www.sciencemag.org/cgi/content/full/331/6014/183/DC1

Supporting Online Material for

# A Biological Solution to a Fundamental Distributed Computing Problem

Yehuda Afek, Noga Alon, Omer Barad, Eran Hornstein, Naama Barkai,*
Ziv Bar-Joseph*

*To whom correspondence should be addressed. E-mail: naama.barkai@weizmann.ac.il (N.B.); zivbj@cs.cmu.edu (Z.B.-J.)

Published 14 January 2010, *Science* **331**, 183 (2010)
DOI: 10.1126/science.1193210

**This PDF file includes:**



**Other Supporting Online Material for this manuscript includes the following:**
(available at www.sb.cs.cmu.edu/MIS/movie1.avi and www.sb.cs.cmu.edu/MIS/movie2.avi)

Movies S1 and S2

**Supporting Methods and Results for:**

# A biological solution to a fundamental distributed computing problem

**Yehuda Afek[1*], Noga Alon[1,2*], Omer Barad[3*], Eran Hornstein[3], Naama Barkai[3^], Ziv Bar-Joseph[4^]**

**1. Sackler School of Mathematics and Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv Israel**
**2. Institute for Advanced Study, Princeton, NJ, USA**
**3. Dept. of Molecular Genetics, Weizmann Institute of Science, Rehovot, Israel**
**4. School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA**

**Table of Contents** **Page**

**Background on the MIS problem**

Several textbooks on distributed computing provide a detailed discussion of the MIS problem, its history, importance and applications. See for example [1,2]. Maximal independent set (MIS) is a symmetry breaking problem extensively used in practice. For networks in which not all processes are directly connected to each other, MIS seeks to elect several local leaders which, together, cover the entire set of nodes in the network. The MIS solution is key for obtaining an initial structure and organization for sensor/radio networks. MIS is used to compute spanning trees in networks reducing communication costs. In addition, once a MIS is selected each node can select one of the MIS nodes adjacent to it leading to a clustering of the nodes which can then be used for routing and for solving other problems in wireless networks [3].

Back in 1982 Valiant mentioned MIS as a challenging problem for parallel and distributed computing stating that "it is difficult to see how this problem can be solved in substantially fewer stages such as $O(\sqrt{n})$ or $O(\log n)$" where $n$ is the number of nodes in the network [4]. As in the problem of electing a leader the problem of deterministically constructing a MIS is impossible in totally symmetric networks. Therefore, many probabilistic MIS algorithms were developed. Karp and Wigderson [5] presented a $O(\log^3 n)$ algorithm for parallel computers. This algorithm was later improved by Luby, and by Alon, Babai and Itai, who reduced the time to $O(\log n)$ [6,7].

The importance of algorithms that do not rely on knowledge of the number of neighbors and are only using limited communication is demonstrated by the following scenario: Assume we are dropping from an airplane thousands of monitoring devices for air quality control at some location and they need to communicate. These devices are expected to be operate for a long time and so need to conserve energy. They are also restricted in terms of their broadcast ability and can only send messages to other sensors that are at a close enough range. In this case we have thousands of nodes and the network is determined by the distance between pairs of nodes and the geographical topology (a distance below a cutoff implies an edge between them). However, since they were all dropped from an airplane they do not know how many neighbors they have and they cannot count as well (in radio networks, if two processes broadcast at the same time none of them gets the message so if everybody tries to broadcast to count neighbors nobody would get any message).

For such a case MIS is a useful procedure for obtaining an initial network structure and a set of local leaders that can communicate with all other processes to relay instructions and coordinate monitoring efforts. See [8] for additional discussion about the importance of these assumptions in practice.

**The relationship between MIS and SOP selection**

As discussed in the main text we see strong similarities and some notable differences between the MIS selection problem and SOP selection during peripheral nervous system development in flies. MIS attempts to select a subset of the nodes A such that:

1. All nodes are either in A or connected to a node in A.
2. No two nodes in A are connected.

These requirements allow other applications, that are based on the existence of MIS, to guarantee correct functioning of network operations [1,2]. These functions include routing where messages are sent via the MIS members, generating a spanning tree for the network and clustering of nodes in the network.
In SOP selection all cells either become SOPs or are physically inhibited by a SOP cell (and so are physically interacting with a SOP cell), similar to point 1 above. In addition, SOPs are required to be spaced such that no two SOP's are physically adjacent similar to point 2. Here the requirements stem from the fact that each SOP will eventually develop to a sensory organ, with an external bristle as the sensor. Inaccurate spacing of these bristles may distort the fly sense of the environment leading to an evolutionary disadvantage. Indeed, in our studies we found that for over 99% of SOPs the spacing requirement was maintained indicating a strong selective bias to maintain such spacing.

In addition to the above two requirements, when designing our computational model and algorithm we attempted to follow other properties of the biological systems as listed below:

1. Topology. It is unlikely that cells can distinguish neighboring cells that have been inhibited from neighboring cells that have not yet determined whether they become a SOP or not. This is especially true if most of the process involved in such a decision takes place within cells (cis mechanisms) as indicated by recent studies [9,10]. Thus, unlike previous computational methods [6,7], in our computational model we do not assume that nodes know the topology of the network and the subset of nodes that have exited the algorithm in each round.
2. Stochasticity. As discussed in the main text and as was demonstrated by our experimental results SOP selection is likely a stochastic process. Our computational model relies on the results of coin flips which provide stochasticity to the MIS election solution.
3. Communication. We assume a binary (1 bit) communication model. While the precise biological communication process is not completely resolved, several models that have been developed for this systems rely on a threshold decision and communication process which is effectively binary (yes / no depending on whether an internal threshold has been achieved) [10, 11,12].
4. Synchronous model. We assumed a synchronous model where communication between nodes proceeds in rounds (or phases in our algorithm). It is clear that the biological system is asynchronous. However, certain aspects of the biological system resemble synchronous models and are not generally assumed in asynchronous computational systems. These synchronous like characteristics include: 1. Coordinated start time (wakeup): Cells start the SOP selection process based on an external biochemical signal that reaches all cells at roughly the same time. This is hard to achieve in a completely asynchronous computational system. 2. Messages have a known delay: This is the major difference between synchronous and asynchronous computational models (the latter models do not assume a bound on message delay). In the biological system the messages involve a physical interaction using a biochemical process (Notch-Delta signaling). This process has a predictable delay which can be used to tune the internal selection

probabilities. Thus, the computational method we could derive from studying the biological system is only applicable to synchronous systems.

