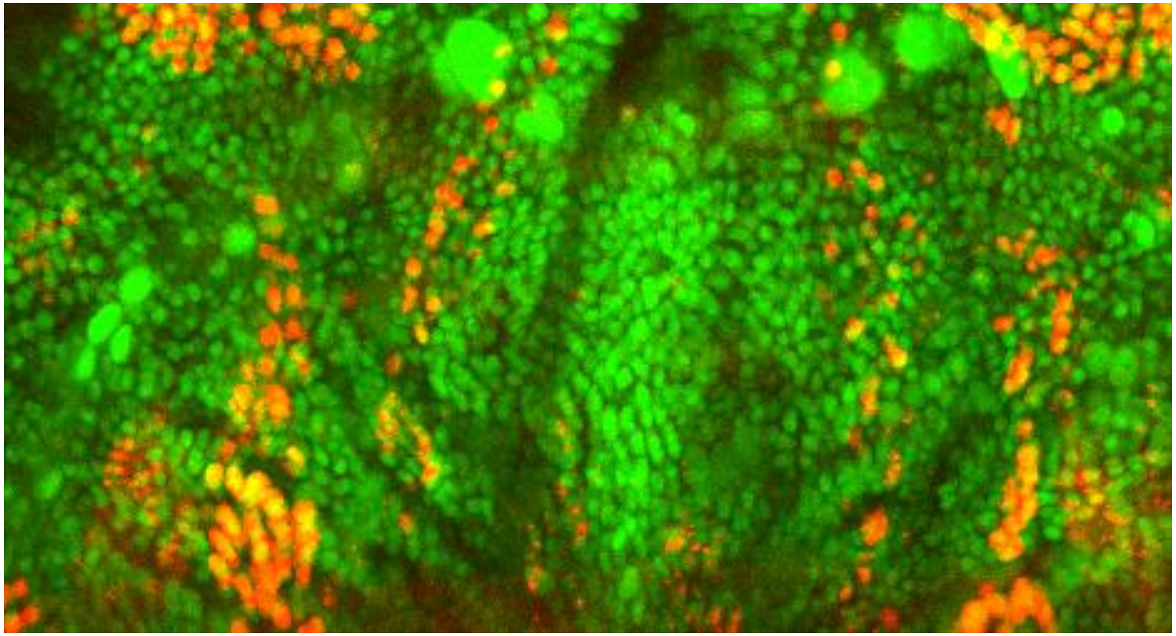
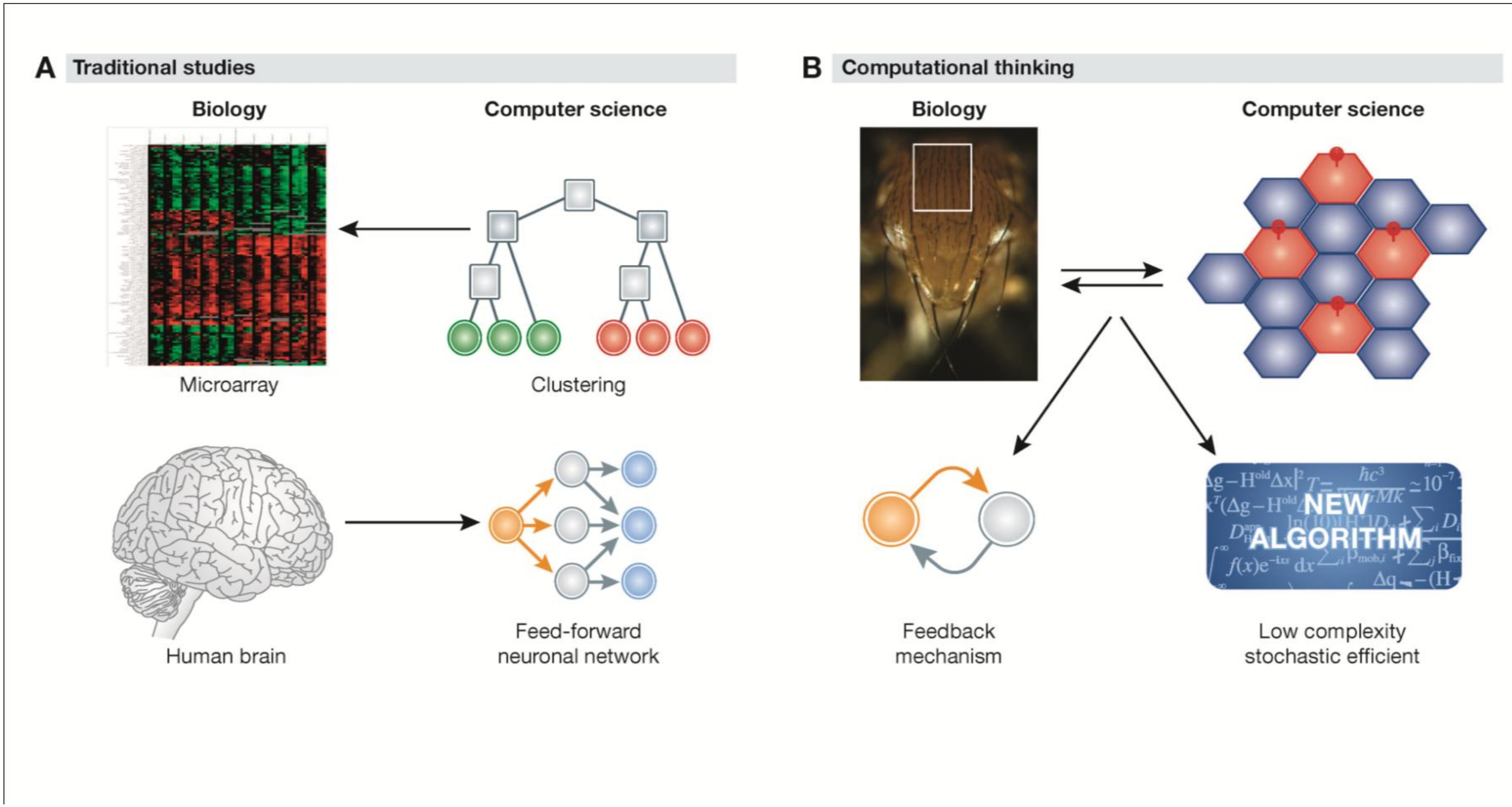
Robust computing over networks : Lessons from nature

