

# **An executable stochastic model for Fatty Acid metabolism**



**Argyris Zardilis**

Department of Applied Mathematics and Theoretical Physics  
University of Cambridge

This dissertation is submitted for the degree of  
*Master of Philosophy*

St Catharine's College

July 2014



I would like to dedicate this thesis to my loving parents ...



## **Declaration**

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration, except where specifically indicated in the text. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Argyris Zardilis

July 2014



## **Acknowledgements**

And I would like to acknowledge ...





## **Abstract**

This is where you write your abstract ...



# Table of contents

|   |             |
|---|-------------|
| <b>Table of contents</b>  | <b>xi</b>   |
| <b>List of figures</b>  | <b>xiii</b> |
| <b>List of tables</b>   | <b>xv</b>   |
| <b>1 Introduction</b>   | <b>1</b>    |
| 1.1 Dynamical systems theory and Flux Balance Analysis in Biology . . . . . | 1           |
| 1.2 Challenge of lipid metabolism . . . . .                                 | 1           |
| 1.3 Computational models in Biology . . . . .                               | 1           |
| 1.4 Outline of work . . . . .   | 1           |
| <b>2 Methods</b>  | <b>3</b>    |
| 2.1 Petri Nets . . . . .  | 3           |
| 2.1.1 Syntax and Semantics of Basic nets . . . . .                          | 3           |
| 2.1.2 Stochastic Petri Net extension . . . . .                              | 5           |
| 2.1.3 Application to Biochemistry . . . . .                                 | 6           |
| 2.2 Stochastic Pi-calculus . . . . .  | 7           |
| <b>3 Work</b>   | <b>9</b>    |
| 3.1 Basic model of Fatty Acid synthesis . . . . .                           | 9           |
| 3.1.1 Original and reduced order model . . . . .                            | 9           |
| 3.1.2 Point estimation of parameters . . . . .                              | 9           |
| 3.1.3 Exact likelihood and parameter profiles . . . . .                     | 9           |
| 3.1.4 An estimation of input flow parameters . . . . .                      | 9           |
| 3.1.5 A stochastic $\pi$ -calculus version . . . . .                        | 9           |
| 3.2 Extended model of FA synthesis . . . . .                                | 9           |
| <b>4 Discussion</b>   | <b>11</b>   |

|  |           |
|--|-----------|
| <b>References</b>                                | <b>13</b> |
| <b>Appendix A Installing the CUED class file</b> | <b>15</b> |

# List of figures

|     |  |   |
|-----|--|---|
| 2.1 | Small basic Petri Net example . . . . .      | 5 |
| 2.2 | The token game for basic Petri Nets. . . . . | 5 |



## List of tables





# **Chapter 1**

## **Introduction**

- 1.1 Dynamical systems theory and Flux Balance Analysis in Biology**
- 1.2 Challenge of lipid metabolism**
- 1.3 Computational models in Biology**
- 1.4 Outline of work**



# Chapter 2

## Methods

### 2.1 Petri Nets

#### 2.1.1 Syntax and Semantics of Basic nets

A Petri Net is a 4-tuple  $(P, T, pre, post)$  defined as:

- a set of *places* (or conditions)  $P$
- a set of *transitions* (or events)  $T$
- a *preconditions map*  $pre : T \rightarrow \mathbf{m}P$  which assigns a multiset of places  $pre(t)$  to each transition  $t \in T$
- a *postconditions map*  $post : T \rightarrow \mathbf{m}P$  which assigns a multiset of places  $post(t)$  to each transition  $t \in T$

where  $\mathbf{m}P$  is the space of multisets over  $P$  with a multiset over  $P$  defined as function  $f : P \rightarrow \mathbb{N}$ . The state of the system, called a marking, is again a multiset  $\mathcal{M}$  over  $P$ . We can think of the marking as the distributed global state of the system. It is also common when defining a Petri Net to give the initial marking of the system usually written as  $\mathcal{M}_0$ .

The operational semantics of Petri Nets is defined in terms of changes in the global state through the action of the transitions  $T$  of the PN:

$$\mathcal{M} \xrightarrow{t} \mathcal{M}'$$

Here when a transition(event)  $t$  occurs it changes the state of the system from marking  $\mathcal{M}$  to marking  $\mathcal{M}'$ . Unlike automata and Turing Machines however a transition does not occur

from a single global state but instead it only affects part of the state:

$$\mathcal{M} \xrightarrow{t} \mathcal{M}' \text{ iff } pre(t) \leq \mathcal{M} \text{ and } \mathcal{M}' = \mathcal{M} - pre(t) + post(t)$$

For 2 multisets  $f$  and  $g$  over set  $X$ ,  $f \leq g$  is defined as  $f \leq g \iff \forall x \in X f_x \leq g_x$ . So a transition  $t$  is said to be 'enabled' if its preconditions are sufficiently marked ( $pre(t) \leq \mathcal{M}$ ) and when it 'fires' it changes the marking in the places in its vicinity, namely the set of places  $\{p \mid pre(t)_p \neq 0 \text{ or } post(t)_p \neq 0\}$  ( $\mathcal{M}' = \mathcal{M} - pre(t) + post(t)$ ). The  $post$  and  $pre$  condition maps define the causal independence between transitions and add the ability of Petri Nets to model concurrency and non-determinism. Two (or more) transitions are concurrent if they do not share any preconditions. Non-determinism is added through dependencies of transitions which share preconditions. In that case a race condition is created because the dependent transitions compete at their shared preconditions with only one of them being able to fire at each step.