The major difference between the computational and biological system is in the number of neighbors each cell / node can have (the initial degree). In computational models we usually do not want to restrict the degree (and so it could be as high as all nodes in the network) whereas in the biological system the degree is constrained by the physical interaction requirement which limits the number of neighbors to roughly 6 per cell. Fortunately, the rate change model that fits the biological process (see also below for more details) can be extended to accommodate larger number of neighbors as we discuss in the paper.

## Supporting Methods

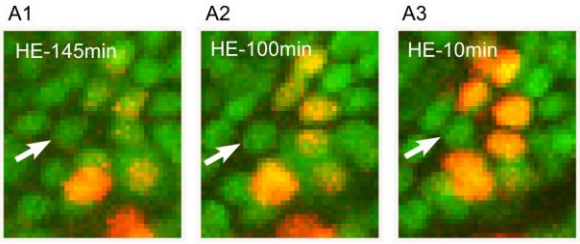
### Live imaging of SOPs selection in pupal discs

Intact pupae of E(spl)mα-DsRed.T4-NLS [13] ; FRT80B-GFP-NLS (Bloomington) strain were collected at ~8-10 hours after puparium formation. Then, pupae were submerged in halocarbon oil and were subjected to time-lapsed fluorescent microscopy in 2 minutes intervals under a Zeiss LSM710 system. Confocal images were z-stacked in 5 μm steps to make a vertical axis of ~100μm. Image analysis was performed with a Zeiss LSM image browser, depicting the maximal intensity pixel from a set of Z axis positions (Figure 2a and Supplementary movie).

### Live imaging data analysis

We collected live imaging movies of ten distinct pupae. Each movie follows SOP selection of the notum microchaetes [14-16] from ~9-12 hours after puparium formation. This time window enables us to follow selection events within the fifth microcheate rows on the left and right sides of the future notum (corresponding to *two* distinct imaginal discs allowing us to follow two SOP selection clusters per movie). The shape of the fifth row pro-neural cluster is an elongated cluster with a characteristic size of 2 by 10 cells. Typically, four SOPs are selected from each of the fifth rows. In one of the movies we looked at we identified 5 SOPs on one of the sides. The four first SOPs selected from that movie were used for the analysis.

A possible drawback of in-vivo imaging is that a live marker for the selected SOP cell is lacking. In fact, we identified the selected SOP as a non-repressed cell within the pro-neural cluster. This indirect detection method is based on the reliability of the reporter [13, 17]. All the cells in the neuroectoderm layer were marked by a ubiquitous nuclear GFP expression, and Notch activity was detected by E(spl)mα-DsRed reporter expression. Thus, we identified a selected SOP as a green cell surrounding by cells expressing the DsRed reporter (Supp. figure 1). Notably, Notch reporter is activated only within the pro-neural cluster [17]. Thus, due to the fifth row narrow shape, the selected SOP is usually not entirely enclosed by reporter expressing cells.




**Supporting Figure 1: Time-lapse imaging of SOP selection in live pupa. Insets from main text figure 2a. All cells are marks with GFP (green) and Notch activation is visualized by the mα-dsRed (red) reporter. SOP is identified by reporter up-regulation in adjacent cells.**

## Simulation studies

Determining the SOP selection rate experimentally is complicated since the precise initiation time of the process is hard to define. Further, the observed selection times need to be transformed in order to derive the underlying selection rate. We have therefore employed in-silico modeling to identify the qualitative properties that could distinguish between selection rates that match our experimental observations and those that do not. In particular, we considered three limiting cases: a constant rate of selection (leading to a temporal decrease in the observed selections, as in Supporting Figure 2), a constant rate of observed selection (corresponding to temporally increasing selection rate, Supporting Figure 3) and normally-distributed selection times (corresponding to a low-noise process with a well-defined time of selection, Supporting Figure 4). Each of these cases can be associated with one or more possible stochastic cellular accumulation models as we discuss below.

### Selection time statistics

As discussed in the paper we observed a stochastic behavior for the selection time of the SOPs selected second, third and fourth in each cluster. Let s2 be the selection time of the second SOP, s3 be the selection time for the third and s4 for the fourth (see Supporting Table 1, s2 corresponds to R2 and L2, s3 to R3 and L3 etc.). We first computed the average value of the differences between these selection times across all clusters by setting mean43 = mean(s4-s3) and mean32 = mean(s3-s2) where the average is over all 20 clusters we analyzed. Next we set $r = \frac{mean43}{mean32}$. In our experiments we found that *r=1.98*. We then simulated the performance of four different possible models leading to stochastic selection of SOPs to determine whether the observed ratio *r* matches the differences that would be obtained from the distributions used in each of these models. These distributions are described in detail below. In all the descriptions below we follow a rounds (synchronous) model. In all simulations, when a cell decides to become a SOP, all of its neighbors exit the process in the same round. In this model errors occur when two neighboring cells decide to become SOPs in the same round. The models we considered are:

**Constant rate:** Constant rate assumes a set of bursts of Delta production in each cell. If a specific burst exceeds the threshold the cell is selected as a SOP. Otherwise the Delta produced is degraded and the cell repeats this process. We simulate this using a threshold *p* for all cells. We assume that each cell draws a random number $v \sim U(0,1)$ at each round (representing the burst value). If $v >= p$ the cells becomes a SOP. Otherwise, it continues to the next round.

**Rate change:** This is based on the model described in the paper and later used in the MIS algorithm. We assume a burst model in which bursts of Delta production lead to enough Delta to pass the threshold in each cell. Bursts are stochastic events. Initially the probability that a burst occurs in each cell is *p*. This probability increases with time (reflecting, i.e. an increased activity of the proneural transcription factors Acheate and Scute). For these simulations we double the probability every $O(\log n)$ rounds as in the algorithm where *n* is the number of cells in the simulation. However, a similar result is

achieved if we continuously increase the rate rather than keeping it constant and than doubling it as long as the increase leads to doubling the rate every *log n* rounds.

**Accumulation:** In this model we assume that each cell accumulates Delta (without degradation) until the amount passes a certain threshold *t*. Unlike the rate change model, however, the distribution is much more centered (a Poisson distribution $v \sim Pois(1)$) and so larger thresholds are required to minimize errors. The amounts generated in each round are added to the total amount so far and when a cell reaches the threshold it becomes a SOP.

