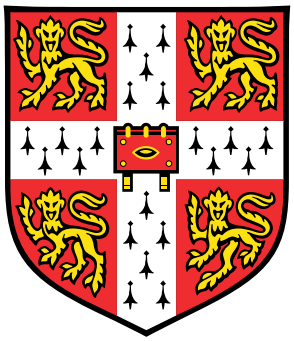
An executable stochastic model for Fatty Acid metabolism



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 August 2014

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 18,000 words including appendices, bibliography, footnotes, tables and equations.

Argyris Zardilis
August 2014

Acknowledgements

I would like to thanks my supervisors, Dr João Dias and Dr James Smith, for helping ease my introduction into this new for me area. I would also like to thank Dr Lee Roberts for providing experimental data for the tuning and validation of the basic model.

Abstract

Metabolites and reactions taking part in Lipid metabolism processes are often poorly annotated. This makes it difficult to analyse with current traditional techniques used for metabolic processes like Flux Balance Analysis. Here we propose an alternative stochastic reaction-centric view of Lipid metabolism to better capture some of its defining characteristics on a more local scale: iterative conversion processes, probabilistic decisions, and crucial regulatory mechanisms. We also assess the formal graphical language of Petri Nets to capture this stochastic reaction-centric view and contrast it with the pi-calculus process algebra that is another popular approach for executable models in Biology. Here we particularly focus on Fatty Acid (FA) synthesis that is a core part of lipid metabolism. A basic high-level Petri Net and stochastic pi-calculus model for the FA synthesis/elongation presented along with techniques for tuning it with experimental datasets. The Petri Net model is then further extended to include an important regulatory mechanism of FA synthesis activity through the action of AMPK protein kinase.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Recent technological advances have led to a wealth of data that move the attention towards understanding biological systems as interacting components. A systems level understanding is really important as it can be considered as distributed information processing giving rise to phenotype and the macroscopic behaviour of cells to sustain life. Biology as a discipline has not traditionally used any formal methods but as the systems under investigation grow in size, more formal approaches are needed. System Biology is a relatively new field that tries to fill the gap by bringing practises from traditionally more formal disciplines like Mathematics, Physics, and Computer Science [14].

In the area of metabolism, we go from a study of the properties of single enzymes and metabolites to the study of the behaviour of entire metabolic pathways and even entire genome-wide metabolisms of mammalian and model organisms. This accumulated knowledge about the interconnections in and between metabolic pathways is deposited in large online databases that integrate the information from various sources. While these annotations are a rich source of information they only provide a static picture. Since metabolism is a temporal process sometimes a more dynamic picture is needed. Mathematical techniques like Flux Balance analysis originating from Dynamical Systems Theory are particularly popular in the analysis of the flows in and out of metabolic pathways.

Recently the biochemical reasons behind different diseases were pinpointed in the loss of balance between anabolic and catabolic activities in cells. Most of these activities are thought to be tightly regulated to avoid excess production and accumulation of metabolites. Where this happens it seems to be consistent with observed disorders. The products and intermediaries of lipid metabolism are central in many metabolic disorders and diseases. While Flux Balance Analysis is of great importance in the study of large metabolic networks it is not so suited in studying Lipid metabolism. Lipid metabolism is poorly annotated in the databases and it is hard to analyse using constraint based methods like FBA.

Constraint-based methologies (FBA) have avoided systems like Lipid metabolism probably because they require more local probabilistic metabolic processes and crucial regulatory mechanisms.

Here we propose an alternative reaction-centric stochastic view of metabolic systems and lipid metabolism in particular which we is an important approach. We hope that this will lead in mechanistic characterisation of crucial local metabolic processes and their regulatory mechanisms. We also propose the formal language of Petri Nets, which has been successfully used before in biochemical networks, as a potential tool to capture this alternative reaction-centric view of lipid metabolism. The goals and the work done towards this project were therefore two-fold: Firstly create a reaction-centric model of the Fatty Acid(FA) synthesis/elongation process which is a core part of lipid metabolism and secondly assess the formal modelling language of Petri Nets as a potential tool to capture this view. Since part of the work done was method development and since we noticed a more general trend towards executable formal models in biology [9] we also note that an alternative version of the model is the process algebra of pi-calculus. We therefore attempt to contrast the two formal modeling approaches: net models and process algebra models.

In this introduction, I start by giving an overview of the currently most popular method for the description of metabolic pathways(FBA), outline the biological importance of lipid metabolism and the motivation for the reaction-centric view, and finally I try to place Petri Nets and pi-calculus(the two methods used) in the spectrum of formal distributed computing modelling language world.

1.1 Dynamical systems theory and Flux Balance Analysis

Systems Biology attempts to master the complexity and understand metabolic pathways and any biological system by usually contructing models. There are two types of models based on their construction method. Bottom-up models are constructed by extracting biological knowledge from experimental datasets. This category includes for example the diagrammatic means of capturing the interconnected structure of metabolic pathwayw. Top-down models on the other hand start from a detailed knowledge of the system to get a mechanistic model of the system [30].

Early attemps to capture the interactions and dependencies between components were made with bottom-up models which resulted in large diagrams that capture the conversion processes happending inside metabolic pathways. While these diagrams are important and are a useful representation of our knowledge about metabolism, they lack the dynamic nature which is crucial in a systems level understanding of a system which should not only

include the connections between the components but also how these components interact over time, how their behaviour is controlled, and how they behave when perturbed by external stimuli [17].

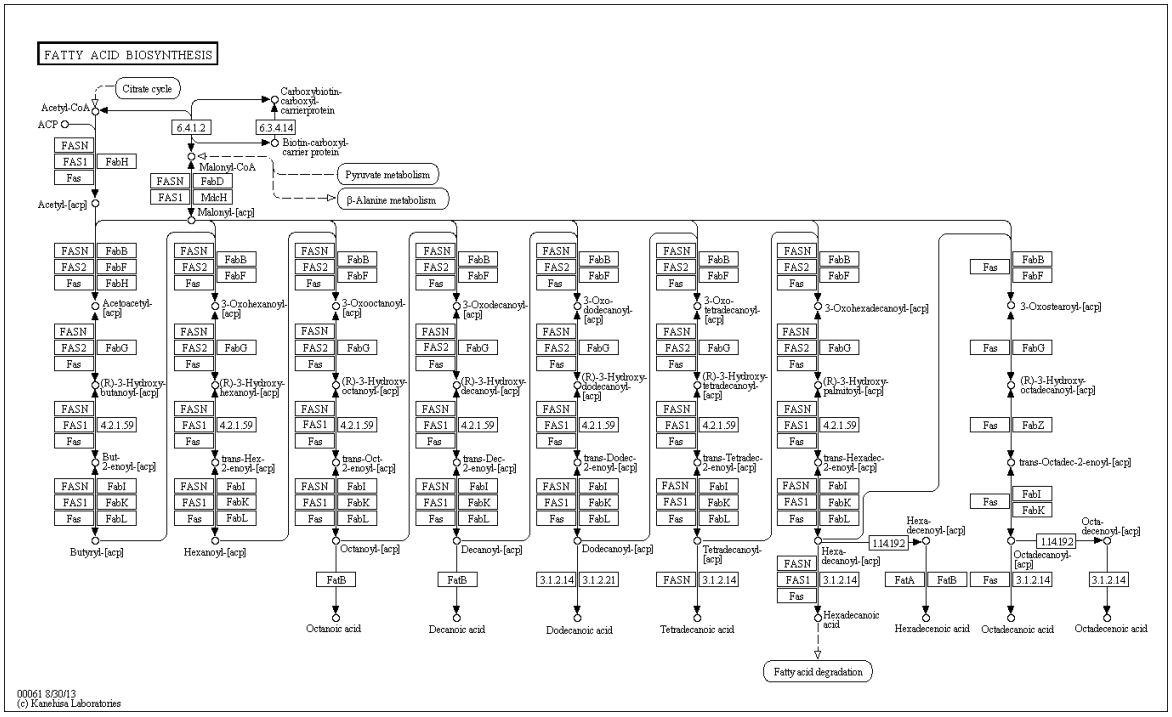


Fig. 1.1 Fatty acid synthesis pathway described by the standard schema in KEGG to capture the component iterative pathway of FA synthesis. The working of these systems (from experiments for example) is integrated with data from multiple sources.

This need to look into dynamic behaviour led to Dynamical System theory which based on its success in Physics has found its way in many other fields. Dynamical system theory captures the relationship between continuous quantities in difference-differential equations that describe the evolution of state variables in terms of changes in other state variables and even themselves thus capturing the interaction element. Since the time of Newton and classical mechanics, where these ideas originated, the toolbox of dynamical systems theory has grown to include techniques for qualitative understanding of system without the need for numerical or analytic solutions.