Ziv Bar-Joseph
Lane Center for Computational Biology & Machine Learning
School of Computer Science
Carnegie Mellon University



Bi-directional studies



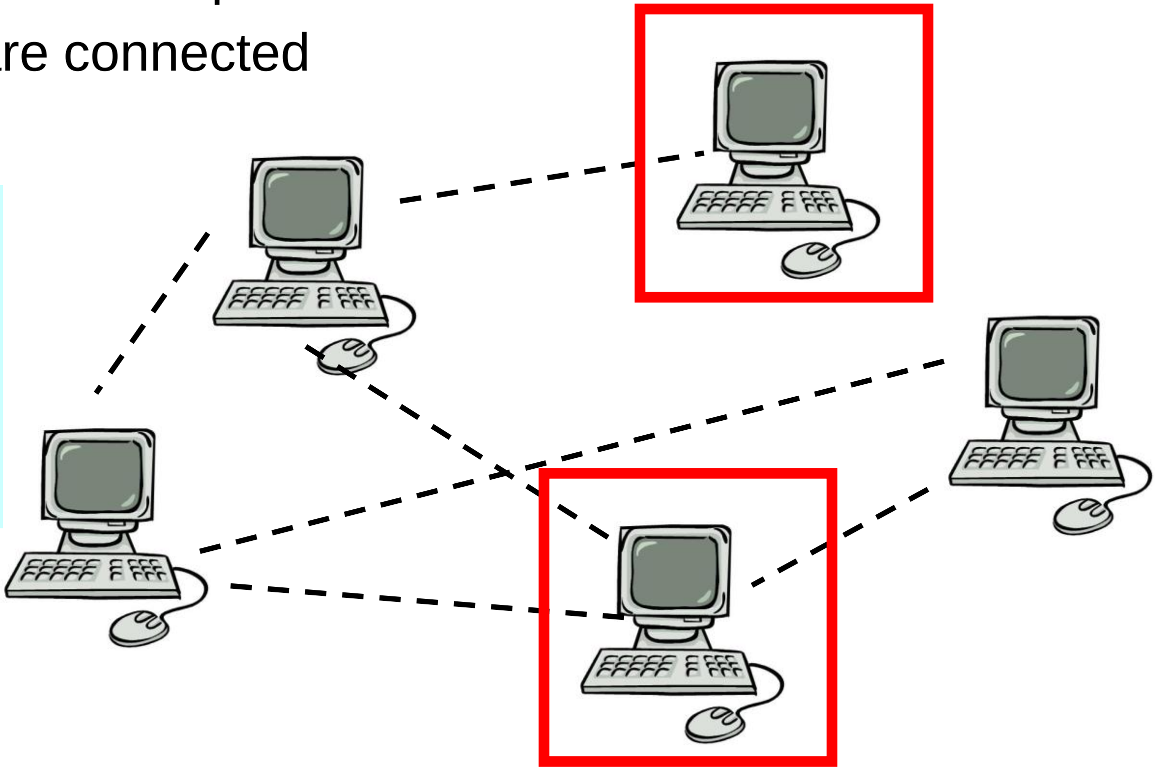
Navlakha and Bar-Joseph *Nature MSB* 2011

MIS and fly development

Maximal Independent Set (MIS)