This formal definition of transitions leads to an algorithm for the executions of a PN as follows:

1. Initialise the net with  $(P, T, pre, post)$  and set state  $\mathcal{M} = \mathcal{M}_0$
2. Find enabled transitions,  $enabled \subseteq T, \forall t \in T \text{ if } pre(t) \leq \mathcal{M}$
3. Choose transition  $t$  from set of enabled at random
4. Update state according to  $\mathcal{M} = \mathcal{M} - pre(t) + post(t)$
5. Repeat steps 2-4 until there are no more enabled transitions or an external stop condition is met (e.g. max number of steps)

The main strength of Petri Nets though lies on their very intuitive and widely used graphical representation. Petri Nets are represented as bipartite graphs with two sets of nodes, the places  $P$  and transitions  $T$  which are denoted by circles and rectangles respectively. A weighted directed edge with weight  $n > 0$  is added between place  $p$  and transition  $t$  in the direction  $p \rightarrow t$  if  $pre(t)_p = n$ . A weighted directed edge with weight  $n > 0$  is added between transition  $t$  and place  $p$  in the direction  $t \rightarrow p$  if  $post(t)_p = n$ . The markings are denoted as dots (tokens) in the circles depicting the places. So for example consider the net with  $P = \{p1, p2, p3\}$ ,  $T = \{t1\}$ ,  $pre = \{(t1, \{(p1, 2), (p2, 1), (p3, 0)\})\}$ ,  $post = \{(t1, \{(p1, 0), (p2, 0), (p3, 2)\})\}$ , and marking  $\{(p1, 3), (p2, 1), (p3, 1)\}$ . The graphical depiction of this net would be the one shown in Figure 2.1. We could then also define the pre-places of a transition  $t$  as all the places  $p$  such that  $pre(t)_p > 0$  and the post-places as all the places  $p$  such that  $post(t)_p > 0$ .

The operational semantics of Petri Nets can also be defined very naturally in this graphical notation. When a transition  $t$  fires  $pre(t)_p$  tokens are consumed from all the pre-places

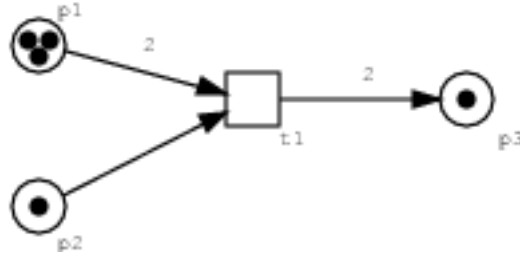


Fig. 2.1 Small basic Petri Net example

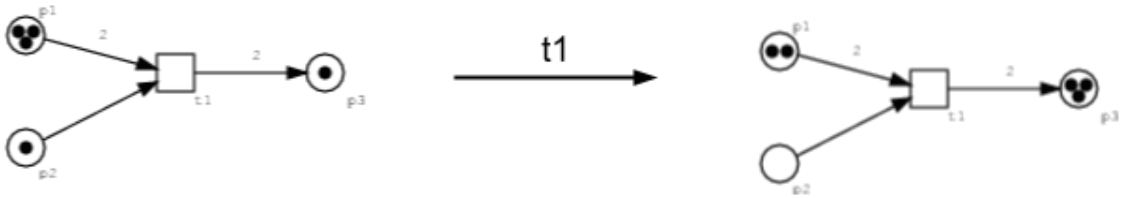


Fig. 2.2 The token game for basic Petri Nets.

$p$  and  $post(t)_p$  tokens are added to all the post-places  $p$ . For the net given above (Figure 2.1) when transition  $t1$  fires the 2 tokens are removed from  $p1$  and 1 from  $p2$  and 2 are added to  $p3$  (see Figure 2.2). The execution of the net can be seen as the flow of tokens through the net as transitions fire at each step according to the execution algorithm given above. This graphical view of the execution is called the 'token game' and it is very useful for gaining an insight into the dynamic behaviour of the system being modelled.