In biochemistry, each species in the system is represented by its concentration leading to a differential equation for each species. The rate of change of the concentration of a species is described in terms of in-flows (increasing the rate of change/production) and out-flows (decreasing the rate of change/degradation). These flows can be dependent on the concentrations of other variables(species) that participate in the same reaction scheme. Consider, for example, a system with n components that we can group into the global state

of the system $\mathbf{x} = (x_1, x_2, \ldots x_n)$. All the reactions that take place in the system change, in discrete levels, the numbers of molecules of the species:

$$\begin{aligned}
\mathbf{x} &\xrightarrow{r_1(\mathbf{x})} \mathbf{x} + \boldsymbol{\delta_1} \\
\mathbf{x} &\xrightarrow{r_2(\mathbf{x})} \mathbf{x} + \boldsymbol{\delta_2} \\
&\vdots \\
\mathbf{x} &\xrightarrow{r_m(\mathbf{x})} \mathbf{x} + \boldsymbol{\delta_m}
\end{aligned}$$

So for every reaction we have one such rule and the discrete levels δ_k by which the species change for every reaction. Each reaction has an associated and possible state dependent rate $r_k(x)$ that is the expected number of times it takes place in a unit of time. The in-flows are the positive deltas and out-flows the negative ones. The differential equation describing the evolution of the average numbers of species i is then the sum of the in- and out- flows over all reactions in the system:

$$\frac{dx_i}{dt} = \sum_m r_m(\mathbf{x}) \delta_{mi}$$

Usually this formulation of the problem leads to integration, with a numerical ODE solver of the above system to get the dynamic behaviour of the system. The usual problem mathematical biologists face is that the reaction rate function (the $r_k(\mathbf{x})$) contain parameter constants that are not usually known.

In the contex of metabolic systems however, there is a mathematical technique, called Flux Balance Analysis, that is used to compute these flows without explicit knowledge of these parameters by making some assumptions about the system to simplify the problem [25]. If we assume that that metabolites cannot accumulate within the cell and their levels are more or less balanced, then the system is immediately at steady-state. The steady-state assumption means that the average positive flux-defined as the sum of all the in-flows- and the average negative flux-defined as the sum of all the out-flows- are equal. This in turn means that the rate of change for every species, defined as the sum of all flows(negative and positive), is zero (cancelled). Solving for the fluxes leads to a linear equation for each species. For a more concise notation, all the δ_{nm} are packaged into $\mathbf{S}$, traditionally called the Stoichiometric matrix, and the fluxes in a vector $\mathbf{r}$. The problem is reduced to solving the system $\mathbf{Sr} = \mathbf{0}$ to calculate the fluxes $\mathbf{r}$ through our system of interest. The only problem with this formulation is that the system is underdetermined since usually the number of reactions is higher than the number of species. To solve this, the solution space is constrained with

the use of an objective function that leads to a linear programming problem formulation that is easily solvable. A small contrived example to illustrate the above process is given in Figure 1.2.

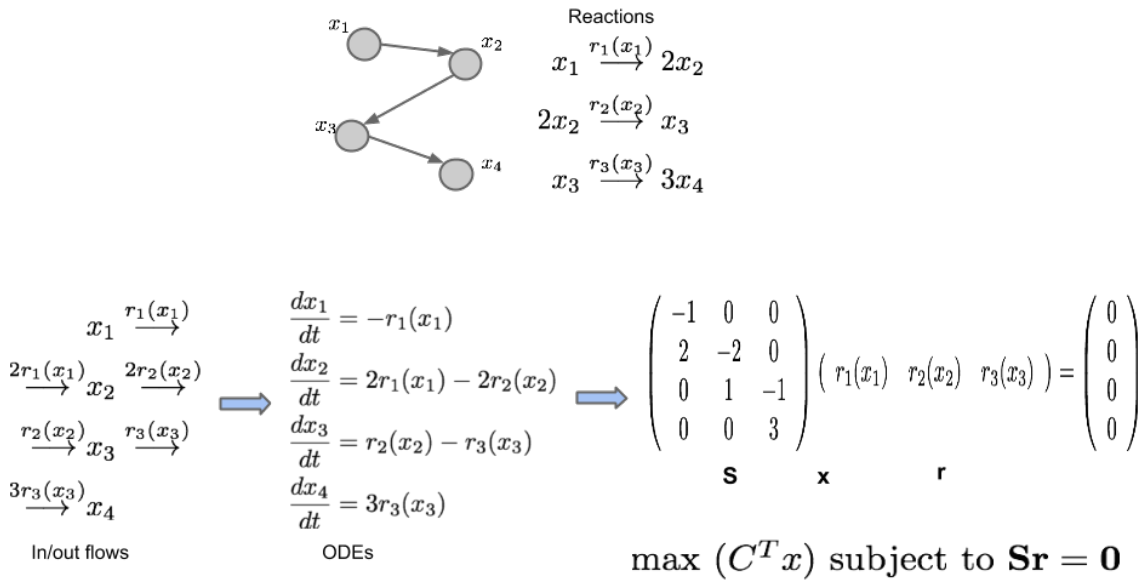


Fig. 1.2 A small example how starting from a bottom-up model like a pathway from KEGG we can go on to apply FBA and calculate the fluxes through our pathway

Flux balance analysis is a very powerful technique since it allows us to overcome the problem of the rate function parameters and calculate the fluxes through the system in a computationally efficient way. It is not without its limitations though. For large networks of components finding a suitable objective function to used to transform to solve the underdetermined problem can be difficult. Different objective functions depend on obserable behaviour based for example on exponential growth or products of the fermentation process. More importantly and since it relies on the average behaviour as described by the differential equations it is not able to capture stochastic, non-deterministic behaviour.

1.2 Challenge of lipid metabolism

Unlike intermediate metabolism, well-characterised with stoichiometries per reaction, there are molecules of metabolism that are poorly annotated, involving iterative chemical processes, hierarchical processes and non-selective processes. Examples include Redox Reactions, toxic metabolism, lipid metabolism. Here we try to capture the latter one. Lipid metabolism is the set of all the anabolic and catabolic processes that involve lipid products

that serve mainly as energy stores or in the form of lipid bilayers as membranes. The starting point of lipid metabolism are the Fatty Acids which act as building blocks for more complex lipids like diacylglycerols and triacylglycerols. Lipids are produced by the organism to serve as intracellular compartments and energy stores in the well-fed state when there is an excess of carbohydrates and no immediate energy requirements. Any excess glucose molecules are converted to lipids as follows: When the glucogen stores are full, the carbon from glucose molecules will find its way through the glycolysis pathway blocked at the level of phosphofructokinase so they take a diversion through the Pentose Phosphate pathway and then join the glycolytic processes at a later stage bypassing the block. Then they continue down the normal route to Acetyl-CoA and the Krebs cycle in the mitochondria. Here since the immediate energy requirements are low Acetyl-CoA only makes it to the first stage of the Krebs cycle producing Citrate instead of continuing through the cycle and then to the Respiratory chain to produce energy. Since Citrate cannot proceed any further in the cycle, its levels accumulate and at some point it translocates from the mitochondrion into the cytosol where it is converted to Acetyl-CoA again to serve as a precursor to Fatty Acid Synthesis which requires the NADPH produced by the Pentose Phosphate pathway. Once Fatty Acids are synthesised they can go on to form more complex lipid products like DAGs or TAGs or get further modified by the Fatty Acid elongation pathway or by adding bonds to their CH tails. The beta-oxidation pathway in mitochondria breaks down Fatty Acids into Acetyl-CoA which is then fed into the Krebs cycle [29].

Production processes have equivalent catabolic processes and a balance between the two is essential for metabolic homeostasis in organisms. This balance is regulated by control signals within and between pathway that also integrate signals from the environment. In diseases like type 1 Diabetes, this balance is disrupted because of insulin deficiency that breaks the regulatory signals. Fatty Acids and lipid metabolism pathways play a key role in this chain of disruption. Also, free Fatty acids are associated with obesity and type 2 diabetes [4].

We notice from the above brief overview and the pictorial description of the interconnected pathways(Figure 1.3) that in metabolism and lipid metabolism products flow through the network occurs with an iterative chain-like conversion process and that at several points there are probabilistic decisions to be made for the next step in the chain. The decisions are tightly regulated by signals from other pathways and the environment and it is assumed that disruption of these signals is observed in diseases like Diabetes. It is important therefore to have a mechanistic explanation of the 'trip' through the metabolic chain and the non-deterministic 'decision-making' that happens and its regulation. We believe that in order to get this mechanistic characterisation, we benefit from a stochastic reaction-centric view of

