

Modeling planar signalling in AP axis development in *Xenopus laevis* using Petri Nets in Higher Level Developmental Biology

Rick van der Zwet, Tiago Borges Coelho
<hvdzwet@liacs.nl>, <borges.coelho@gmail.com>
LIACS - Leiden University, The Netherlands

December 20, 2009

1 Abstract

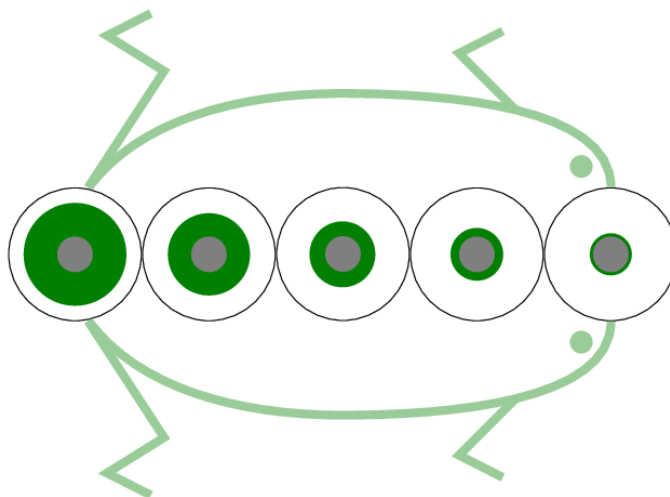
Planar signaling is a process that is part of the development of the AP axis in *Xenopus laevis* [Bertens09], in which cells accumulate proteins based on the saturation of nearby cells. If one cell produces n amount of proteins, it will initiate a transferring cascade to cells in the vicinity. This dissemination of proteins will eventually cease, considering that n is a finite variable. There is a gradation in the amount of proteins transferred, meaning that neighboring cells get $n/2$ the amount of proteins of the most saturated cell.

We are going to model this into Petri-Nets, since it supports a mathematical modeling language as well as a graphical interactive representation, which is well suited for this purpose, as we could model the process based on algorithms and also used it for automated model tracking and analysis, while having a visual representation of the process.

2 Method

Our approach to modeling Planar signaling is firstly construct a biological model of the process, one that illustrates the phenomenon taking place (shown at figure 1). Next we decided to create a conceptual model, one that abstracts from the biological model and contains some mathematical equations that describe the processes occurring in nature.

Figure 1: Conceptual abstract model



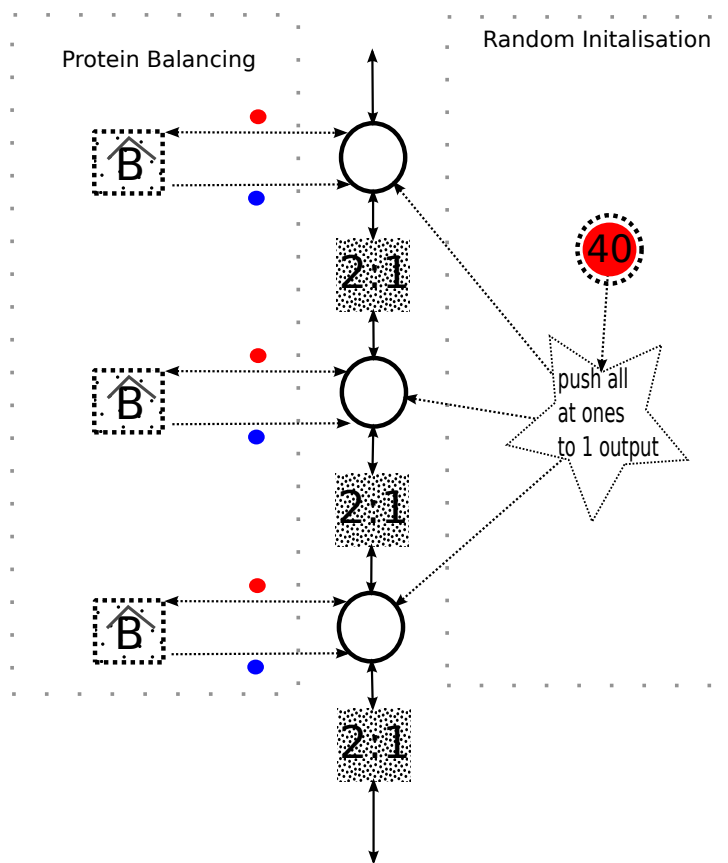
Finally we are going to construct a Petri-Net model using the software *CPNTools*¹.