**Fixed accumulation:** This model is similar to the accumulation model in the sense that cells accumulate Delta and become SOP when they pass a threshold *t*. However, unlike that model the decision of how many units to accumulate in each round is predetermined. Specifically, at the beginning of the process cells chose a value $v \sim N(\mu,\sigma^2)$ and they accumulate in each round *v* units of Delta (see below for details about the values of $\mu$ and $\sigma^2$). We truncate *v* setting it to 0 if $v<0$. This distribution may represent cases in which cells differ slightly at the beginning of the process (for example, some are larger than others etc.).

**Parameter estimation**

For the first three distributions we need to determine either the fixed or initial probability *p* or the threshold parameter *t*. We choose parameters for these three models such that the percentage of errors (two neighboring cells becoming SOPs in the same round) is roughly 10%. The resulting parameters are presented in the legends of Supporting Figures 2-5. Reducing this percentage did not change the results reported in the paper but led to much longer simulation times since it required a large increase in the number of rounds performed by each cell. For the fourth model (fixed accumulation using a Gaussian) we need to set both *t* and the distribution parameters. However, we noted that when the variance is small compared to the mean the method behaves very similar to the constant rate method (regardless of the threshold used). Thus, we have set the standard deviation to be half the mean and varied *t* to achieve the 10% error rate mentioned above.
Supporting Figures 2-5 provide a graphical view for each of the distributions for each of these stochastic models. These plots are based on 100000 simulation runs in which a single cell is run until it becomes a SOP (without taking into account neighbors, i.e. first passage time).

**Topology for the simulation**

For the full simulation we need to select 3 SOPs in each cluster. We have used a 7 by 2 grid for this simulation (see Supporting Figure 6). All cells touching each other are neighbors including direct diagonal upper or lower cells (see Supporting Figure 6). We have used the time of selection (in terms of rounds) of the first three SOPs to compute the ratio for each distribution.

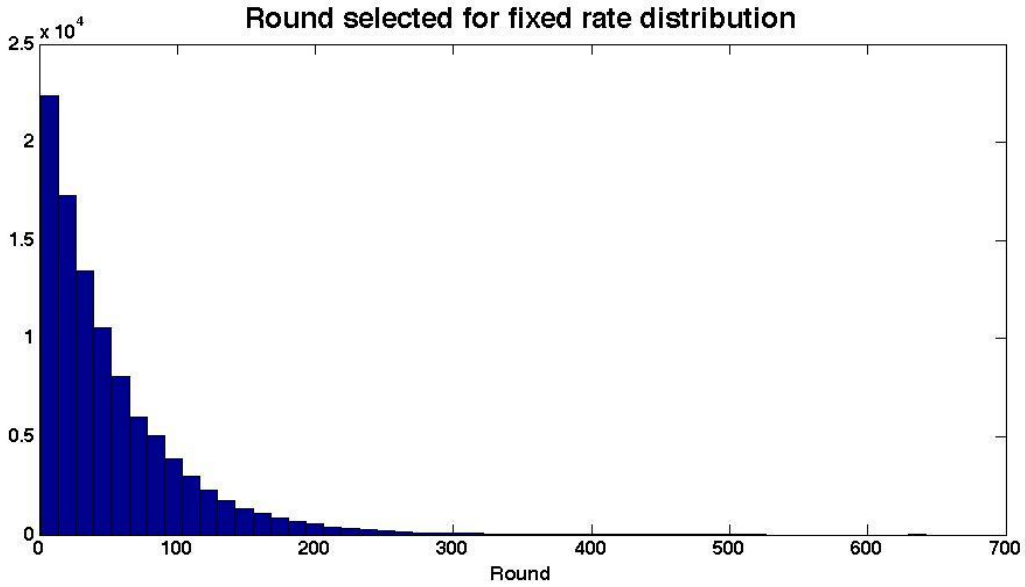
**Simulation time**

Each unit of simulation for each distribution involves 20 runs on the above grid (so that the results are comparable to the experimental results). Runs resulting in errors (two

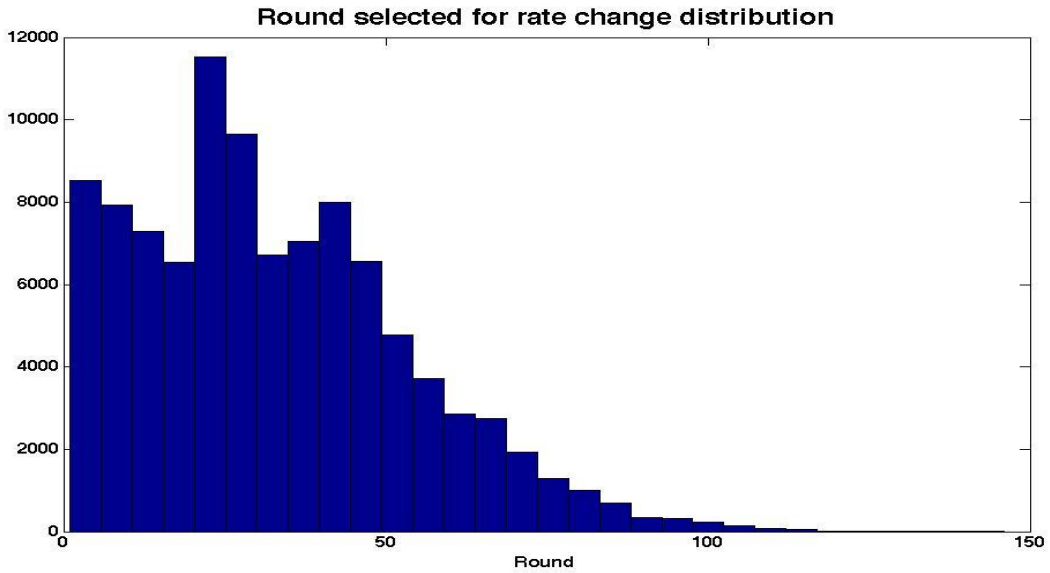
neighboring cells selected in the same round) were discarded from further analysis. To determine the expected ratio and its standard deviation we repeated each unit run 1000 times for each distribution. Thus, we performed 20000 runs for each model. The results of these simulations were used in Figure 3.