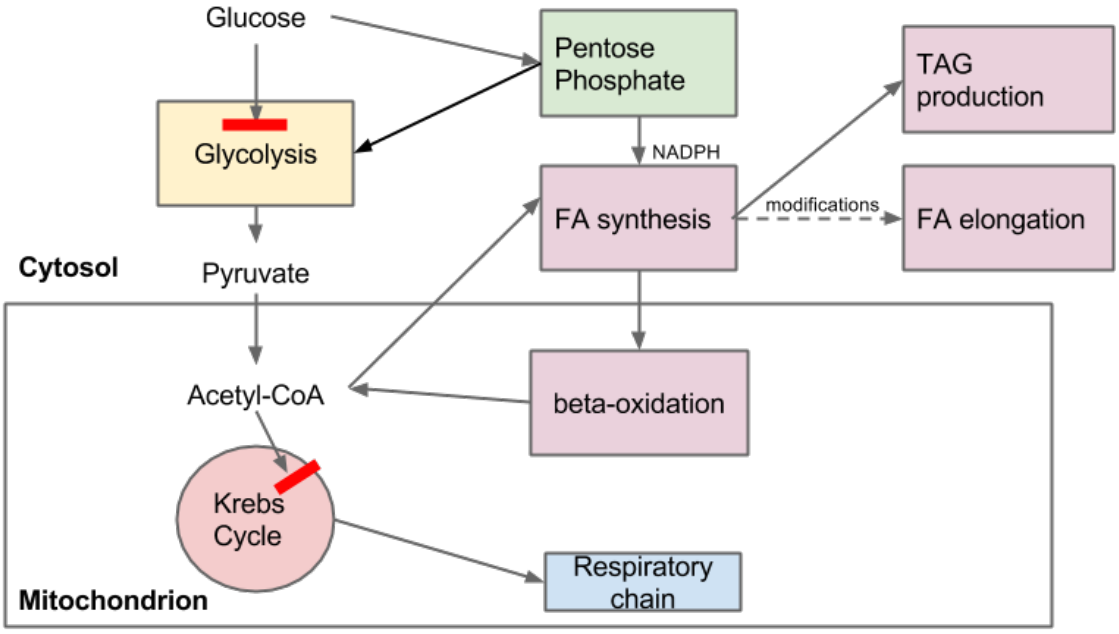


Fig. 1.3 A diagram outlining the main pathways involved in lipid metabolism.

lipid metabolism to give a more local perspective of the iterative conversion processes and the stochastic nature of the probabilistic decision-making process. In this study we focus on the Fatty Acid Biosynthesis pathway which is an important part of lipid metabolism. It exhibits the characeteristics that are suited to a reaction-centric analysis: iterative behaviour with non-deterministic decisions and regulatory mechanisms from other pathways.

A biochemical system, like the one given in the previous section, consisting of a number of reactions that alter the integer numbers of molecules of the species describes a stochastic process and a Continuous Time Markov Chain (CTMC) in particular. The differential equations we used as a basis for FBA are just a deterministic view of the average behaviour. The dynamics of the system can be set in motion inside a computer using the Gillespie exact simulation algorithm that creates sample time-series [12]. Statistical information, like the strength of the fluctuations (normalised variance) can be computed by collecting a lot of these traces.

In this study we propose Petri Nets as an alternative language for capturing this reaction-centric view. We think this is more expressive than the standard Markov processes formulation. While Petri Nets and their evolution through their formal operational semantics also describe a CTMC and are thus equivalent in power with the standard Markov jump processes. We believe that they possess some characteristics (see next section and Methods)

that makes them useful in the analysis of the reaction-centric view of lipid metabolism. Language is important not only because a syntax is needed to define our models but also because the notation we use is our tool of thought and the use of the correct language for thinking about a problem can enhance our understadnding and ultimately help us solving it [15]. That is why we have more than one high-level programming language despite the fact that all of them are Turing complete and therefore have the ability to express all computations.

In the next section, we try to place Petri Nets, the main modelling language used, and pi-calculus, the alternative used as a way to contrast net and process algebras models, in a spectra of formal modelling methods for distributed computation.

1.3 Computational models in Biology

Computer Science, although a fundamentally different discipline than Biology, has undergone a similar transformation in its focus from single information processing entities to systems of interacting entities(see the Internet, clusters). The term computation at the distributed level is now broader; it not only describes mere calculation but it also includes the interaction between components. This change in the term computation can be seen through the models we use to capture the notion of computation. At the early days of computing(even before actual electronic computers) the models of computation included lambda-calculus and Turing Machines. These early formalisms capture computation differently, Turing Machine as global state transitions in an Abstract Machine and lambda-calculus as reduction rules in a calculus, but they are effectively equivalent in expressive power(see Church-Turing thesis). As computer systems grew there was an interest for similar minimalistic languages to express distributed computation. Some formalisms that were invented during that period were Milner [20] Calculus of Communicating Systems, Lafont [19] Interaction nets, Milner et al. [21] pi-calculus, and Petri Nets [22].

The analogy between distributed computer systems and biochemical systems was made explicit by Berry and Boudol [3] and the Chemical Abstract Machine. The modelling langugages used for distributed computation are applicable in biological systems since they both include a high number of concurrent components with dependencies introducing non-deterministic behaviour. Fontana and Buss [10] went as far as to propose pi-calculus as an alternative to dynamical systems theory since he considered a calculus of objects that can change and their interactions more appropriate for describing biochemical systems than a calculus of continuous interacting quantities. Priami et al. [28] went on to use a stochastic version of pi-calculus to describe a biochemical systems and from then other formalisms were developed like Cardelli [5] with Brane-calculi, Danos and Laneve [7] with kappa, and

Ciocchetta and Hillston [6] with Bio-PEPA.

Petri Nets lean towards the automata and Turing Machines style of computation because their operational semantics are in terms of state transitions from a distributed global state. They have a very intuitive graphical notation and concurrency, non-determinism, and the causal independencies between events is inherent in their structure. They also have a very natural correspondence to biochemical reactions and in their basic form have been used mainly to capture qualitative properties of biochemical systems [2]. In this study we use the stochastic version of Petri Nets that explicitly models time to describe and quantitatively analyse the reaction-centric view of the FA elongation and synthesis process.

Process algebras, which pi-calculus is an example of, on the other hand define syntax to specify the behaviour and state transitions of the individual components of the system independently therefore reactions and compositionality are not explicitly captured. In this study we use Stochastic Pi Machine (SPiM) which is a programming language derived from stochastic pi-calculus but extended to include operational semantics for execution on an Abstract Machine and also a formal graphical notation [26, 27]. The fact that this variant shares characteristics from both net models and process algebras approached makes it an interesting case-study.

1.4 Outline of work

The main aim of this study was to assess the applicability of the Petri net formal language to capture a reaction-centttric view of lipid metabolism by producing a model of the Fatty Acid biosyntesis/elongation process. For completeness we also created a stochastic pi-calculus version of the model to contrast the two methods as part of the more general exploration towards executable formal models in Biology. An extended version of the basic FA synthesis model is also given that includes part of its regulatory mechanisms.

In chapter 2 an overview of the methods used in modelling the process is given, namely Petri Nets and stochastic pi-calculus. In Section 3, the basic and extended model are presented along with their tuning with available metabolomics data.

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* $pre : T \to \mathbf{m}P$ which assigns a multiset of places $pre(t)$ to each transition $t \in T$
- a *postconditions map* $post : T \to \mathbf{m}P$ 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 $f : P \to \mathbb{N}$. The state of the system, called a marking, is again a multiset $\mathscr{M}$ 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 $\mathscr{M}_0$.

Markings can change as transitions occur:

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

The change in the marking through the action of transition t is defined in the *pre* and *post* maps:

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

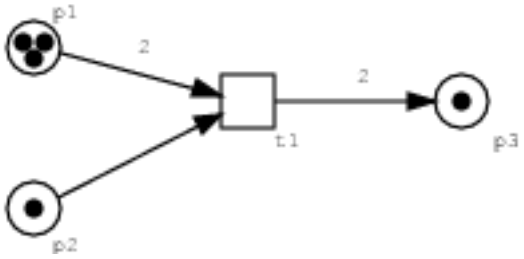


Fig. 2.1 Small basic Petri Net example

There is a widely used graphical notation for Petri Nets where transitions are represented by squares, places by circles, and the condition maps with directed weighted edges between the two types of nodes(squares and circles). Edges defined in the *pre* map are represented as inbound edges and edges defined in the *post* map are defined as outbound edges. Markings are represented by tokens inside the place. For example a Petri Net with three places and the following marking,

$$M_{p1} = 2, M_{p_2} = 1 \text{ and } M_{p_3} = 2$$

, would be represented by 2 tokens residing in the circle for place p_1, 1 token in the circle for place p_2, and 2 tokens in the circle for place p_3. If the net also has one transition t_1 and the following maps,

$$pre(t_1) = (p_1 = 2, p_2 = 1, p_3 = 0) \text{ and } post(t_1) = (p_1 = 0, p_2 = 0, p_3 = 2)$$

then the net would have inbound edges from p_1 and p_2 to the transition node weighted with 2 and 1 respectively and an outbound edge weighted by 2 between the transition node and the circle representing p_3(Figure 2.1). The places connected with inbound edges with a transition are called the pre-places and the places connected with outbound edges with a transition are called the post-places of that transition. When a transition fires, tokens are removed from its pre-places and are placed in its post-places according to the weights on the corresponding edges. For example in the previously defined net, if transition t_1 fires 2 tokens will be removed from p_1, 1 token from p_2, and 2 tokens will be added to place p_3(Figure 2.2).