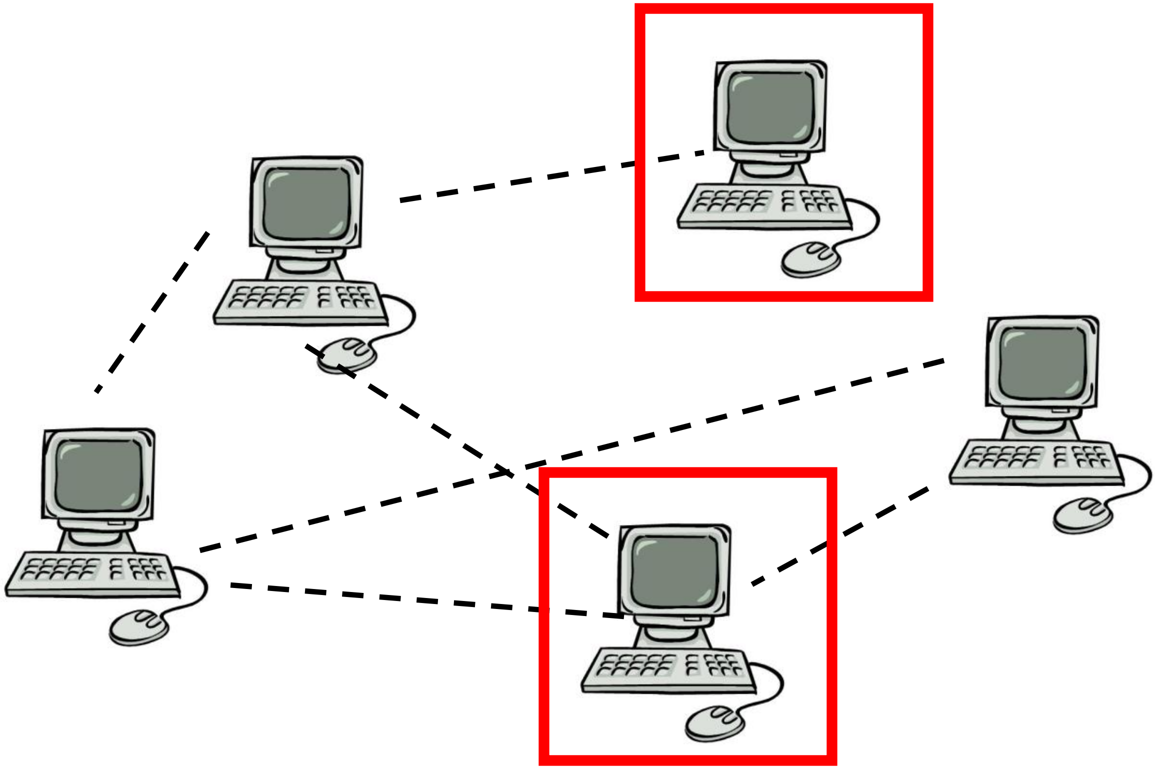
- A fundamental problem in distributed computing:
  - Often required to establish wireless networks
  - Used for routing messages, grouping sensors etc.
- For a set of nodes select a subset *A* such that:
  - Each processor is connected to a processor in *A*
  - No two processors in *A* are connected

"it is difficult to see how this problem can be solved in substantially fewer stages such as O(√*n*) …" (Valiant 1982)



Maximal Independent Set (MIS)

- Fast algorithms (e.g. Luby and Alon et al) exist for distributively selecting the MIS set but:
  - They Assume nodes know the status of their neighbors and also the topology of the graph (which is changing)
  - Use large messages



Algorithm for MIS

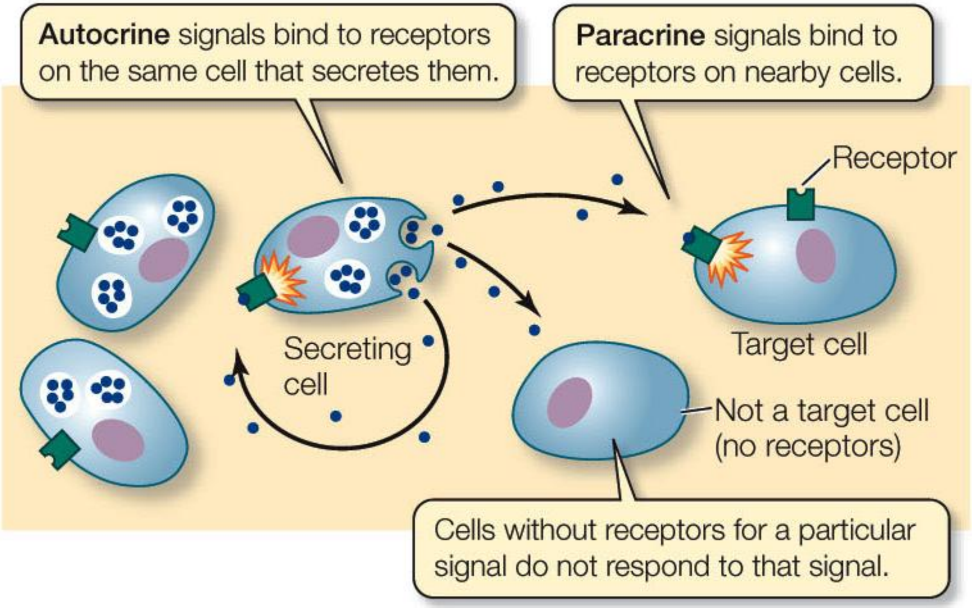
Each process needs to know how many neighbors it has

