

Practical 1 Morphogen Gradients and Patterning

In this practical, we are going to look at how an organism can form segments along its body axis. The mathematical model that we will implement and study is an implementation of the so-called French flag conceptual model first proposed by Lewis Wolpert [1]. It assumes the spatially graded expression of a morphogen “M” that influences the expression of some downstream genes A, B and C. Their expression is often visualized by red, white and blue and the arising pattern resembles the French flag, hence the name (see the power of visualization).

One of the most well-studied organisms when it comes to body axis segmentation (although its segmentation mechanism is evolutionary derived and atypical!) is the development of the fruit fly *Drosophila melanogaster*. Supporting the ideas of Wolpert, it was found that through tethering maternal Bicoid mRNA to one side of the embryo, Bicoid protein can form a gradient extending along the anterior-posterior axis, with so-called gap genes as a first tier in the segmentation hierarchy differentially responding to different Bicoid protein levels [2]. Later it was found that often at least two opposing morphogen gradients drive downstream gene expression and genes typically respond to multiple inputs.

Mathematical modeling – integrating multiple signals

Promoters/enhancers driving gene expression frequently make use of so-called OR and AND gates to integrate inputs from different transcription factors. An OR gate can be implemented mathematically with a sum of the effect of the transcription factors, while an AND can be implemented mathematically with a product. Some examples:

- $dX/dt = a(tf1) + b(tf2)$ means gene X is induced by transcription factor 1 and 2 in an OR fashion, it is sufficient for either $a(tf1)$ or $b(tf2)$ to be high to give high transcription of X.
- $dY/dt = a(tf1)b(tf2)$ means gene Y is induced by transcription factor 1 and 2 in an AND fashion, both $a(tf1)$ and $b(tf2)$ need to be high to give high transcription of X.

Note that the shape of functions $a(tf1)$ and $b(tf2)$ (increasing or decreasing function of the transcription factor) determines whether $tr1$ and $tr2$ are repressing or activating.

Relevant literature:

[1] Wolpert L (October 1969). "Positional information and the spatial pattern of cellular differentiation". *J. Theor. Biol.* **25** (1): 1–47.

[2] Driever W, Nüsslein-Volhard C (July 1988). "The bicoid protein determines position in the *Drosophila* embryo in a concentration-dependent manner". *Cell*. **54** (1): 95–104.

1) Algorithmic thinking:

Have a look at the reaction-diffusion equation for the morphogen and the implementation of it in the file *morphogengradient_to_segments.py* in the `reaction_diffusion_step` function, as well as at the initialization of the array `u`. How is this different from the gradient formation modeling we discussed during the lecture? What is done at the terminal boundary in the length direction and why does this make sense? (Hint: uncommented we provided code doing essentially the same but

not using numpy arrays and hence written in a less compact matter to help you understand what is happening)

- 2) **Important concept:** Play with the morphogen diffusion rate and the morphogen decay rate and describe what happens. What happens in terms of dynamics and steady state if you change both, but the ratio stays the same? Hint: it might help to draw a horizontal line at a certain height to ease comparison.
- 3) **Biology & mathematical thinking:** Next, we are going to introduce the genes A, B and C in the model. We want these genes to be expressed dependent on M, and in the head, trunk and tail respectively, so A on the left, B in the middle and C on the right. Think of what conditions in terms of M should lead to expression of A/B/C. What Hill functions (normal/inverse/any combination) corresponds to those conditions? Write down (pen and paper, not in code) full equations for the genes, do we need any other terms than just Hill functions, would we need specific parameter conditions?
- 4) **Biology & algorithmic/mathematical thinking:** From hereon we assume that the morphogen gradient reaches steady state very quickly, and no longer use the numerical implementation of the diffusion equation and instead work with a superimposed morphogen profile defined by `reaction_diffusion_gradient` to save time. (Hint: to not call the function any more use `#` in front of where it is called) Change the simulation loop such that it computes the morphogen gradient once. What type of function is the superimposed morphogen profile and how does this relate to question 2.
- 5) **Biology & algorithmic/mathematical thinking**
 - a) Now create functions to update A, B and C according to your equations from the previous question. You may use the predefined `hill` and `ihill` functions that are provided in the code. For simplicity, you may keep most of the parameters the same across genes, but some *have to be different* to ensure the right location of the genes (see your reasoning to the previous question). (Hint: using array properties to update A/B/C, like in the `reaction_diffusion_step` function, makes your code run a lot faster than using for-loops)
 - b) Also make sure that the levels of A, B and C are updated in the simulation loop. (Hint: using array properties to update A/B/C, like in the `reaction_diffusion_step` function, makes your code run a lot faster than using for-loops)
 - c) Next, ensure A, B and C are visualized in the bottom plot axis (copy and adapt the code for the visualization of M). You can add an extra third axis to the plot to visualize the (French) flag pattern, by getting which of the three genes is maximal at each location with `np.argmax(np.array([A, B, C]), axis=0)` and turning that into an array of RGB colors of choice.
 - d) Do you get the expected “French flag” pattern? If not, think of why not and improve your equations from previous question, parameters or your code.

