 2003

1 Introduction

In this assignment you will be implementing a two-dimensional grid-based model of a simple colonial organism known as Hydractinia. These colonies grow on two-dimensional surfaces (on the shells of hermit crabs). When two colonies grow into one another (known as tissue contact) there are several possible outcomes. By far the most prevalent outcome is that the two colonies will “reject” one another and “fight,” with one colony establishing dominance and destroying the other colony. A small proportion of the time, the two colonies will fuse together and become a single larger colony, and occasionally there is an intermediate outcome known as *transitory fusion* in which the two colonies initially fuse together and then fight. For now, we will not model transitory fusion. The ability of the colony to distinguish between its own tissues and those of another colony is known as *alloreognition*. Alloreognition is interesting in part because it might represent the ancestral stage of histocompatibility systems of vertebrates. On our class website (under Assignment 3), there are two scientific papers describing this biological organism and its alloreognition properties. One of the authors of these papers is Prof. Luis Cadavid of the UNM Biology Dept. He helped me develop this model and is available for some questions about the biological details. These papers are for your reference, but you are not required to read and understand all of the details, as I have abstracted the model away from some of the biological detail, in order to have a reasonable assignment.

What determines how two colonies of identical organisms interact? The interactions are partially determined by two linked genes, and partially determined by morphological characteristics which we will model as local state in individual cells. Interaction patterns are dictated by cell-to-cell interactions between cells on the periphery of the colonies.

2 Model Specification

Your model will consist of a two-dimensional array of locations (called *sites*) and cells (or *agents*). Each site can either be empty or it can be occupied by a single agent. In the following, we need to specify what an agent is, and rules for how agents replicate, die, move, and interact with other agents:

1. Agents: Consist of a genome and local state:
 - (a) The genome is diploid and has two genes, $gene_1$ and $gene_2$. Each gene has two alleles (two possible values). The alleles for $gene_1$ are: f and r . The alleles for $gene_2$ are α and β . Thus, there are 9 possible genotypes: $ff\alpha\alpha$, $fra\alpha$, $rr\alpha\alpha$, $ff\alpha\beta$, $fra\beta$, $rr\alpha\beta$, $ff\beta\beta$, $fr\beta\beta$, $rr\beta\beta$. These genes are used only for alloreognition.
 - (b) Each cell can be in one of three morphological states: Stolon, fused mat, and unfused mat. Stolon cells grow in tendrils, known as canals, and both kinds of mat cells are feeding structures that form a dense mat. Colonies with many canals and a small amount of mat are known as stoloniferous colonies. Colonies consisting of lots of mat and few canals are known as mat colonies. When a stoloniferous colony fights with a mat colony, the stoloniferous colony always wins. However, we will not model this directly.
2. Agent lifecycle: Each cell lives for t time steps (a parameter of your system) and then dies (you get to decide whether death should be a deterministic event after t time steps, or a probabilistic event with a mean of t). During their lifespans, agents replicate probabilistically, where the mean of the replication time, n , is less than the mean lifespan, t . Agents can only replicate, however, if there is a vacant site immediately adjacent (you will decide if immediate adjacency should be a Moore or a von Neuman neighborhood). Replication also depends on how many mat cells are in the vicinity of replicating cells. The idea here is that “feeding cells” are good for sustaining other cells. Thus, the probability $P(\text{replication})$ is computed based on n and

$\frac{m}{c}$ where c is a constant (you will need to play with this to get a reasonable value) and m is the fraction of mat cells within a two-cell distance of the replicating cell.

3. Agent replication: The daughter cell of an agent has identical morphology as its parent (unless you add a small amount of phenotypic mutation).
4. Agent death: When an agent dies naturally, it is deleted and the site it occupied become empty. When an agent is defeated in combat, the victorious agent replicates into its space, preserving the morphology of the victorious parent.
5. Agent movement: In our model, agents will not move. Once born, they will occupy the same site (location) for their entire lives.
6. Agent interactions: Agents from the same colony go through their lifecycle without interference from other agents of the same colony, except for the indirect constraint that replication requires a vacant site. However, on every time step each agent will need to determine whether or not its neighbors are from its own colony or from another colony. It does this using the allorecognition genes, as outlined below.