- Algorithm (proceed in rounds)
  - Each processor flips a coin with probability *1/d* where *d* is the number of its neighbors
  - If result is 0, do nothing
  - If result is 1, send to all other processors
    - If no collisions, Leader; all processes exit
    - Otherwise process with highest number of neighbors wins and becomes leader

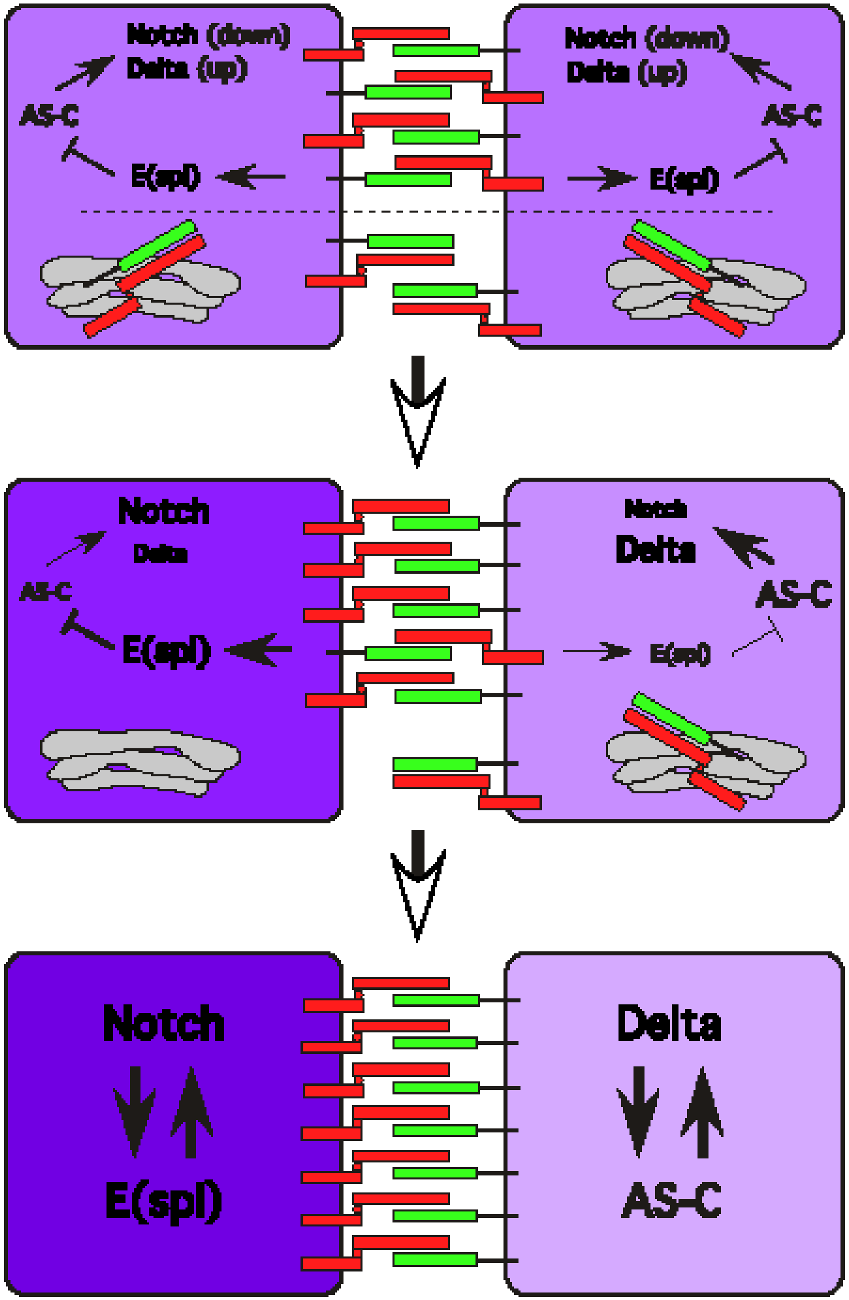
Each process needs to know how many neighbors its neighbors have