**Deriving a computational algorithm from based on the simulation results**

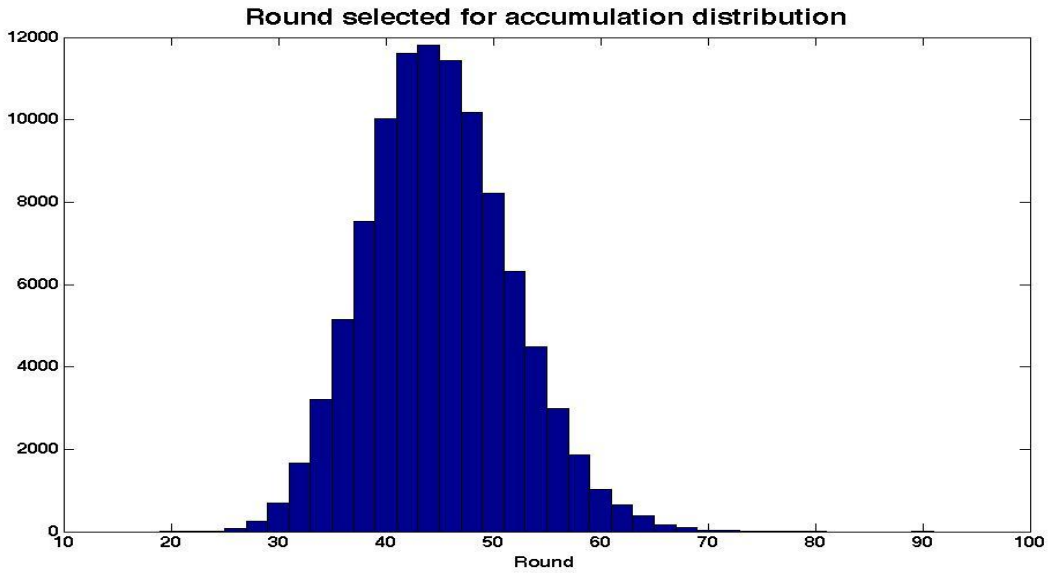
While both the rate change and fixed accumulation models agreed with our experimental observations, we have focused on the rate change model when deriving the algorithmic solution to the MIS problem. The reason we have focused on this model is that it is based on a similar idea as previous algorithms for MIS (for example, the algorithms of Luby [6] and Alon et al [7] that are mentioned in the paper) in which nodes change their probability during the execution of the algorithm. However, while in these algorithms the new probabilities depend on the changes in topology and require knowledge of the degree in each round, in the biological solution no such assumption is used. We have thus asked whether the rate change process used in biology can be applied to the computational problem which led to the algorithm we presented. As for the fixed accumulation model, while this model works well in the fly system, it is heavily dependent on the fact that each cell is only connected to a small number of other cells. When looking for a solution to the computational problem we have tried to identify a solution that would generalize well to much larger networks. The rate change model indeed follows this requirement as discussed in the text while the fixed accumulation model does not and so we have not pursued this model further.



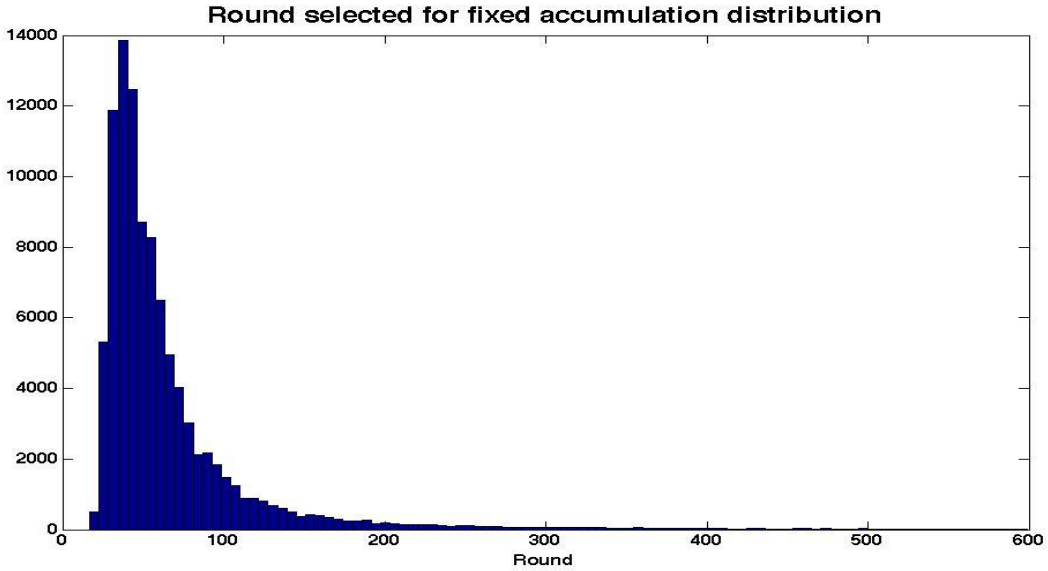

**Supporting Figure 2: Distribution of first passage time for the round selected using the fixed rate distribution. The probability that was used for each coin flip in each round was 0.019. As discussed above this value leads to two neighbors selected in the same round in 10% of the runs.**




**Supporting Figure 3. Distribution of first passage time for the round selected using the rate change distribution. The initial probability that was used for coin flips was 0.018. Interestingly, while the error rate for this distribution is the same as the fixed rate above, the median round selection time was much earlier (30 vs. 36) highlighting the advantages of rate change for the MIS problem as observed by prior computational methods.**

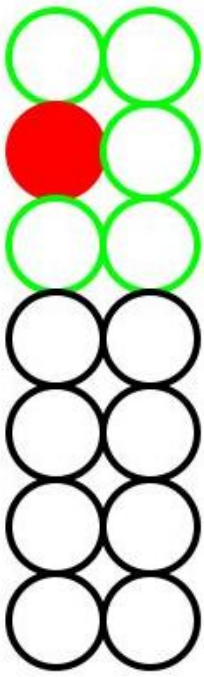



**Supporting Figure 4: Distribution of first passage time for the round selected using the accumulation distribution. The threshold was set for 45. The resulting distribution resembles a Gaussian as expected.**




**Supporting Figure 5: Distribution of first passage time for the round selected using the fixed accumulation distribution. Note that it rises much more rapidly than the regular accumulation distribution which explains the difference in the ratios between the two. However, this distribution can also lead to very large selection times. Roughly 1% of the selections occur after 600 or more rounds (not shown, the results above are truncated at 600 for presentation issues) with some cases leading to selections after 10000 rounds.**

**Supporting Figure 6: Grid used for simulation and neighbor definition. The figure shows the 7 by 2 grid used for simulation. The red node has joined A and all its neighbors (green circles) have exited the algorithm. All other nodes (black circles) continue to the next round until they or one of their neighbors join A.**