3 Modeling

For the conceptual model (Fig. 2) we start with a building block that is an abstraction of a cell (figure: circle), which can then be coupled to other cells (figure: arrows). The abstraction contains two different token types. For one the proteins are modeled through red tokens, and secondly the proteins generate a second process, the level of posterisation (blue tokens) which is also required in the model. We assume a 1:1 mapping between the amount of proteins and the posterisation, taking into consideration that when an **INITIAL** protein is used (e.g. has a posterisation counterpart) in this process is called **ACTIVATED**. We assume that the proteins to posterisation process is taking place at the same time as the proteins distribution. This is represented in a special format, represented by the square B in the figure. It tries to match the posterisation to the same level as the proteins present. The moment the protein level lowers, the posterisation will remain the same. In pseudo-code:

¹<http://wiki.daimi.au.dk/cpntools/cpntools.wiki>

Figure 2: Planar signaling model



```

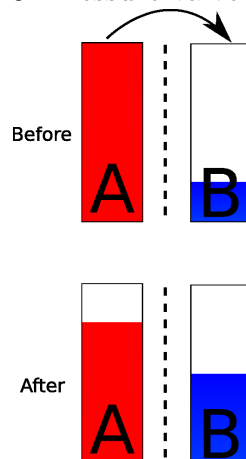
if numPos < numProteins then
  numPos = numPos + 1
endif

```

`numProteins` is the proteins available and `numPos` is the posterisation present.

The connectors between the cells (the membranes) have a special property. One can see them as pressure valves (like figure 3) that close when the volume in the containers (cells) complies with the following property $A/2 < B$, or siphons when the property before is not achieved, passing volume from A to B at a certain rate (`flowSpeed`). This rate could depend on the difference, actual value present or something else. Please keep in mind that negative values could sometimes appear, hence there is a need to check whether the source is bigger or equal than the `flowSpeed`.

Figure 3: Pressure valve example



For this case there exists no standard Petri-Net component, hence this requires the creation of a new property, which is described in pseudo-code below:

```

flowSpeed = n
if A > 2 * B and A => flowSpeed then
  A = A - flowSpeed
  B = B + flowSpeed
else if B > 2 * A and B => flowSpeed then
  B = B - flowSpeed
  A = A + flowSpeed
endif

```

Planar signaling could theoretically start in every cell, by inserting some amount of proteins. In our model this is represented as n `INITIAL` tokens being put in a cell chosen at random.

4 CPNTools implementation

CPNTools has some shortcomings when it comes to modeling higher level developmental biology. One is the shortcoming of balancing. It does not allow the reading of how many tokens are present in a certain state and base action upon them.

As workaround for this (see Fig 4) we used a dump gradation function. In our case it simply takes 3 tokens and pushes 1 forward while converting 2 directly to `ACTIVATED`. This does not take into consideration if some external source changes the amount further up.

Secondly it misses the possibility to easily randomize the initialization. As a quick fix we hacked it to choose between starting at the head or the tail. In this implementation the protein to gradient process is taking place at cell *A* at the same time that the proteins get transferred from cell *A* to *B*. It should be noted that the model avoids the notion of timed firing sequences; meaning that the firing sequences will not occur at pre-determined times. This could be changed in the future to trigger a timed activation of the `INITIAL` to `ACTIVATED` process as modeled in figure 2.

5 Conclusion

Using Petri-Nets for modeling biology processes is a powerful framework, one which could be well expandable. The proof of concept implementation and visualization however is lacking. *CPNTools* does not currently provide a powerful enough tool-set for the modeling purposes. This could be improved with the support of a programming language that can produce algorithms that can replace the mathematical functions of arcs.

References

- [Bertens09] Laura M.F. Bertens et al., Using Petri Nets in Higher Level Developmental Biology: A case study on the AP axis development in *Xenopus laevis* Extended Abstract, 2009

Figure 4: CPNTools implementation