Luby ,*SIAM J. Comput.* 1986
Alon et al *J. Algorithms* 1986

Cell signaling



Lateral inhibition: Notch-Delta signaling

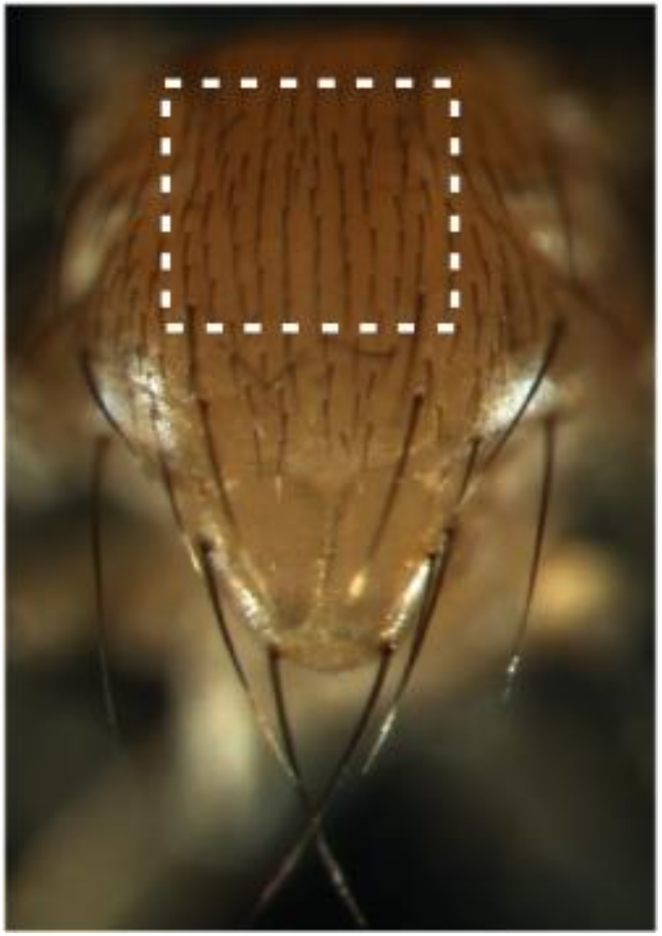


SOP selection in flies

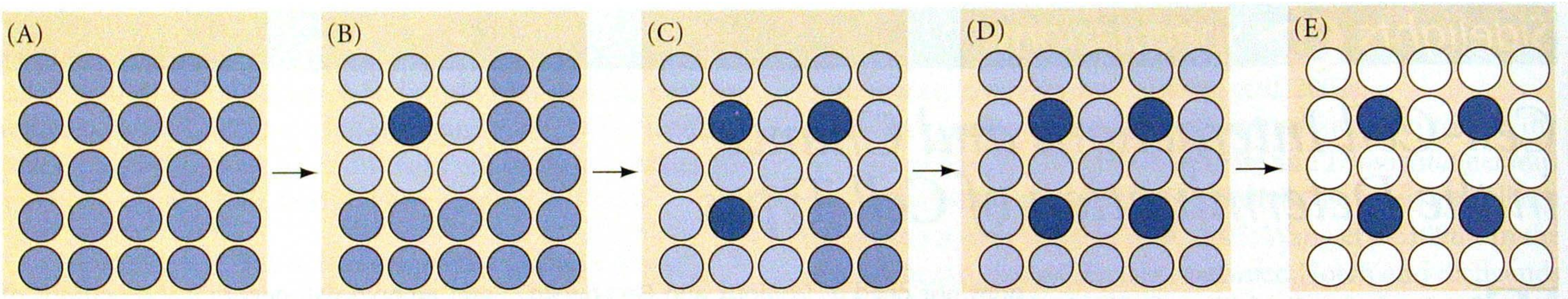
- When the fly’s nervous system develops several cells are selected as sensory organ precursors (SOPs)
- These cells are later attached to the fly's sensory bristles
- Similar to the MIS requirements, in a highly accurate process each cell in a predefined cluster is either:

  - Selected as a SOP

  - Laterally inhibited by a neighboring SOP so it cannot become a SOP



Generation of regular patterns of cells from equivalence groups



Similarities between MIS and SOP selection

- Both are performed using a stochastic processes

  - Proven for MIS, experimentally validated for SOP
- Both are constrained by time

  - A cell that is not inhibited by certain time becomes a SOP
- Both only send messages if a node (cell) decides to join *A*

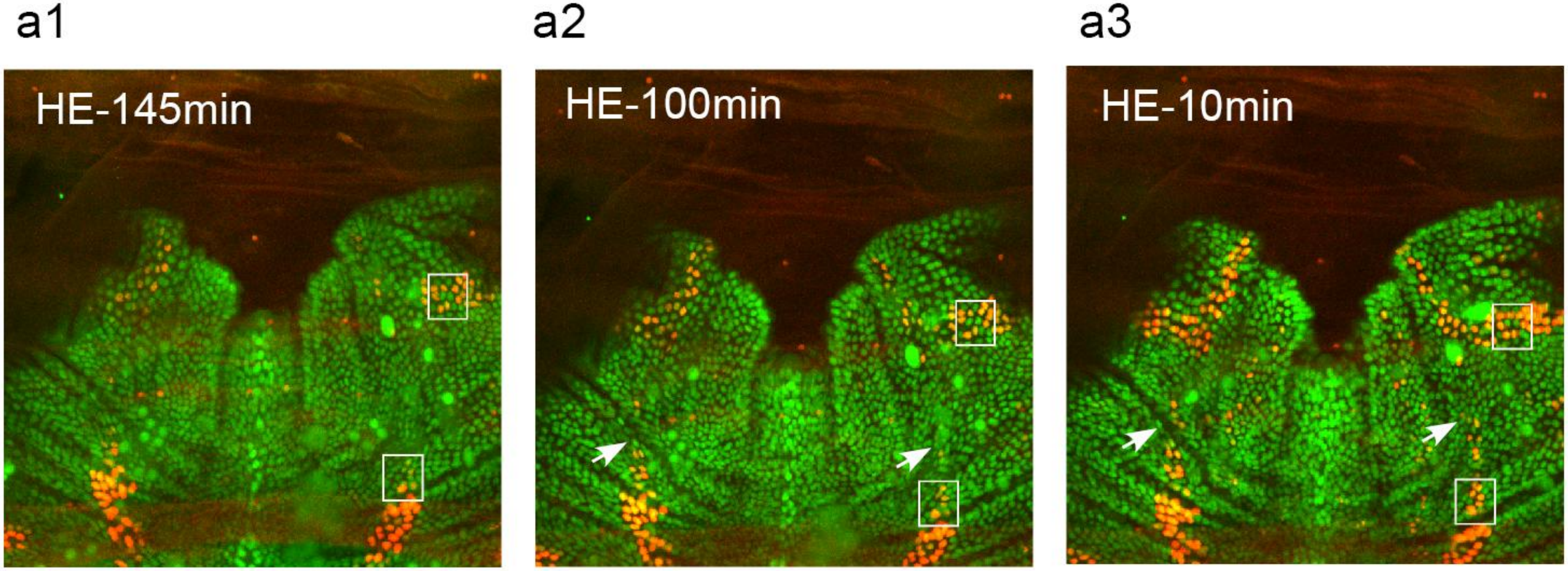
  - Reduces communication in computational systems, based on *cis* interactions for cells