This graphical view of the operational semantics of Petri Nets is usually referred to as the 'token game' because of this flow of tokens from pre-places to post-places for every transition that occurs. This leads to a simple algorithm for the execution of Petri Net models:

1. Initialise the net with $(P, T, pre, post)$ and set state $\mathscr{M} = \mathscr{M}_0$
2. Find enabled transitions, $enabled \subseteq T, \forall t \in T$ if $pre(t) \leq M$

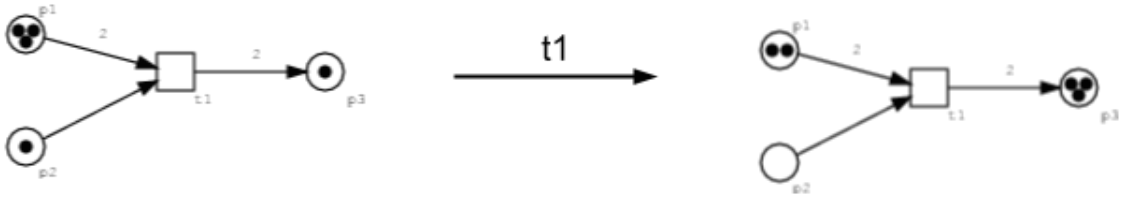


Fig. 2.2 The token game for basic Petri Nets.

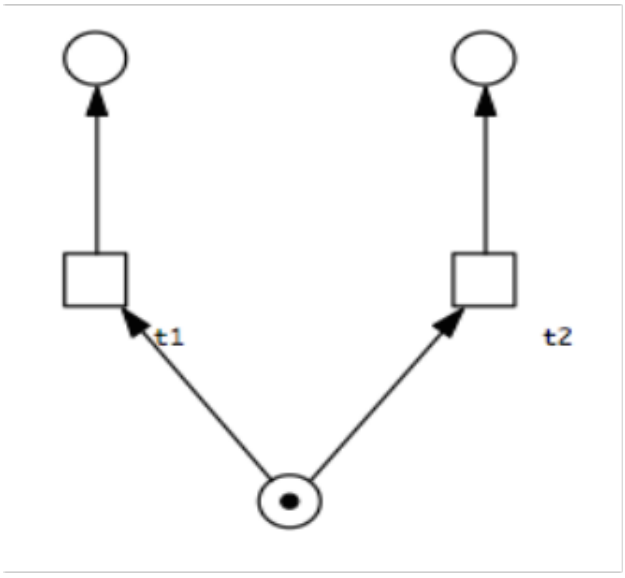


Fig. 2.3 Non-deterministic decisions encoded in Petri Net structure.

3. Choose transition t from set of enabled at random
4. Update state according to $\mathscr{M} = \mathscr{M} - pre(t) + post(t)$
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)

Non-determinism comes in the system from step 2 where we choose a transition to fire at random from a set of enabled ones. This way multi-option probabilistic decisions can be incorporated by having one or more transitions that share pre-places. From the example Figure 2.3 when the shared pre-place is sufficiently marked both transitions would be enabled. The token would have to make a non-deterministic decision between the two alternatives, t_1 and t_2, and flow to the corresponding post-place.

There is a very natural correspondence Petri Nets and biochemical systems. Every reaction is represented by a transition with the reactants becoming pre-places and the products post-places of that transition. The stoichiometry of a reaction is represented in the pre and post maps. A reaction like $2H + O \rightarrow H_2O$ will lead to a Petri Net with three places and one transition removing two tokens from the H places and 1 token from the O place and adding one token to the H_2O place. The tokens in each place indicate the amount of the corresponding species. Notice that despite the fact that there is an implicit ordering of events during the execution of the net, time is not explicitly modelled. Hence biological models captured with simple Petri Nets cannot capture the dynamic behaviour of the system. They have however been used for qualitative modelling of metabolic pathways using their structural and marking-dependent properties. The fact that they do not model time has another implication in that all reactions are equally likely to happen when they are enabled. In decision structures, like the one in Figure 2.3, all the available options are equally likely. This decision process is crucial in our use-case therefore we need more tight control, for example being able to set the probabilities of the outcomes. This makes basic Petri Net an unsuited language for our purposes so we used an extension, Stochastic Petri Nets, that is described in the next section.

2.1.2 Stochastic Petri Net extension

Stochastic Petri Nets explicitly model time by introducing a wait time for each transition. Wait times are random variables that follow an exponential distribution with the rate of the transition. Stochastic Petri Nets are a 5-tuple $(P, T, pre, post, \nu)$. Everything is defined as before with the only addition being the ν that is a map between transitions and rate functions. A rate function for a transition t is a possibly marking dependent function that returns the rate $\lambda(M)_t$ for the wait time of that transition. Now the operational semantics also change slightly. When a transition t is enabled a wait time is sampled from a negative exponential with rate $\lambda(M)_t$ which has to lapse before the transition fires. Therefore time is explicitly captured. When more than one transition is enabled a wait time is sampled from all of them and the one with the least wait time gets to fire. This suits our modelling requirements for assigning probabilities to outcomes of non-deterministic decisions. For a decision structure like the one in Figure 2.3 we can set the likelihoods of the two outcomes t_1 and t_2 by adjusting their relative rates λ_{t_1} and λ_{t_2}.

2.2 Stochastic Pi-calculus

Pi-calculus is a process algebra designed to capture decentralised computation as the action of independent agents/processes which communicate with each other. Stochastic pi-calculus is an extension to the original formulation to capture stochasticity and time similar to the Stochastic Petri net extension for basic Petri Nets. Processes can do a sequence of actions which are composed with the '.'(dot) operator. Each action can be a receive or send action. Communication between processes is done with message-passing through send/receive actions between procesess. For example consider a process A:

$$A = !a.?b.P$$

The behaviour of A is a chain of events composed with the . operator. First A offers to send a message on channel `a`, then it offers to receive a message on channel `b` and then it evolves into a process P. Communication happens between two processes that offer to receive and send on the same channel. Consider the following evolution of a system consising of two parallel processes(composed with the 'l' operator):

$$(!a.?b.P)|(?a.Q) \Rightarrow (?b.P)|Q$$

Once the communication via channel `a` happens, both processes move to the next action in their sequence. Communication over a channel has the ability to pass messages between the two processes that can be further used in subsequent actions. This message passing feature is mainly used to pass private channels, which can be created by processes, for private communication.Non-determinism is added to the system by having a choice operator '+'. This operator can join sequence of actions together. For example consider an extension to the A process we defined above:

$$A = !a.?b.P + ?c.Q$$

Now A has a choice between two series of actions. If the first action for both series can occur then one of them is chosen non-deterministically. This is equivalent to the decision structure we have seen in the basic Petri Nets. As was the case with the Stochastic Petri Net extension, Stochastic pi-calculus adds a wait time or communication delay for every channel which is distributed exponentially with a rate specific to that channel. If more than one communication action can happen the one with the least delay in its channel will occur. This mechanism again allows us to encode probabilities of options for decisions.

In this study we used SPiM which is a programming language based on stochastic pi-

calculus with a well-degined syntax that one can use to program a biological model. It also, and more importantly, has a graphical notation which corresponds to the textual notation. The language operators change slightly and the language is more like a traditional programming language. To see the correspondence, two parallel process in pi-calculus, $(!a.?b.P+?c.Q)|(?a.Q)$, become the following SPiM program:

```
new a@0.3: chan
new b@0.5: chan
new c@0.4: chan

let A()=
(
    do !a; ?b; P()
    or ?c; Q()
)
and P()=()
and Q()=()
and B() = ?a;Q()

run (A()|B())
```

Channels can be declared with the `new` keyword and processes similar to how functions are traditionally declared in programming languages. Processes can also take arguments which is one difference between SPiM and traditional stochastic pi-calculus. The '+' operator for choice becomes a 'do or' statement, the dot operator becomes ';'. Another addition to the notation is the 'delay' operator which can be used to delay an action with an exponentially distributed wait time based on the rate of that delay:

```
let A = delay@0.3;!a;()
```

The graphical notation for the above system is given in Figure 2.4. We can think of this notation as representing state transitions again but this time between different states of one components/process. This notation emphasises the fact that the behaviour of each component is defined independently. A node stands for a process state and there is an edge between two nodes labelled with an action if it is possible to go from one process state to the other through that action. A new state of a process could a new process. Choice is represented by having two or more outgoing edges from a process state.

The processes and communications of pi-calculus are very generic and can be used for all systems of interacting agents that communicate. In biochemical systems molecules

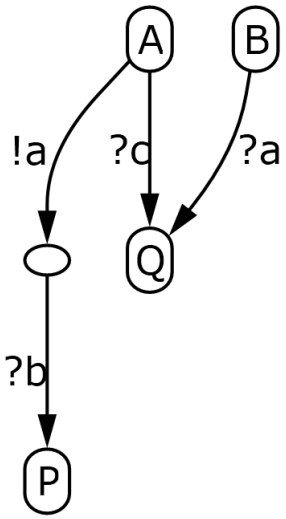


Fig. 2.4 Graphical notation for SPiM

become processes and their communications are the reactions. Defining a pi-calculus model is not a straight translation from a series of reactions because reactions are broken down to a send and receive part which are then defined as actions for the reactant proesses. The products of the reaction are defined as the reactant processes evolving to a new process after the communication. Despite the fact that the model is not defined in terms of reactions but in terms of species, the evolution of a pi-calculus model is still reaction-centric since the basic unit of computation is a communication event between two processes. Also the decision between reactions can be encoded nicely with the choice operator. These characteristics meet our requirements for a language to capture a reaction-centric view of the system. Their different approach in system definition along with their graphical notation makes for an interesting alternative to Petri Nets which are closer to the standard chemistry view of the system as series of reactions.