Agents from different colonies *interact* in a pairwise fashion if they are NEWS neighbors (a von Neumann neighborhood). Agent interactions are always pairwise. When two agents interact, there are three possible results:

- (a) Colony fusion: Colonies fuse (the absence of fighting) if they share a single haplotype, that is if they have at least one chromosome that is identical. For example, the genotype $f\alpha\beta$ will fuse with itself, and it will fuse with the genotypes such as $f\alpha\alpha$ and $rr\beta\beta$.
- (b) Colony rejection: Agents that do not share a haplotype fight with each other. For example, the genotype $f\alpha\alpha$ will fight with the genotype $rr\beta\alpha$. In the biological system, cells migrate towards the contact zone and when triggered, cells release a long thread (similar to a harpoon) which attempts to pierce the opponent cell. We will not model this mechanism directly. Instead, we will use the outcomes of individual cell vs. cell fights to approximate colony rejections. To determine the outcome of a single agent-to-agent fight, first check the cell morphologies:
 - i. A stoloniferous / fused mat interaction will result in fight, and the stoloniferous agent wins with very high probability regardless of colony size.
 - ii. A stoloniferous / unfused mat interaction results in a fight. The stoloniferous cell wins ONLY if it belongs to a large colony. For example, if the stoloniferous colony is larger than half the size of the unfused mat colony. If the stoloniferous colony is smaller than the threshold, then the unfused mat will win the fight. You may need to experiment to find an interesting threshold value.
 - iii. A stoloniferous / stoloniferous interaction results in the agent from the larger of the two colonies winning (in terms of the number of cells in each colony) with very high probability.
 - iv. A fused mat / fused mat interaction results in passive rejection, that is, there is no fight but the colonies stop growing when they encounter each other.
 - v. An unfused mat / unfused mat interaction, results in fight, where the larger colony wins with very high probability.
 - vi. An unfused mat / fused mat interaction results in fight where the unfused mat type wins.

When an agent defeats another agent in a fight, the victorious agent replicates itself into the cell of the opponent.

3 Details

1. Updating the state of your model: An important decision is how to update the states of the model. In a completely synchronous update method (like cellular automata), you would compute the “next state” for every agent or site in the model and then perform all the updates simultaneously. This update method will not work for our model, because only one agent can occupy a cell at a time, which cannot be enforced using the synchronous method. Another update method is asynchronous, in which a single agent is picked at a

time (with replacement) and it determines then carries out its actions (replicate, die, fight, etc.) completely before another agent is selected for update. This method has the drawback that one agent could be picked several times in a row while its neighbor is never picked. We will adopt a “quasi-synchronous” update method, in which every agent is selected once per “time step,” but the order in which the agents are updated is randomized (you will need a new randomization for each time step). Under our scheme, once an agent determines its actions (e.g., replicate or die) the actions are performed in the model (and the appropriate states are updated) before the next agent is selected (e.g., if the agent dies its site becomes empty). In the case of a fight, you will need to modify this rule slightly (that is, if one agent decides to fight, then the agent it fights with will have to fight at the same time).

2. Visual displays: Some sort of visual display will be crucial to running and interpreting your simulation. At a minimum, you can write out your files in PBM format and view using xv or gimp. If you have time to put a simple GUI on your model, you will find it much easier to experiment with.
3. Initial conditions: An important component of your simulation will be determining interesting initial conditions to seed your simulations. Here are some ideas for simple experiments with which to begin your experiments and debug your model:
 - (a) Start the simulation with exactly one cell and watch it grow to consume the entire grid of cells. Use the results of these runs to set some of the simulation parameters (e.g. n and the size of your grid).
 - (b) Start the simulation with two colonies of genetically identical organisms. Watch them grow into each other and fuse. Experiment with different morphology mixtures in the colonies.
 - (c) Start the simulation with two colonies of organisms, with different sets of allorecognition genes and morphology types. When the colonies encounter one another, watch the resulting fights to make sure your simulation is behaving reasonably.

Once you have completed some simple debugging runs such as these, you will be ready to tackle some of the questions listed below.

4 Questions

1. In the field, about 99.5% of encounters between different colonies result in rejection. What percentage do you observe in your experiments?
2. Will diversity of allorecognition be maintained over time or will one colony type dominate, a phenomenon known as competitive exclusion? First, ask the question without mutation of allorecognition genes, and then add in a low rate of mutation. What initial conditions lead to competitive exclusion and which ones lead to maintenance of diversity?
3. Paper, scissors, and rock game (e.g., see <http://www.worldrps.com>). Can you model the dynamics of the classic paper/scissors/rock game? (If you are not familiar with this game, please ask me or one of the students in the class.) Start with three types of colonies, two that will fuse and one that will reject (the rejecting colony is stoloniferous), and see what happens.
4. What happens when mutation is added? What happens if you add a simple rule for cells to switch their morphology? What is the role of mutation?

4.1 What to hand in

You are allowed to work with one partner, if you choose. You are expected to hand in a source listing of your working program, some snapshots of interesting runs that will convince me that your program works correctly, and a 3-5 page writeup. The writeup should discuss important implementation decisions you made, which questions you chose to study (from the list above, or others) and how you structured your runs to answer the questions, the results you obtained, a discussion of the results (including possible suggestions for how to improve the model), and your conclusions.