## Supporting Results

### SOP selection order

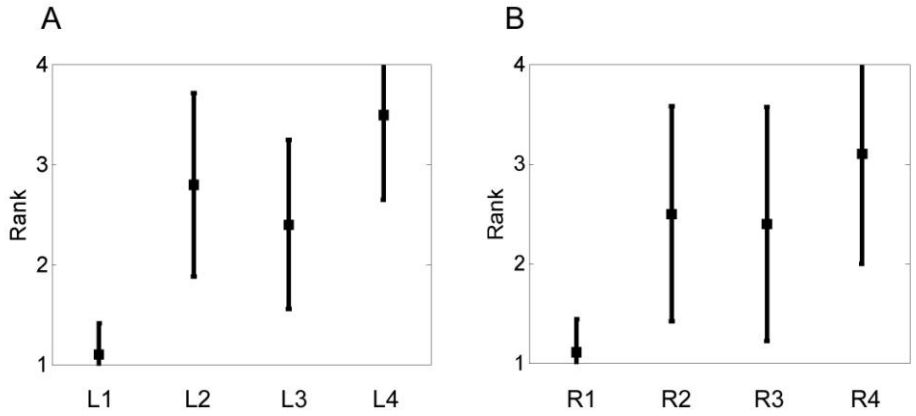
We manually extract the estimated SOP selection times of twenty clusters, two clusters from each distinct pupae (Supp. table 1). The selected SOPs were mapped between the clusters based on their locations order within each cluster. We define the most posterior SOP as L1 (or R1) and the most anterior SOP as L4 (or R4). The allocation of specific selection time for each SOP was based on the earliest time step of Notch activity detection in one of its neighbors. Note that the absolute time values are dependent on the exact age of each pupa at the beginning of the live imaging follow up. Therefore, to assess the stochasticity of this selection process we first used the selection rank of the SOP within each cluster rather than the actual timing information.

Supp. figure 7 presents the average ranks for each SOP position. Clearly, the selection of the first SOP is biased toward L1 (or R1). To formally evaluate the statistical significance of this bias, we compared the measured rank of the selected SOPs to a random SOP selection model. Random selection was simulated as follows. Each cell in a cluster of 2X10 cells is assigned a selection time from a uniform distribution (the results are independent on the specific distribution shape). Then, cluster cells are marked as SOPs or non-SOPs. Iteratively, an un-marked cell with the earlier selection time is marked as an SOP and its nearest neighbor cells are marked as non-SOPs. When the status of all cells is determined, we extract the rank of SOP selection based on their order from top (anterior) to bottom (posterior).

Our null hypothesis is that SOP selection is random and we tested whether the measured result can reject this hypothesis. To test this, we compared the average correlation of SOPs ranks between each pair in the 20 measured selection process to the corresponding average correlation of the simulation ranks (with similar 20 repeats in each of the 10,000 simulations). P-value of the experimental ranks was obtained from the random model statistics. The measured result was below any incidence from the 10,000 simulation results (P-val $<10^{-4}$), verifying that the posterior SOP tends to be selected first.

As we can see from Supp figure 7, additional biases in the selection order of the remaining SOPs, if exist, are much less significant. Comparison of the L2-L4 (and R2-R4) selection order to a random model (with a cluster of 2X7 cells) following the same procedure as described above yields a non significant P-value of 0.06. Moreover, 5 out of the 6 possible selection orders on 3 SOPs were detected in our limited measured samples. Thus, we conclude that stochasticity has a major role in SOP selection within the microchaetes fifth row proneural cluster.

Next, we validated that there is no correlation in the selection order of the left and right clusters within the same pupa. We found that the average selection order correlation between L2-L4 and R2-R4 in the same pupa (~0.15) is below the average correlation between L2-L4 and R2-R4 of two distinct pupae (~0.22) again indicating a stochastic process.




**Supporting Figure 7: Measured microchaetes SOP selection ranks. Average and standard deviation of SOP selection ranks. (A) L1-L4: left sides SOPs in a posterior to anterior order, and (B) R1-R4: right sides SOPs in a posterior to anterior order selection ranks.**

| #Movie | L1 | L2 | L3 | L4 | R1 | R2 | R3 | R4 |
|---|---|---|---|---|---|---|---|---|
| 1 | 90 | 150 | 100 | 125 | 85 | 85 | 145 | 150 |
| 2 | 0 | 61 | 50 | 81 | 0 | 50 | 0 | 22 |
| 3 | 0 | 44 | 28 | 60 | 0 | 16 | 38 | 0 |
| 4 | 41 | 92 | 82 | 112 | 61 | 88 | 61 | 147 |
| 5 | 48 | 0 | 48 | 68 | 12 | 0 | 28 | 78 |
| 6 | 82 | 84 | 88 | 110 | 72 | 104 | 90 | 98 |
| 7 | 0 | 21 | 107 | 4 | 0 | 49 | 77 | 129 |
| 8 | 2 | 146 | 47 | 13 | 30 | 49 | 127 | 49 |
| 9 | 34 | 57 | 34 | 92 | ND | 71 | 34 | 55 |
| 10 | 0 | 2 | 33 | 63 | 0 | 33 | 33 | 44 |

**Supporting Table 1: Measured microchaetes SOP selection times. L1-L4 and R1-R4 are left and right sides SOPs in a posterior to anterior order. SOPs with a zero selection time were already detected in the first time step of the live imaging movie. ND stands for a not detected SOP.**

## Supporting movies

The two supporting movies depict, in each frame, the maximal intensity pixel from the set of Z axis positions that were tracked for each time point ("maximal intensity projection"). The movies provides a clock presenting the relative time of each frame (the experiment initiation time was set to zero for all movies) and a scale bar of 25μm.
We have manually annotated the two supporting movies (Sample Movie 1 and Sample Movie 2 corresponding to pupa #7 and #5 in Supporting Table 1, respectively). These movies provide annotations displaying the approximate region of the fifth row pro-neural clusters (surrounded by white rectangle in the first few frames of each of these movies), arrows referring to selected SOPs within the regions of interest, and an annotation for pupa head eversion initiation time.
In many cases the information in the maximal intensity projection movies was not accurate enough to determine the selection times of all SOPs since the neuroectoderm layer is placed with an incline (on the anterior-posterior axes) with respect to the

microscope focal plane, such that a stripe of the neuroectoderm layer with ~5 cells width is in focus at each z-section layer. In such cases the data of the neuroectoderm layer in the maximal intensity projection movies is interfered with irrelevant fluorescence signals coming from cells expressing GFP/DsRed reporter above and below the neuroectoderm layer, and from auto-fluorescence of the pupa cuticle. Thus, the relative SOP selections times that are listed in Supporting Table 1 were manually determined from the raw (Z stack) data according to the time step in which the level of DsRed reporter in the first neighbor of the selected SOP reaches a threshold.