Chapter 3

Work

The aim of this study was to capture the iterative elongation process of even-chain saturated Fatty Acids. Fatty Acids are carboxylic acids with long hydrocarbon side groups (Figure 3.1) and they are usually characterised by the number of carbons in these side-groups, for example a FA with a chain of 6 carbons is usually denoted as C_6. The side-chains of FAs grow by successive C_2 concatenations. This concatenation process takes places at the FA biosynthesis pathway in the cytosol for chain lengths up to C_{18}. Further elongation takes places at the FA elongation pathway in the Endoplasmatic Reticulum(ER).

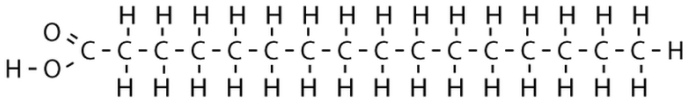


Fig. 3.1 The structure of Fatty Acids(FA) which consists of a long hydrocarbon side-group. The length of this side-group charactersises the FA.

In this study we are interested in the elongation process in its entirety so our models capture the combined effect of the two pathways responsible for it, FA biosynthesis *and* FA elongation. In this section I first introduce the basic model and the assumptions made to simplify it, the tuning of that basic model based on FA measurements, a version of the model in stochastic pi-calculus and finally the extension of the model with the introduction of some control mechanisms.

3.1 Basic model

3.1.1 Petri Net implementation

FA biosynthesis starts with Acetyl-CoA which is being converted to Malonyl-CoA. A reaction between Malonyl-CoA and Acetyl-CoA starts off the first C_4 FA product and a CO_2 molecule. After that there are successive C_2 concatenations to elongate the FA product with each elongation step that requires 4 reactions, requiring another Malonyl-CoA(Figure 3.2). The full pathway from KEGG can be seen in Figure 3.3. The elongation pathway located in ER is similar in nature but the intermediaries are CoAs instead of ACPs (Figure 3.4). ACPs (Acyl Carrier Proteins) is where the growing FA chain is bound during elongation.

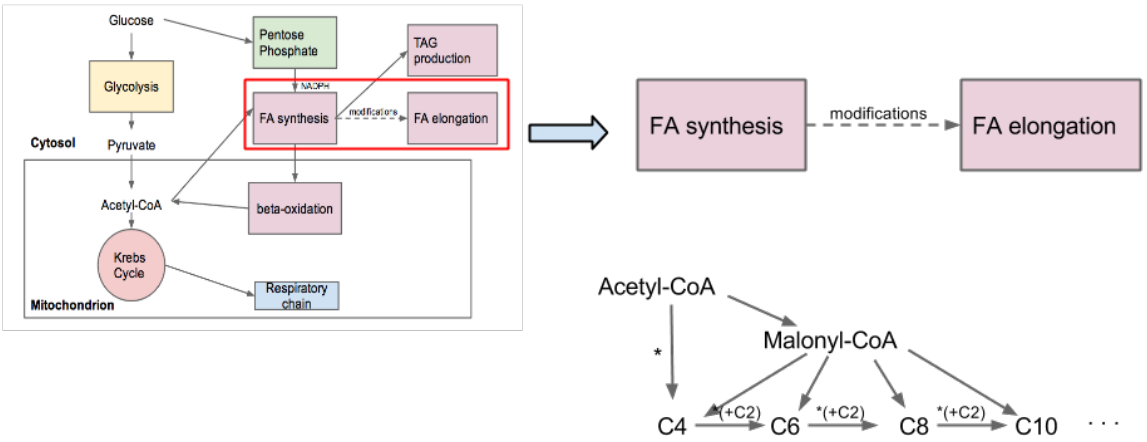


Fig. 3.2 FA metabolism in relation to other pathways.

Because of the natural correspondence between biochemical reactions as they are annotated in the KEGG static model to Petri Net model constructs a straight translation between the two is straightforward. All the pathway reactions become transitions with pre-places the reactants and post-places for the products. Here we made our first assumptions by omitting the enzymes participating as catalysts from the reactions. Information about the enzymes is not so important for our purposes since we are interested in the numbers of molecules of the metabolites involved in the process. Information about the enzymes can be incorporated in the transition rates. The net can be animated using its formal operational semantics and as transitions/reactions occur tokens/metabolites move through the net. The FA products are represented as sinks, places that are not pre-places to any transition, so once a token reaches one of these sinks it is trapped. That way the probabilistic iterative nature of the process is captured completely, an intermediate of the process can either remain at that length or go on to form longer FAs. The sinks are therefore the outputs of the stochastic process with the

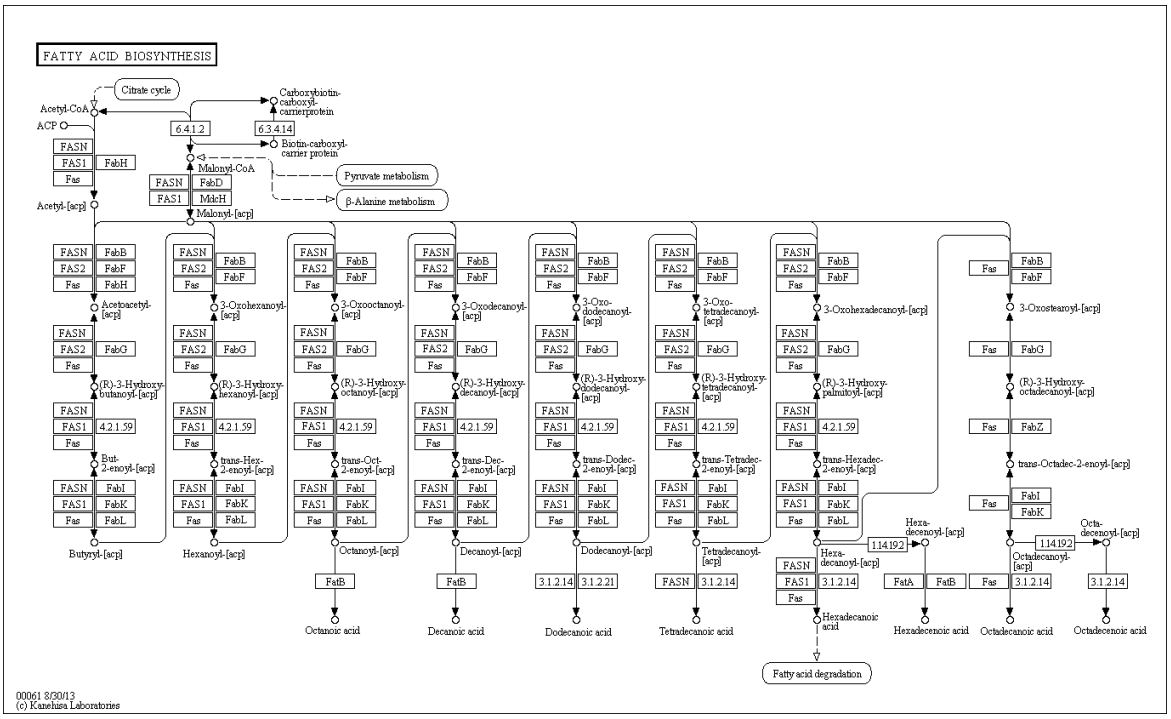


Fig. 3.3 FA biosynthesis in the cytosol. Notice the successive concatenations. Each concatenation takes 4 reactions.

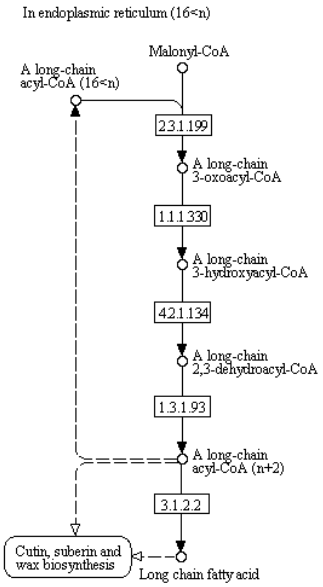


Fig. 3.4 General form of FA elongation in ER. Notice that the intermediate products are CoAs instead of ACPs as in biosynthesis.

inputs being the places that do not act as post-places for any transition. Starting with some finite number of molecules at the inputs, these will be consumed throughout the process until we reach a dead-state where no further transitions are enabled. The translation from the KEGG pathway model to the Petri Net model was done manually with the SNOOPY [13]tool which allows you to draw a net, play the token game for basic nets to observe its behaviour, and get the dynamic behaviour of the metabolites over the integration time.