### 2.1.2 Stochastic Petri Net extension

Stochastic Petri Nets are an extension to the original Petri Net formalism to explicitly model time. Notice that even though there is an ordering of transitions implied by their execution according to the algorithm given in the previous section the concept of time is not modelled explicitly in the basic Petri Net formalism. In order to model time explicitly Stochastic Petri Nets (SPNs) introduce a waiting time associated with each transition  $t$ , defined as random variable  $X_t$  distributed exponentially with potentially marking-dependent rate  $\lambda_t(pre\_places(t))$  where  $pre\_places$  is defined as before. Formally a SPN is a 5-tuple  $(P, T, pre, post, v)$  with everything defined as before with the addition of the map  $v : T \rightarrow H$  where  $H$  is the set of all hazard functions  $H = \{h_t \mid h_t : \mathbb{N}^{|pre(t)|} \rightarrow \mathbb{R}\}$ . Hazard function is

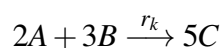
the typical name used in stochastic processes for the function giving the rate of the exponential time variable. In this case the hazard function  $h_t$  gives  $\lambda_t$  and the domain of  $h_t$  we will restrict to only the marking of the pre-places of  $t$ . This wait time associated with each transition changes the execution algorithm for Petri Nets; now when more than one transition is enabled a wait time is sampled from all of them and the one with the least waiting time gets to fire:

1. Initialise the net with  $(P, T, pre, post, v)$ , set state  $\mathcal{M} = \mathcal{M}_0$ , and  $time = 0$
2. Find enabled transitions,  $enabled \subseteq T, \forall t \in T$  if  $pre(t) \leq M$
3. For each enabled transition  $t$  sample a wait time from an exponential distribution with rate  $\lambda_t(pre\_places(t))$ . Pick the transition  $t_i$  with least wait time to fire.
4. Proceed time  $time = time + \tau_i$
5. Update state according to  $\mathcal{M} = \mathcal{M} - pre(t_i) + post(t_i)$
6. Repeat steps 2-4 until there are no more enabled transitions or an external stop condition is met(e.g. max number of steps)

With this new variant of Petri Nets we maintain the non-determinism through the competition of enabled transitions with shared preconditions but by changing their rates we can favour some transitions over others. This Stochastic Petri Net formulation gives rise to Continuous Time Markov Chain and the execution algorithm defined above is more or less the Gillespie algorithm which is an exact algorithm for simulation of Markov jump processes. It is this Petri Net variant that we will use to model our system of interest, the Fatty Acid synthesis/elongation pathway. In the next section we describe the correspondence between a biochemical network and especially a metabolic pathway to a Stochastic Petri Net and some examples of previous uses of SPNs in biology/biochemistry.

### 2.1.3 Application to Biochemistry

The correspondence between a biochemical system and a Stochastic Petri Net is as follows: the chemical species of the system become places, reactions become transitions and their rates the rates of the transitions, and the stoichiometry of the reactions is represented in the *pre* and *post* maps. This is a very natural correspondence between chemical reactions and Petri Nets. In fact the standard notation for chemical notation is itself a process algebra albeit not a very formal one. So for example consider a made-up reaction that takes 2 molecules of A and 3 molecules of B and produces 5 molecules of C:



This is an event, a reaction with rate  $k$ , which tells us how the state of the system changes but although implicitly we know the conditions for it to take place they are not formally captured by the informal process algebra of chemical reactions as defined by the standard arrow notation. A Petri Net representation of this reaction however captures the reaction more formally:

## 2.2 Stochastic Pi-calculus

This is going to contain a very brief introduction to stochastic pi-calculus.





# **Chapter 3**

## **Work**

### **3.1 Basic model of Fatty Acid synthesis**

#### **3.1.1 Original and reduced order model**

#### **3.1.2 Point estimation of parameters**

#### **3.1.3 Exact likelihood and parameter profiles**

#### **3.1.4 An estimation of input flow parameters**

#### **3.1.5 A stochastic $\pi$ -calculus version**

### **3.2 Extended model of FA synthesis**



# **Chapter 4**

## **Discussion**



## References



# Appendix A

## Installing the CUED class file

$\text{\LaTeX}$ .cls files can be accessed system-wide when they are placed in the  $\langle\text{texmf}\rangle/\text{tex}/\text{latex}$  directory, where  $\langle\text{texmf}\rangle$  is the root directory of the user's  $\text{\TeX}$  installation. On systems that have a local  $\text{texmf}$  tree ( $\langle\text{texmflocal}\rangle$ ), which may be named “ $\text{texmf-local}$ ” or “ $\text{localtexmf}$ ”, it may be advisable to install packages in  $\langle\text{texmflocal}\rangle$ , rather than  $\langle\text{texmf}\rangle$  as the contents of the former, unlike that of the latter, are preserved after the  $\text{\LaTeX}$  system is reinstalled and/or upgraded.

It is recommended that the user create a subdirectory  $\langle\text{texmf}\rangle/\text{tex}/\text{latex}/\text{CUED}$  for all CUED related  $\text{\LaTeX}$  class and package files. On some  $\text{\LaTeX}$  systems, the directory look-up tables will need to be refreshed after making additions or deletions to the system files. For  $\text{\TeX}$ Live systems this is accomplished via executing “ $\text{texhash}$ ” as root.  $\text{MikTeX}$  users can run “ $\text{initexmf -u}$ ” to accomplish the same thing.

Users not willing or able to install the files system-wide can install them in their personal directories, but will then have to provide the path (full or relative) in addition to the filename when referring to them in  $\text{\LaTeX}$ .