Differences between SOP and MIS selection

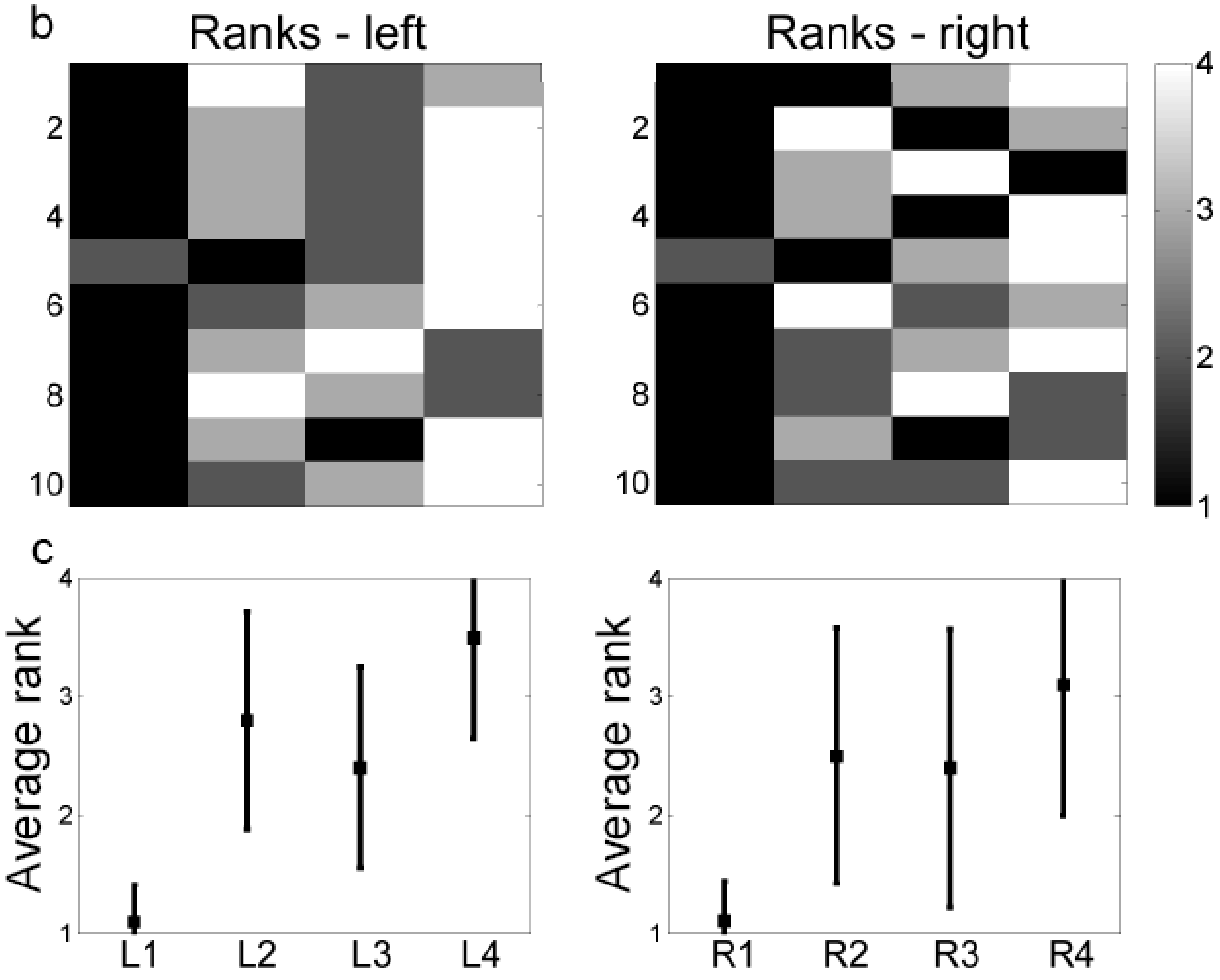
- In SOP selection cells do not know the status of their neighbors and the overall topology
- Messages in SOP selection are binary

Can we improve current algorithms for MIS by understating how the biological process is performed?