**Lemmas for the correctness of the algorithm**

**Safety properties**

We first prove a few safety lemmas. These show that when the algorithm stops, no two neighboring nodes are in A (the final set produced by the algorithm) and that every node that has become inactive is a neighbor of a node in A.

**Lemma 1:** No two nodes in A are connected to each other.
**Proof:** Node *w* joins set A if it broadcast at an exchange of type 1 and none of its neighbors broadcasts at the same exchange. This means that none of its neighbors has joined A in a previous step since when a node joins A all its neighbors are marked inactive in the exchange in which it joins A. Thus two neighboring nodes cannot join A at the same exchange, and when the earlier of them joins, the other one is marked inactive and will never join A. ■

**Lemma 2:** If node *w* exits the algorithm and it did not join A, then it is connected to a node in A.
**Proof:** This can only happen if node *w* executes the "then" part in Line 12 of the code. This occurs only upon receiving a message from a neighboring node that has just joined *A.* ■

**Lemma 3:** If *w* is in A then all the neighbors of *w* become inactive.
**Proof:** By lemma 1 we know that none of *w*'s neighbors is in *A*. By the code (exchange 2) when *w* joins *A* it broadcasts to all its neighbors that then become inactive and exit the algorithm (Line 12). ■

**Run time proof**

We define the degree of a node to be the number of active neighbors this node has plus 1 (that is, we assume that all nodes are connected to themselves). Note that the degree may change during the execution of the algorithm as nodes exit the algorithm. We prove that by the time the algorithm ends all nodes have exited the algorithm with high probability.

**Lemma 4:** With probability at least $(1-\frac{i}{n^2})$ there are no nodes with degree > $\frac{D}{2^i}$ at the end of phase *i*.
**Proof:** By induction on *i.*

Base: for i=0 this is trivial (since D is an upper bound on the number of neighbors).

Induction step: Assume correctness for *i-1*. Based on the induction hypothesis, with probability at least $(1-\frac{i-1}{n^2})$ by the end of phase *i-1* there are no nodes with more than $\frac{2D}{2^i}$ active neighbors. If there are also no nodes with more than $\frac{D}{2^i}$ neighbors we are

done. Otherwise, let *v* be a node with more than $\frac{D}{2^i}$ neighbors. At each step in phase *i* (at least as long as it has more than $\frac{D}{2^i}$ neighbors in that phase) there is a probability greater than (*1-1/e*) that *v* or one of its neighbors broadcasts in line 6. To see this note that *v* and all its neighbors are flipping coins with probability $\frac{1}{2^{\log D-i}} = \frac{1}{(D/2^i)}$ and thus the probability that at least one of them would broadcast is:

$$p(v \quad or \quad neighbor\ of \quad v \quad broadcasts) \quad \geq \quad 1-(1-\frac{1}{D/2^i})^{D/2^i} \quad \cong \quad 1-1/e$$

On the other hand, based on the induction hypothesis no node has more than $\frac{2D}{2^i}$ neighbors. Thus, the probability that a node that broadcasts a message does not collide with any other node is:

$$\begin{aligned} p(no \qquad collisions) \quad &\geq \quad (1-\frac{1}{D/2^i})^{2D/2^i} \\ &\cong \qquad 1/e^2 \end{aligned}$$

Thus, in every step of phase *i*, node *v* has probability of at least $(1-\frac{1}{e})\frac{1}{e^2} \geq \frac{1}{2^4}$ to be removed. Since there are *Mlog n* steps in this phase with probability $\geq 1-(1-\frac{1}{2^4})^{M\log n}$ *v* would be removed in this phase. Setting *M*=34 leads to:

$$1-(1-\frac{1}{2^4})^{M\log n} = 1-(1-\frac{1}{2^4})^{34\log n} \cong 1-(\frac{1}{e})^{2.1\log n} > 1-(\frac{1}{e})^{3\ln n} = 1-\frac{1}{n^3}$$

Since there are at most *n* nodes with more than $\frac{D}{2^i}$ neighbors, if the induction hypothesis holds then the probability that all nodes with this degree would be removed at the end of this phase is $\geq 1-\frac{1}{n^2}$. Combining this with the induction probability, the probability that there is no node with a degree greater than $\frac{D}{2^i}$ at the end of round *i* is $(1-\frac{i-1}{n^2})(1-\frac{1}{n^2}) \geq 1-\frac{i}{n^2}$. The last product is based on the chain rule which states that *p(a,b) = p(a)p(b|a)* and replacing *a* with the induction hypothesis and *b* with the event that there would not be a node with a degree greater than $\frac{D}{2^i}$ at the end of round *i.* ■

Lemma 4 leads to the following theorem regarding the probability that by the end of the algorithm it has found a MIS:

**Theorem 1:** With probability $1-\frac{\log D}{n^2} \geq 1-\frac{\log n}{n^2}$ all nodes are either in A or connected to a node in A by the end of the algorithm.

**Proof:** From lemma 4, with probability $1-\frac{\log D}{n^2}$ all nodes that are left in the algorithm at the end of phase log D have a degree of 1 (meaning that they are not connected to any other active node). These will all insert themselves into A. Since they are still active lemma 3 guarantees that they do not have any neighbors in A and so with this probability by the end of the algorithm all nodes are either in A or connected to a node in A. ■

Note that while we ended up with a probability of $1-\frac{\log D}{n^2}$ for reaching a MIS when the algorithm terminates, we can obtain a probability of (1-ε) for arbitrary values of ε by changing the value of the parameter *M*. For example, doubling *M* would lead to an error probability at most $\frac{\log D}{n^5}$ for Theorem 1 instead of $n^2$ in the current denominator. In general, a linear increase in *M* leads to an exponential decrease in the probability of an error.

**Message complexity**

In this section we show that the expected message complexity of our algorithm is linear in the number of nodes in the graph. Note that in the analysis we count a message only if it is sent from an active node to an active node. As explained at the end of the section it is not difficult to see that if we charge for messages sent to inactive nodes as well, the expected number of messages can be quadratic.

Let B denote the event that there are no nodes with degree > $\frac{D}{2^i}$ at the end of phase *i* for *all* values of *i*. The expected number of messages received in our algorithm, E(#M), can be written us:

$$E(\#M) = P(B)*E(\#M|B) + P(\neg B)*E(\#M|\neg B)$$

where E(#M|B) is the expected number of messages sent when B holds and E(#M|¬B) is the expected number of messages sent when B does not hold.