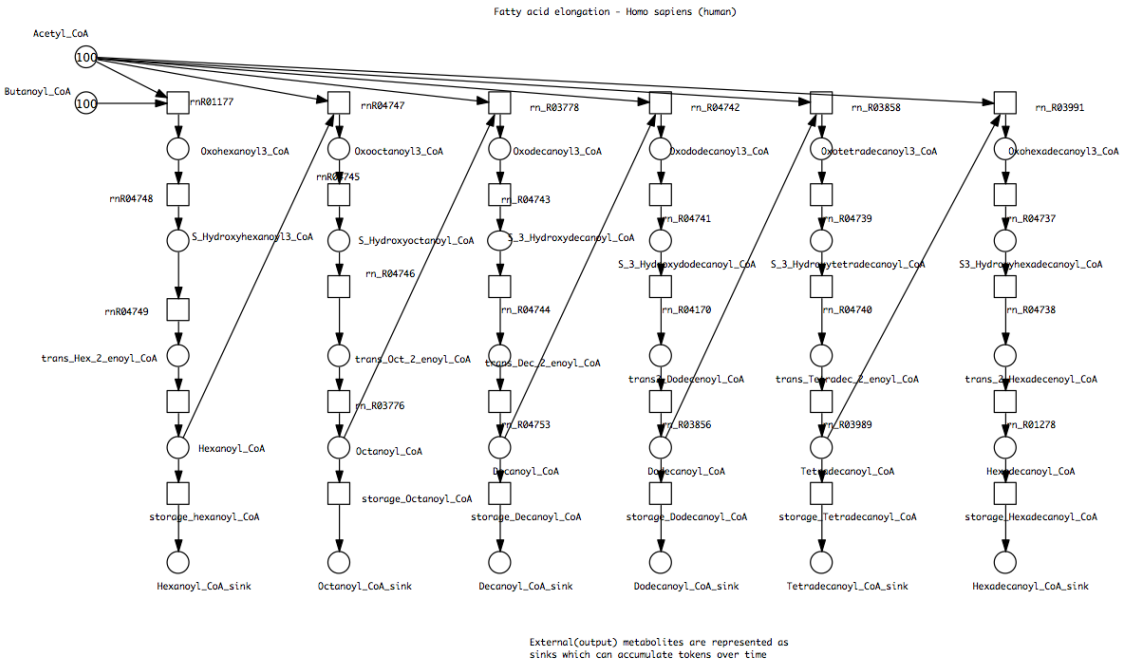


Fig. 3.5

A direct translation from KEGG is certainly useful for capturing a low-level biochemical view of the system. The model can offer an even more fine-control view of the system than the model presented in Figure 3.5 by including the enzymes in the reactions and information about their activity to the reaction substrates. However including too many details in the model can sometimes hide the true aspect of the system the modeller is trying to understand and analyse. Here we are interested in the probabilistic, non-deterministic nature of the iterative FA elongation process which happens as this series of C_2 concatenation steps in FA elongation and biosynthesis pathways in ER and cytosol respectively. The outputs we are particularly interested to see are the numbers and proportions of FAs at different lengths. In Petri Nets model language these are the number and more importantly the proportions of molecules/tokens that end up in the sink places at the stop of the model execution. Because the process is inherently iterative and probabilistic these numbers will be different at the end

of each model execution and in the limit the average number of metabolites will be the same as the equivalent deterministic model. This is in contrast with most dynamic modelling approaches as we are not interested in the time traces but rather only at the end result of this stochastic process and matching with the experimental dat awe have afterwards. We can think of the Petri Net as a magic box governed by some probabilities(that we can tune) where we throw some inputs and according to these probabilities some output will come out at the other end in the form of proportions of FAs at different lengths. The probabilities are just the control parameters that guide the operation of the magic box.

Having outlined our modelling goals in the previous section we can now make some assumptions that will guide our design of the simplified synthesis/elongation model that will be our basic model throughout this study. First of all the model will capture the combined effect of both pathways. The two pathways are in different compartments of the cell, one in cytosol and one in ER, but we will ignore the transportation of FAs from cytosol to ER and we will assume that the two process are sequential so a C_{18} for example can be elongated directly to a C_{20} while in reality it would have to first be transported to the ER. Since we also take this view of the net model as a control box turning input signal through a series of steps to an output signal, we can safely squeeze the four step reaction chain needed for each C_2 concatenation step to a single step reaction and at the same time consider only the forward direction reactions. Also, since we no longer consider exact chemical reactions we will consider constant values for the the reactions rate functions and we will treat those as probabilities that will govern the non-deterministic decisions during the execution of the model. Finally we also decided to stop the elongation process at C_{22}. Therefore the sinks will only include products C_{12} - C_{22}. FAs with lengths 4-10 will be included but no output will be considered. These decisions are supported by the output from the real experimental data available for tuning the model (see Results section).

The basic model was built with SNOOPY [13] using the previous conditions and assumptions. This basic model that will be the basis for our work in this study can be seen in Figure 3.6. The model is similar in nature to the more full model presented previously but the concatenation steps are represented as single transitions and the elongation process goes up to C_{22} because the elongation process in the ER is also included. Initially, I consider as simple model with two input points Malonyl-CoA and Acetyl-CoA whereas in reality the only precursor of FA biosynthesis is Acetyl-CoA and the 2 input points of the pathway are products coming from Acetyl-CoA. The model as presented in Figure 3.6 starts with inputs: 300 Malonyl-CoA and 100 Acetyl-CoA molecules. The model can then be set in motion with SNOOPY until it reaches a dead state which corresponds to the point where the system runs out of input metabolites to 'fuel' further reactions. The output of the system is the

number of tokens accumulated at each sink or FA of different lengths.

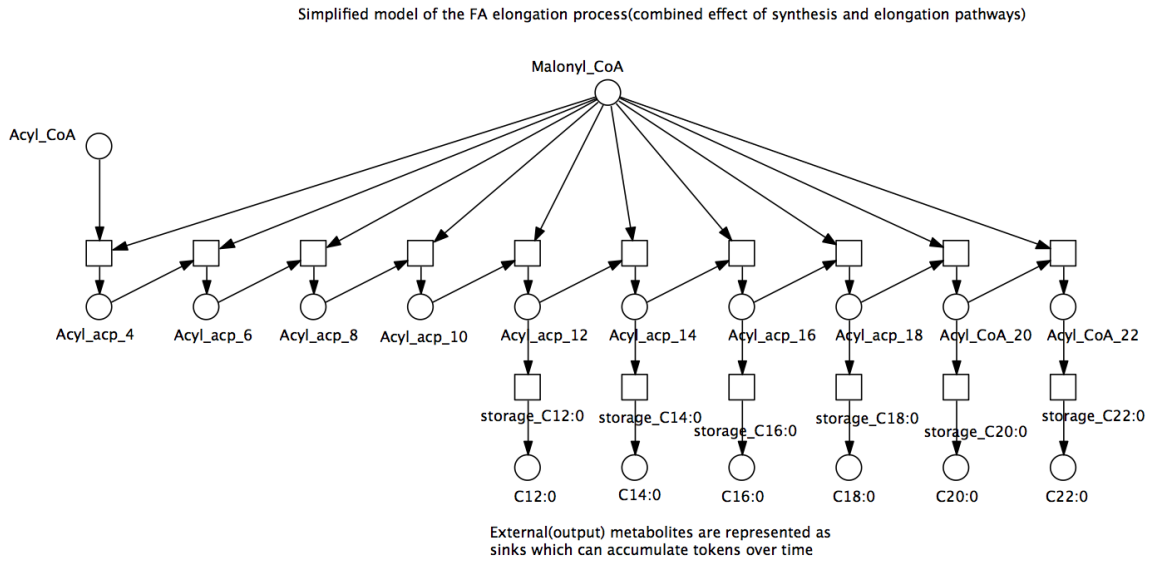


Fig. 3.6 Petri Net implementation of the basic model as described in the main text

Observing the execution pattern of the basic net model we notice an execution pattern which leads to an elegant view of the system as a series of binary probabilistic decisions or Bernoulli trials. At the start of the model execution only the initial transition producing the first product C_4 is enabled. If we consider that after this transition it will only fire again after its initial product has reached a sink then at each iteration only one token is travelling through the net. This token goes through an iteration of repeating the same binary decision: to either stay at the current length or continue to form a longer FA. More formally as the token gets transformed to an intermediate product there are only two transitions enabled: the transition taking it to the next longer intermediate and the transition taking to be stored at its current length(Figure 3.7). Let these two transitions be t_1 and t_2 respectively. According to the operational semantics of Stochatic Petri Nets (see Section 2.1.2) a wait time is sampled for each of the enabled transitions from a negative exponential distribution with rate the value of the rate function of the transition. In this case the rate function of the two transitions are constants (see assumptions) λ_{t_1} and λ_{t_2}. Since we know that one of the reactions will fire we can say that the firing probabilities of the two transitions or the probabilities of the token's decisions are:

$$P(staying) = P(t_1) = \frac{\lambda_{t_1}}{\lambda_{t_1} + \lambda_{t_2}}$$
$$P(continuing) = P(t_2) = \frac{\lambda_{t_2}}{\lambda_{t_1} + \lambda_{t_2}} = 1 - P(t_1)$$

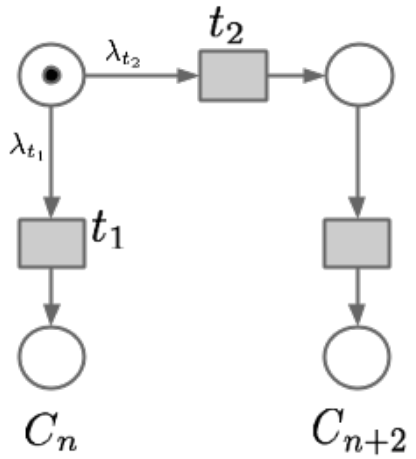


Fig. 3.7 Binary decision at a particular point in the execution.