Movie



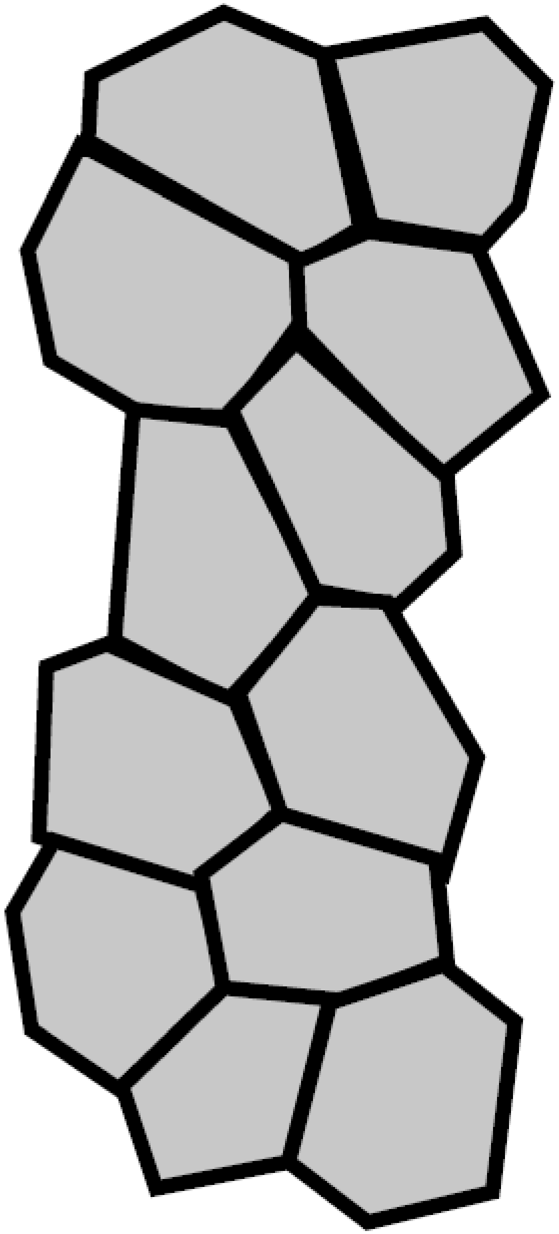
Observation 1: SOP selection is stochastic



Simulations

• 2 by 6 grid (also tried 2 by 7)

• Each cell touches all adjacent and diagonal neighbors



Simulations

- All models assume a cell becomes a SOP by accumulating the protein Delta until it passes some threshold

Four different models:

1. Accumulation

   - Accumulating Delta based on a Gaussian distribution

2. Fixed Accumulation

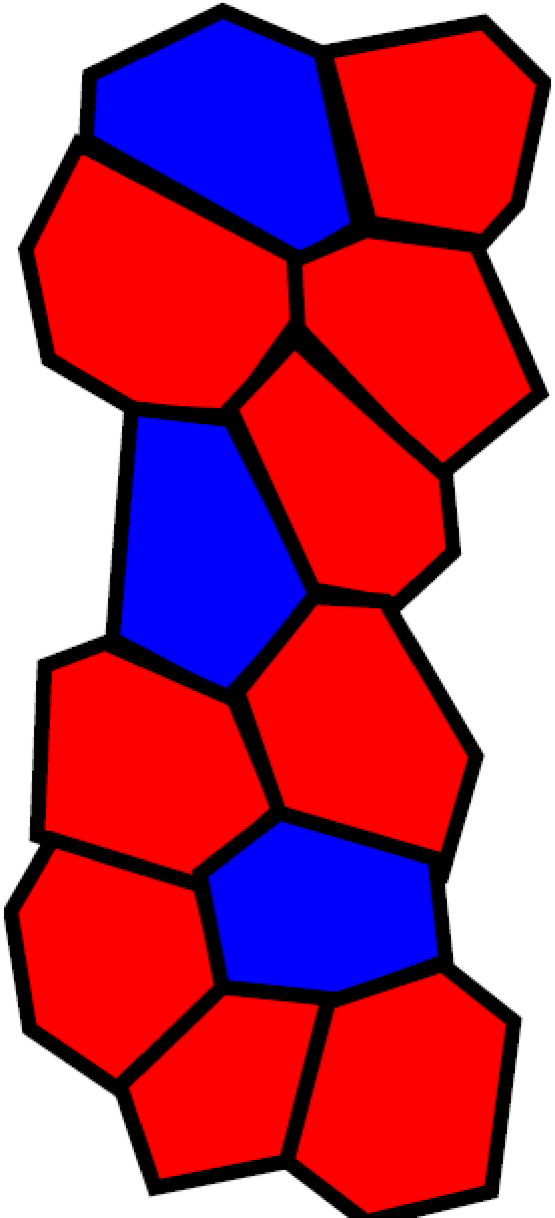
   - Randomly select an accumulation rate only once