To derive the value of E(#M|¬B) note that the total number of messages that can be received during the execution of the algorithm is $n^2 log^2 n$ (in each subphase a node can receive at most *n* messages, there are *n* nodes and at most $log^2 n$ subphases). As we proved (Theorem 1), $P(\neg B) \leq \frac{\log n}{n^2}$. Thus:

$P(\neg B)*E(\#M|\neg B) \leq \frac{\log n}{n^2} n^2 \log^2 n = log^3 n \leq 10n.$

We now derive the value of E(#M|B). First, note that for any two events A and C in probability space where P(C) = 1-ε we have: $P(A \mid C) = \frac{P(A,C)}{P(C)} \geq P(A,C) \geq P(A) - \varepsilon$.

On the other hand $P(A \mid C) = \frac{P(A,C)}{P(C)} \leq \frac{P(A)}{P(C)} \leq P(A) + 2\varepsilon$ . Combined we have $P(A) - \varepsilon \leq P(A \mid C) \leq P(A) + 2\varepsilon$. Thus for every event A of probability significantly bigger than epsilon, P(A) and P(A | C) are essentially the same.

Since P(B) $\geq 1 - \frac{\log n}{n^2}$ this reasoning shows that when conditioning on B we may use the probabilities without referring to the condition, as it holds with probability close enough to 1 that ensures the computation of the expectation is accurate enough while assuming the condition.

Call the part of the MIS Algorithm (Table 1 in the main text) corresponding to a specific phase *i* ($0 \leq i \leq log\ D$) and step *j* ($0 \leq j \leq log\ n$) subphase *(i,j)*. For a node *u* and a subphase *(i,j)*, define two random variables $X_{i,j}^{(u)}$ and $Y_{i,j}^{(u)}$ as follows.

If *u* is inactive already at the beginning of subphase *(i,j)*, then $X_{i,j}^{(u)}$=0. Otherwise, $X_{i,j}^{(u)}$ is the number of messages sent by *u* to its active neighbors during the subphase.

If *u* is active at the beginning of subphase *(i,j)* and becomes inactive by the end of the subphase, then $Y_{i,j}^{(u)}$=1. Otherwise, $Y_{i,j}^{(u)}$=0.

We proceed to estimate the expectation of the above two random variables. Suppose *u* is still active in the beginning of subphase *(i,j)*, and let *d* denote its degree at the beginning of this subphase (that is, the number of its active neighbors at the beginning of the subphase). The probability that in line 6 of the algorithm the marker *v* of *u* gets the value 1 is $\frac{1}{2^{\log D - i}}$. When *u*'s marker *v* gets the value of 1, *u* has to broadcast at most *2d* messages to its active neighbors (*d* in line 6, and either *d* or 0 in line 10). Otherwise, *u* does not send any messages. Thus, the expected value of $X_{i,j}^{(u)}$ in this case is at most $\frac{2d}{2^{\log D - i}}$. (If *u* is inactive at the beginning of the subphase then the expectation of $X_{i,j}^{(u)}$ is clearly zero.)

Turning to the expectation of $Y_{i,j}^{(u)}$ note, first, that here, too, if $u$ is inactive at the beginning of the subphase then the expectation of $Y_{i,j}^{(u)}$ is zero. Assume, thus, that $u$ is still active in the beginning of the subphase, and recall that we may assume that its degree $d$ in the beginning of the subphase is at most $\frac{2D}{2^i}$. The probability that $Y_{i,j}^{(u)}=1$ is at least the probability that exactly one of the $d$ active neighbors of $u$ gets its marker $v$ set to 1 in line 6, and none of the markers of its active neighbors is set to 1. By the arguments in the previous proofs, this probability is at least:

$$d\frac{1}{2^{\log D-i}}(1-\frac{1}{2^{\log D-i}})^{d-1}\frac{1}{e^2}\geq\frac{d}{2^{\log D-i}}\frac{1}{e^4}$$

We conclude that $\mathrm{E}(X_{i,j}^{(u)}) \leq 2e^4\ \mathrm{E}(Y_{i,j}^{(u)})$. Summing over all values of $i$ and $j$, using linearity of expectations we get $E(\sum_{i,j}X_{i,j}^{(u)})\leq 2\mathrm{e}^4\ E(\sum_{i,j}Y_{i,j}^{(u)})\leq 2\mathrm{e}^4\cdot 1$ since $\sum_{i,j}Y_{i,j}^{(u)}\leq 1$, as $u$ can become inactive only once during the whole algorithm. We thus conclude that the expectation of the total number of messages sent by $u$ to active neighbors during the algorithm when B holds is at most $2e^4$.
Combining the parts of the sum we have:
$\mathrm{E}(\#\mathrm{M}) = \mathrm{P}(\mathrm{B})*\mathrm{E}(\#\mathrm{M}|\mathrm{B}) + \mathrm{P}(\neg\mathrm{B})*\mathrm{E}(\#\mathrm{M}|\neg\mathrm{B})$ $\mathrm{E}(\#\mathrm{M}) \leq 2e^4n+10n$

proving our claim.

**Remark**: If we charge for messages sent to inactive nodes as well, then the expected message complexity may well be quadratic in $n$ as shown by the following example. Let $G$ be the graph consisting of a matching on a set $U$ of $0.9n$ nodes and a complete graph on a set $W$ of $0.1n$ nodes, where each node of $U$ is adjacent to each node of $W$. The maximum degree here is $D=n$, and with probability $\Omega(1)$ in the very first subphase only one node of $U$ joins the independent set, while his mate in the matching as well as all nodes of $W$ become inactive. If this happens, then in the rest of the algorithm, whenever a node of $U$ joins the independent set, it sends more than $0.2n$ messages, and as altogether $0.45n$ nodes of $U$ must join the independent set, the total number of messages sent in this case is $\Omega(n^2)$. Of course, most of these messages are being sent to the inactive nodes of $W$, and are thus not charged in our analysis above. Note also that since all messages are broadcast the proof also shows that the total number of broadcasts performed (in line 6) is $O(n)$. This is the optimal value for a general MIS broadcast algorithm for arbitrary networks since in complete bipartite networks with vertex classes of equal size, for example, half the nodes must send at least one message.

**A note on maximal network degree**

The biological process that motivated our work has a bounded degree with each cell surrounded by a few (up to 6) other cells. For such networks the run time of our algorithm is O($\log n$), similar to the best known algorithms for MIS even though our

algorithm relies on fewer assumptions. Note that for bounded degree networks, a fixed probability distribution would also lead to an O(*log n*) run. However, this run time analysis is asymptotic. In the simulations we performed we observed that for the biological topology we used a rate change method leads to faster selections when compared to a fixed rate method. In addition, the rate change model seems to lead to better agreement with the experimental data when compared to the fixed rate model (Figure 3 in the main text).