In other words this is a Bernoulli trial with probability of success $p = P(t_1)$. Since we only considered the outputs at the end of the execution and not the timeframe of the process from start to finish, then only the ratio of these two successive transitions rates is important and not their absolute numbers. The entire journey of a token through the network from the initial C_4 product until it reaches a sink and gets stored, can be thought of a series of Bernoulli trials(Figure 3.8). The entire stochastic process can be thought of a sequence of successive realisations of this series of trials with tokens travelling through the net one after the other with each one going through the series of decisions or Bernoulli trials. The initial assumption that only one token travels through the net at each time can in fact be dropped since the output of the net only depends on the ratios of the pair of transitions representing each binary decision. We consider this assumption, because it is intuitive and reduces the problem to a successive realisations of chains of Bernoulli trials. It is also interesting to note how the binary decision corresponds to a conflict between two events that share pre-conditions which create non-determinsim through a race condition between

the two depdendent events. The 'conflict' that represents this decision process is captured inherently in the structure and semantics of Petri Nets.

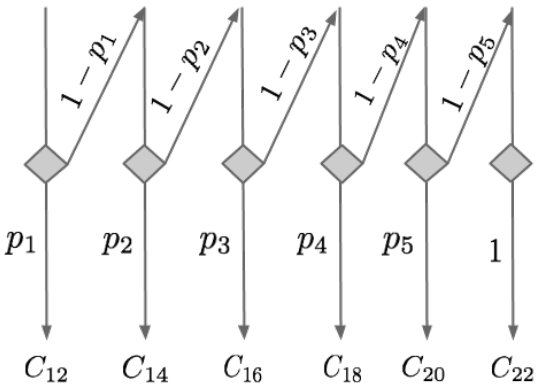


Fig. 3.8 The entire stochastic process a series of Bernoulli trials.

Each realisation of the stochastic process described by the Petri Net model is itself a series of successive realisations of series of Bernoulli trials that have different outputs. While SNOOPY is a great tool for the visual side of the process, creating the model and animating it, in order to be able to execute the model many times and with many orders of multiple iterations to get output profiles we needed a programmatic solution. We wrote code in Python that can read a model, execute it once or a number of times, and write out the results. For the description of the model I used the PySCeS model description language for biochemical systems which is aimed at the Python language [24]. The reactions are defined in the standard chemical notation and since this corresponds exactly to Petri Net transitions it was easy to load the description into an appropriate Petri Net representation (Figure 3.9).

3.1.2 Model parameters

In Section 3.1.1 we described the building of the basic model based on the goals and assumptions of the study. The elongation aspect that we are interested in was reduced down to a sequence of successive series of Bernoulli trials to determine the length that a FA will stop its elongation at. The interesting parameters of the model, the ones that will affect the output, are the success probabilities for the stay-continue decisions for FA intermediaries with lengths $[C_{12}, C_{22}]$ since they are ones that have sinks. In this section two methods are presented for the identification of these parameters for tuning the model so that its execution is biologically valid. They both rely on having experimental real dataset of realative proportions of FAs. These data can be seen as different realisations of the stochastic process

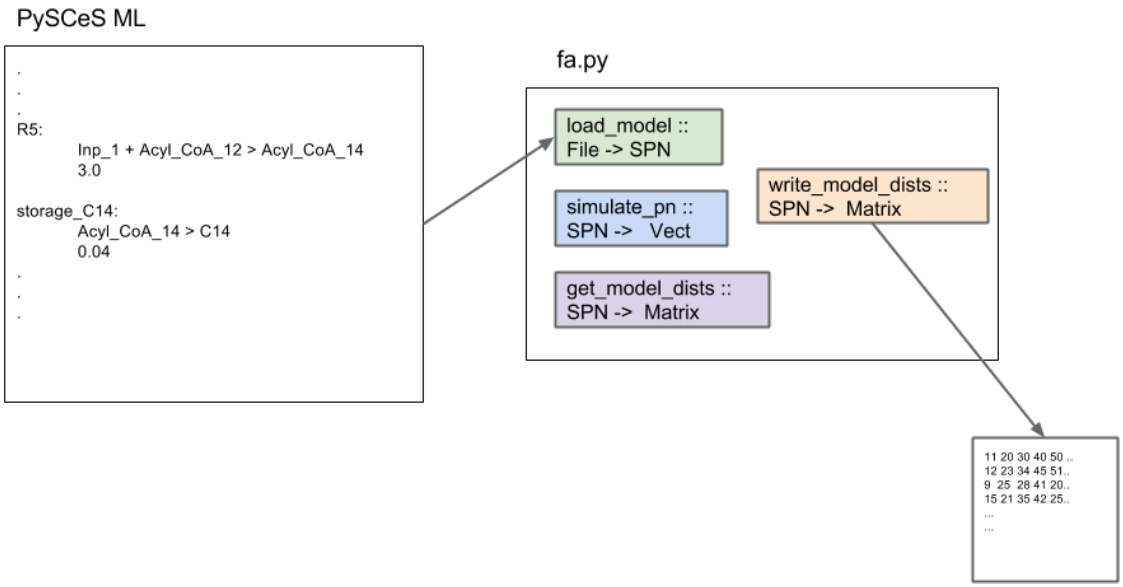


Fig. 3.9 Structure of the code and workflow.

that acts in nature. The attempt is then to tune our model to be as close to the real model as possible by trying to tune the proportions of the outputs of our model to match the proportions of the outputs by the in the experimental data. This is classical Machine Learning formulation of the inverse problem(Figure 3.10). In this section I first present a method to get point estimates and then profiles for the success probabilities parameters of the series of Bernoulli trials representing the chain of decisions for a token travelling through the net. The two methods presented here are used to analyse a real dataset in Section 3.1.3.

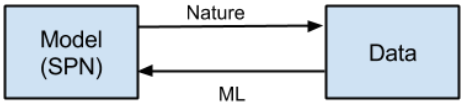


Fig. 3.10 ML inverse problem formulation

Point estimates

The experimental data are more formally a matrix D with an element D_{ik} being the number of tokens accumulated in the k-th FA in the i-th sample. Since we have multiple realisations

of Bernoulli trials, the success probabilities can also be thought of as success probabilities in binomial distributions. The Maximum Likelihood point estimator for binomial success probabilities is given for the i-th decision corresponding to the i-th FA,

$$\hat{p}_i = \frac{k_i}{n_i}$$

where k_i is the number of successes and n_i is the number of trials. If we extract information about the number of successes and number of trials for each binary decision in the process from data that are different realisation of the process, then we can straightforwardly compute the success probabilities.

If we a take a specific sample (a row from data matrix D) then the number of successes for an FA is simply the number of tokens of this FA in the sample since those are the tokens that when faced with the decision of whether to get stored as that FA or continue to form longer FAs they chose the former. Success and failure are arbitrarily assigned to staying and continuing respectively. The number of trials is the number of all the tokens that reached that FA intermediary, both the ones that stayed *and* the ones that continued. Since this is an iterative process the ones that continued are the number of tokens that made it to the sinks after the FA(the ones on the right as we look at the Petri Net model picture, Figure 3.6). To illustrate this consider a sample which could be a row from our data matrix D: $\{C_{12} = 5, C_{14} = 10, C_{16} = 34, C_{18} = 23, C_{20} = 3, C_{22} = 5\}$. The ML estimator for the i-th FA product in that sample will be:

$$\hat{p}_i = \frac{C_i}{\sum_{n \geq i} C_n}$$

The success probabilities p_1 to p_6 for the above sample, with p_1 corresponding to the first binary decision and C_{12}, p_s to the second decision and C_{14} and so on, are:

$$\begin{aligned}
\hat{p_1} &= 5/(5+10+34+23+3+5) = 0.0625 \\
\hat{p_2} &= 10/(10+34+23+3+5) = 0.1333 \\
\hat{p_3} &= 34/(34+23+3+5) = 0.523 \\
\hat{p_4} &= 23/(23+3+5) = 0.742 \\
\hat{p_5} &= 3/(3+5) = 0.375 \\
\hat{p_6} &= 1
\end{aligned}$$

This is for one sample but if all the samples are from the same population, we can combine the successes/trials data from all samples and generalise the ML success probability

estimation for the i-th FA product as:

$$\hat{p}_i = \frac{\sum_k D_{ki}}{\sum_k \sum_{n \geq i} D_{kn}}$$

We assume that the columns of data matrix D are ordered according to the order of FAs in the system.

At the end, the relative proportions of the transitions are important not the timeframe of generating the iterations in our process. We can therefore use the calculated success probabilities as the rates of the transitions related to the binary decision(s). So for example if $\hat{p_3} \approx 0.82$ then the rates of the transitions corresponding to the binary decision for the 3rd FA, C_{16}, will be: rate of transition for C_{16} storage (success) 0.82 and rate of transition for C_{18} formation (failure) $0.18(1 - P(success))$.

Profiles

In the previous section we presented a method to get point estimates for the success probabilities guiding the binary decisions taking place during the elongation process. We did that by considering the success probabilities that maximise the likelihood function $L(D|p_i)$ for the i-th FA and data D being the number of successes and number of trials. In this section I will present a likelihood function for the entire process which comes naturally by thinking of the process as before as a sequence of series of Bernoulli trials. We then use this likelihood function to present a method for calculating the profiles of the parameters.