- 6) **Biology:** At some point in development, the morphogen gradient will disappear, for example in the case of the *Drosophila* Bicoid gradient because the maternal mRNA is degraded. Predict what will happen to the expression of A, B and C (and hence the French flag pattern) if the morphogen gradient disappears over time and assume A, B, and C are regulated by our equations (first try think about this without actually simulating this).
- 7) **Algorithmic/mathematical thinking:** Now write the code in `reaction_diffusion_gradient` that updates the morphogen concentration, such that after the time point A, B and C have gotten close to equilibrium, the morphogen gradient gradually disappears. Adapt the simulation loop where necessary and run your code: was your prediction on A/B/C from the previous question correct? Why/why not? Hint: you might need to increase the duration of your simulation, especially if it takes a long time for A, B and C to reach equilibrium (or you can change the parameters to speed things up by using same production/degradation rate ratio yet higher absolute values of the individual parameters).
- 8) **Biological & mathematical thinking:** A, B and C cannot remain in a stable pattern if they are only influenced by M. How can we stabilize the pattern in absence of M? Test your ideas by creating new update functions for A, B and C and let these new 'rules' kick in at the same time when M starts to decline. Again, you may find Hill functions useful and perhaps also the before/after switch time structure used in `reaction_diffusion_gradient` for M. (Hint: think about how the genes should affect each other, and assume that in this new phase, when genes have already been initialized in absence of repression the genes will be expressed). Can you maintain the expression domains of A, B and C and does their shape change?
- 9) **Biology & algorithmic/mathematical thinking:** A sudden switch from phase 1 (stable M gradient) to phase 2 (decaying M gradient) resulting in the genes following different differential equations is biologically implausible. In reality, genes have complex promoters and enhancers integrating different inputs that arise in different developmental stages. Try to come up with one integrated expression for each gene, incorporating simultaneous input from the morphogen gradient and the other genes. Create new update functions for A/B/C and test your ideas. Can you get a stable pattern before and after the decline in M? A couple of things you could consider:
- Think of how positively and negatively M regulated expression behave once the gradient starts declining: what is the best way to combine (AND/OR) that with the regulation by the genes?
 - Consider splitting up the M regulated expression of middle gene B into a positive and negatively regulated morphogen part before integrating the other genes inputs.

- c) It is also possible to give certain genes a bit of a constant boost to prevent their takeover by other genes due to timing issues

10) Biology & algorithmic/mathematical thinking: Write code to get noise in the morphogen gradient, both in its steady state and decaying phase. During which phase do the expression domains of A, B and C suffer more from the noise. Explain why? How could we make the system more robust in a manner that is also likely occurring in nature?

Extra questions if you're done early/ For master students

These questions need not be made in the order they are provided, you can choose what you would like to investigate.

11) Biological thinking: Play with the size of the domain to see what happens to the gene expression domains. What would this mean for an organism?

12) Biological & algorithmic thinking: Various mechanisms have been proposed to ensure that the domains of the morphogen controlled genes scale with the size of the domain. One proposed mechanism suggests the existence of an also diffusible “expander” molecule which expression is repressed by the morphogen but which itself either represses the degradation of the morphogen or enhances its diffusion (see e.g. <https://www.pnas.org/doi/full/10.1073/pnas.0912734107> and <https://onlinelibrary.wiley.com/doi/10.1111/dgd.12337>).

For the expander we can write:

$$\frac{\partial E}{\partial t} = p \frac{K^h}{K^h + M^h} - dE + D_E \Delta E$$

Assume E reduces degradation of M (easier to implement than enhancement of diffusion) and study the effect of scaling. (Hint, vary the size of the domain in the length direction but plot domains as a function of relative instead of absolute domain size to compare domain sizes).

13) Algorithmic thinking: In b, we can also implement other boundary conditions, especially for the right boundary. What happens to the profile if we make a no-flux boundary by copying the value at $n-2$ to $n-1$? And what if we set it to a sink (force concentration to zero)?

14) From question 4 onwards, how do things change if we do not assume a quasi-steady-state for the morphogen gradient (i.e. keep the morphogen dynamics instead of replacing it by the superimposed exponential)? How would you implement a disappearing gradient and how does it shape change the outcomes of the flag?

15) Implement your solution to make the system more robust for noise from question 10. Did it work?