3. Rate Change

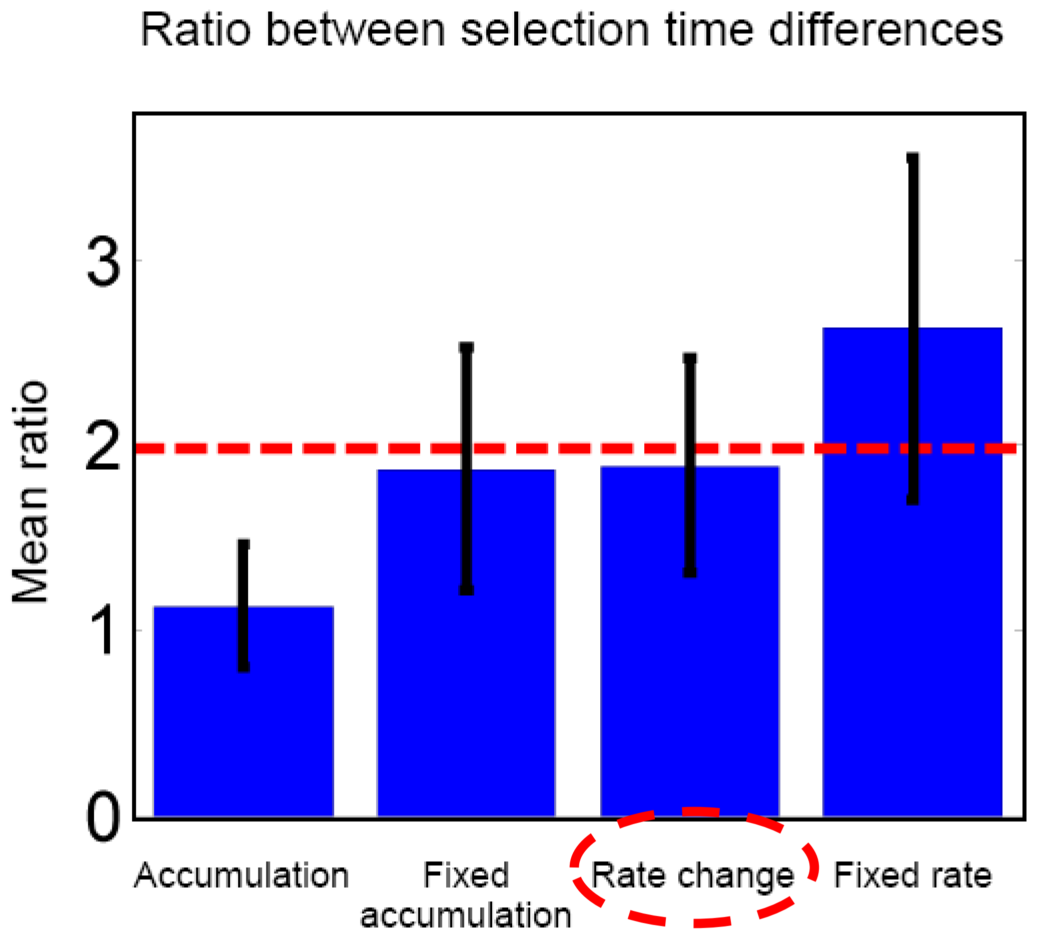
   - Increase accumulation probability as time goes by usind feedback loop

4. Fixed rate

   - Fix accumulation probability, use the same probability in all rounds



Observation 2: Comparing the time of experimental and simulated selection



MIS Algorithm (revised)

MIS Algorithm (n,D) // n – upper bound on number of nodes

D - upper bound on number of neighbors

- *p = 1/D*
- round = round +1
- if round > log(n)

  *p = p * 2* ; round = 0 // we start a new phase
- Each processor flips a coin with probability *p*
  - If result is 0, do nothing
  - If result is 1, send to all other processors
    - If no collisions, Leader; all processes exit
    - Otherwise

Afek et al *Science* 2011, Afek et al DISC 2011

Why does it work?

- Can show that by phase *i* there are no processes with more $n/2^i$ neighbors
- Overall running time is O(log (n) log(D)) where D is an upper bound on the number of neighbors
- For grids this is as fast as the best known algorithm for this problem.
- Message complexity is also extremely low: O(n)

Can be extended to:

1. Continuous probability increases
2. Unsynchronized settings
3. Cases where no upper bound is given on degree
4. No collision detection

(several) open problems

- Can we learn how to protect important nodes from the way cells rely on redundancy?
- Can we improve coordination among agents (or robots) based on bacterial quorum sensing?
- What other specific biological problems can gain from the information processing prescriptive, and how insights into these will improve computational algorithms?

Thanks

Distributed computing
Saket Navlakha
Tony Gitter

Alison Barth

CMU

Funding

- NIGMS
- NHLBI
- NSF

Naama Barkai,
Omer Barad

Weizmann

Yehuda Afek,
Noga Alon

Tel-Aviv Universtiy

www.algorithmsinnature.org