## References


[1] N. Lynch. Distributed Algorithms. 71-74. Morgan Kaufmann, San Francisco, CA, USA (1996).

[2] D. Peleg. Distributed Computing: A Locality-Sensitive Approach. Chapter 8. Society for Industrial and Applied Mathematics (SIAM), 2000.

[3] P.J. Wan, K.M. Alzoubi, O. Frieder, Distributed construction of connected dominating Sset in wireless ad hoc networks. *Mobile Networks and Applications* **9**(2), 141-149 (2004).

[4] L.G. Valiant. Parallel computation. *Proceedings of the 7th IBM Symposium on Mathematical Foundations of Computer Science*. IBM, New York, 1982.

[5] R.M. Karp and A. Wigderson. A fast parallel algorithm for the maximal independent set problem. *JACM,* **32:** 4, 762 – 773 (1985).

[6] M. Luby. A Simple Parallel Algorithm for the Maximal Independent Set Problem. *SIAM J. Comput.* **15**(4): 1036-1053 (1986).

[7] N. Alon, L. Babai, and A. Itai. A fast and simple randomized parallel algorithm for the Maximal Independent Set Problem. *Journal of Algorithms*, **7** 567-583 (1986).

[8] F. Kuhn, T. Moscibroda, and R. Wattenhofer. Initializing Newly Deployed Ad Hoc and Sensor Networks. *Proceedings of 10 th Annual International Conference on Mobile Computing and Networking (MOBICOM)*, 260-274, 2004.

[9] O. Sprinzak, *et al*. Cis-interactions between Notch and Delta generate mutually exclusive signalling states. *Nature* **465**, 86-90 (2010).

[10] O. Barad, D. Rosin, E. Hornstein, N. Barkai. Error minimization in the Notch-Delta lateral inhibition system. *Science Signaling*. **3**(129):ra51 (2010).

[11] J. R. Collier, N. A. Monk, P. K. Maini, J. H. Lewis, Pattern formation by lateral inhibition with feedback: A mathematical model of Delta-Notch intercellular signalling. *J. Theor. Biol.* **183**, 429–446 (1996).

[12] E. Plahte, Pattern formation in discrete cell lattices. *J. Math. Biol*. **43**, 411–445 (2001).

[13] Barolo, S., B. Castro, and J.W. Posakony, *New Drosophila transgenic reporters: insulated P-element vectors expressing fast-maturing RFP*. Biotechniques, 2004. **36**(3): p. 436-40, 442.

[14] Usui, K. and K.-i. Kimura, *Sequential emergence of the evenly spaced microchaetes on the notum of Drosophila.* Roux's Arch Dev Biol, 1993. **203**: p. 151-158.

[15] Parks, A.L., S.S. Huppert, and M.A. Muskavitch, *The dynamics of neurogenic signalling underlying bristle development in Drosophila melanogaster.* Mech Dev, 1997. **63**(1): p. 61-74.

[16] Renaud, O. and P. Simpson, *Movement of bristle precursors contributes to the spacing pattern in Drosophila.* Mech Dev, 2002. **119**(2): p. 201-11.

[17] Castro, B., et al., *Lateral inhibition in proneural clusters: cis-regulatory logic and default repression by Suppressor of Hairless.* Development, 2005. **132**(15): p. 3333-44.

Supporting movies for:

# A biological solution to a fundamental distributed computing problem

## Live imaging experiments

Intact pupae of E(spl)mα-DsRed.T4-NLS ; FRT80B-GFP-NLS strain were collected at ~8-10 hours after puparium formation. Then, pupae were submerged in halocarbon oil and were subjected to time-lapsed fluorescent microscopy in ~2 minutes intervals under a Zeiss LSM710 system. Confocal images were z-stacked in 5 μm steps to make a vertical axis of ~50-100μm (resulting in between 10 and 20 images per time point).

We followed SOP selection in the neuroectoderm layer from ~9-12 hours after puparium formation. This time window enables us to follow selection events within the fifth rows of the small bristle on the left and right sides of the future adult fly notum (allowing us to follow SOP selection in two distinct clusters per pupa).

All the cells in the neuroectoderm layer were marked by a ubiquitous nuclear GFP expression, and Notch activity was detected by nuclear E(spl)mα-DsRed reporter expression. The trace for a SOP selection event is characterized by DsRed reporter accumulation in the selected SOP neighboring cells. Once it inhibits all surrounding cells a SOP can be detected as a green (GFP expressing) cell surrounded by red (DsRed expressing) cells. The selected SOP is usually not entirely enclosed by reporter expressing cells since Notch reporter is activated only within the pro-neural cluster which in the case of the fifth row has a narrow shape.

## Movies

This movies provided in this DVD depict, in each frame, the maximal intensity pixel from the set of Z axis positions that were tracked for each time point ("maximal intensity projection"). Two movies are presented for each pupa (except for pupa #1): one with merged green and red channels and the second with only the red channel (the red channel movies contain "RFP" as a suffix). Movies are numbered as in Supplementary Table 1. Each movie provides a clock presenting the relative time of each frame (the experiment initiation time was set to zero for all movies) and a scale bar of 25μm.

We have manually annotated two of the movies (Sample Movie 1 and Sample Movie 2 corresponding to pupa #7 and #5, respectively). These movies provide annotations displaying the approximate region of the fifth row pro-neural clusters (surrounded by white rectangle in the first few frames of each of these movies), arrows referring to selected SOPs within the regions of interest, and an annotation for pupa head eversion initiation time.

## Data analysis and raw data files

In many cases the information in the maximal intensity projection movies (as presented in this DVD) was not accurate enough to determine the selection times of all SOPs since the neuroectoderm layer is placed with an incline (on the anterior-posterior axes) with respect to the microscope focal plane, such that a stripe of the neuroectoderm layer with ~5 cells width is in focus at each z-section layer. In such cases the data of the neuroectoderm layer in the maximal intensity projection movies is interfered with irrelevant fluorescence signals coming from cells expressing GFP/DsRed reporter above and below the neuroectoderm layer, and from auto-fluorescence of the pupa cuticle. Thus, the relative SOP selections times were manually determined from the raw (Z stack) data according to the time step in which the level of DsRed reporter in the first neighbor of the selected SOP reaches a threshold.