The data from one of the samples can be seen as one realisation of the stochastic process with the numbers of tokens at each FA being the outputs of the series of Bernoulli trials that make up one realisation of the entire process. Each series of Bernoulli trials describes the journey of one token through the net which ends when it finds a sink. If for example the token ends up at the third sink that means that the outcome of the series of Bernoulli trials was: fail, fail, success. Therefore the data consisting of the number of tokens that ended up at each FA gives us a history of the series of trials that happened during that realisation of the process. Consider the following sample $\{C_{12} = 5, C_{14} = 10, C_{16} = 34, C_{18} = 23, C_{20} = 3, C_{22} = 5\}$ which is one realisation of the process from the start until reaching a dead-state. The numbers tell us the 'fate' of each of the tokens that went through the net during that realisation of the process, 5 tokens made the decision to stay at the first trial- [success]- , 10 tokens made a decision to continue on the first decision and then to stay on the second decision- [fail, sucess]-, 34 tokens continued on the first two decision and stayed on the third- [fail, fail, success], and so on. We can therefore calculate the probabilities for each

FA product sinks as follows:

$$
\begin{aligned}
P(C_{12}) &= p_1 \\
P(C_{14}) &= (1-p_1)p_2 \\
P(C_{16}) &= (1-p_1)(1-p_2)p_3 \\
P(C_{18}) &= (1-p_1)(1-p_2)(1-p_3)p_4 \\
P(C_{20}) &= (1-p_1)(1-p_2)(1-p_3)(1-p_4)p_5 \\
P(C_{22}) &= (1-p_1)(1-p_2)(1-p_3)(1-p_4)(1-p_5)p_6
\end{aligned}
\tag{3.1}
$$

The entire process can then be seen as a multinomial draw with 6 outcomes, $\{C_{12}, C_{14}, C_{16}, C_{18}, C_{20}, C_{22}\}$, with their respective probabilities given above and with the number of trials being the number of tokens that ended up in sinks for that particular realisation of the process. This gives the likelihood function of the data given the success probabilities as probability mass function of the multinomial distribution:

$$L(D|\theta) = f(C_{12}, \ldots, C_{22}; n; P(C_{12}), \ldots, P(C_{22})\}) =$$

Using the likelihood function for the entire process we can find the posterior of the initial success probabilities p_1 to p_6,

$$P(\{p_i\}|D) \sim L(D|\{p_i\})P(\{p_i\})$$

where the data D is simply the outputs of one realisation of the process or the counts at each sink and $P(\{p_i\})$ are the priors of the parameters. If we assume uniformative priors the above parameter posterior simply becomes: $P(\{p_i\}|D) \sim L(D|\{p_i\})$. Since the parameters are not exactly the parameters of the multinomial but they are only related through Equations 3.1 an analytic solutions is not very attractive but since it is still a small problem we can resort to a simple simulation scheme to sample from the required posterior. I used the standard Metropolis-Hastings Markov Chain Monte Carlo method to sample from the posterior as given above starting from a random point in parameter space and a symmetric normal proposal distribution with mean the previous accepted guess and an arbitrary chosen variance.

Parameter	Value
p_1	0.0003000656
p_2	0.0400323889
p_3	0.8235217126
p_4	0.9888339514
p_5	0.778597786
p_6	1.0

Table 3.1 Success probabilities estimated from the Control samples in the dataset.

3.1.3 Results

In this section we use the methods described previously to analyse and calculate the success probabilities from a real experimental dataset. The dataset comes from an experimental study for the effects of a treatment on the profiles of Fatty Acids on mice. FAs were extracted from brown adipose tissue from 91 mice subjected to 5 levels of treatment. The FAs in the samples include both the free FAs and FAs that were cleaved from more complex lipid products. The amount of the extracted lipid species was measured with Gas Chromatography and a Flame ionization detector that gave relative intensities of each product. Each sample contained measurements from 34 FAs products that were identifiable in the measurements. From those 34 only 6 correspond to the outputs of our process of interest, namely the even-chain saturated FAs C12:0, C14:0, C16:0, C18:0, C20:0, C22:0. This guided the model construction choices with sink outputs appearing at FAs with only these lengths and the entire process stopping at C22. The outputs of the net model are integer number of tokens accumulated at each of the sinks while the experimental data are relative intensities which are continuous numbers. In order to be able to compare with the data for the tuning of the parameters the experimental data were transformed from relative intensities to relative absolute numbers. A sample of relative intensities $\mathbf{s} = \{s_1, s_2, s_3, s_4, s_5, s_6\}$ became a transformed sample $\mathbf{ts} = \{ts_i | ts_i = \lfloor s_i / min(\mathbf{s}) \rfloor\}$. Since we are only interested in the relative proportions of the outputs going from relative intensities to relative numbers should not affect the outcome.

Point estimates of the success probabilities calculated with from the Control(treatment level 0) subset of the real dataset can be seen in Table 3.1. We then applied Baysian Hierarchical Clustering (BHC) to the samples in the dataset to investigate changes in the parameters in the different groups. The clustering algorithm gave back 4 clusters. The centroid of each of the 4 clusters was used to calculate point estimates of the parameters. The centroids for each cluster can be seen in Figure 3.11 and the calculated success probabilities for each cluster can be seen in Table 3.2.

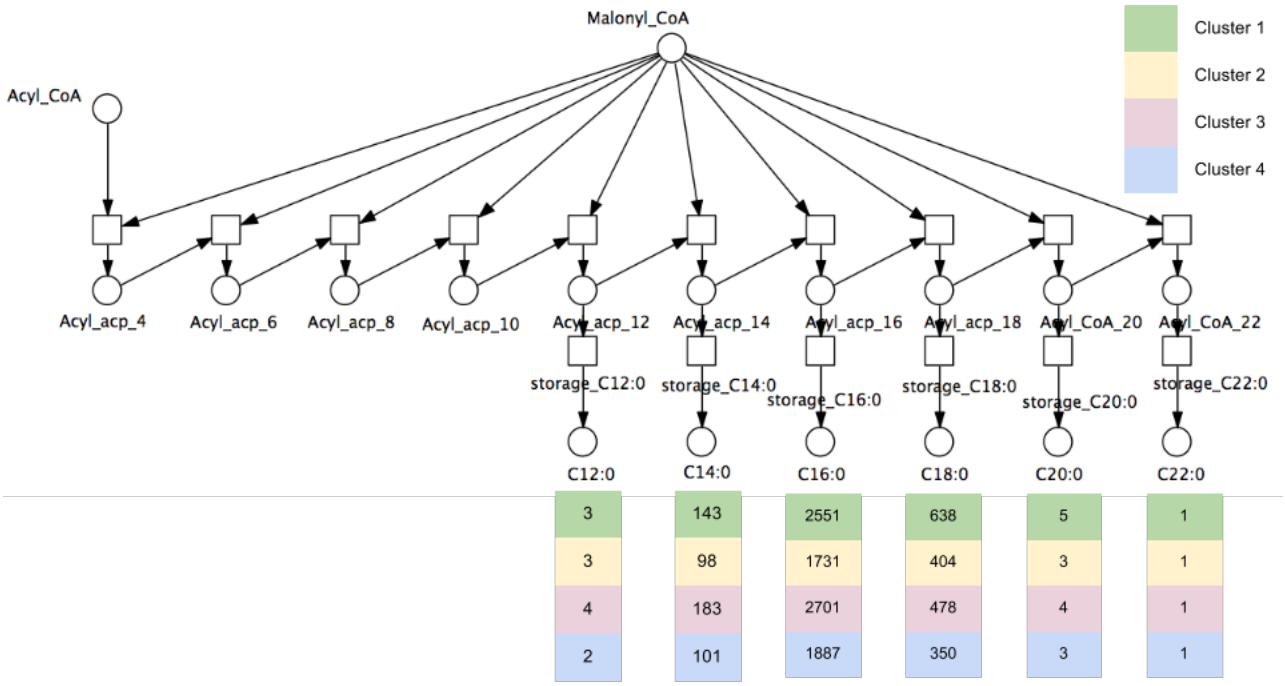


Fig. 3.11 Centroids for each of the four cluster discovered in the data with BHC.

Parameter	Cluster 1	Cluster 2	Cluster 3	Cluster 4
p_1	0.0008979348	0.0013392857	0.0011865915	0.0008532423
p_2	0.04284002	0.04380867	0.05435105	0.04312553
p_3	0.7984351	0.8092567	0.8483040	0.8420348
p_4	0.9906832	0.9901961	0.9896480	0.9887006
p_5	0.8333333	0.7500000	0.8000000	0.7500000

Table 3.2 Point estimates for the success probabilities for each cluster.

3.1.4 Stochastic π-calculus implementation

In this section I present a stochastic pi-calculus version of the model which is specifically written in its SPiM variant which extends the traditional pi-calculus first by introducing explicitly time which is needed for quantitive analysis of biochemical networks and also by adding operational semantics in terms of an Abstract Machine and a graphical representation language. The SPiM code is given for the processes making up the system to introduce further some of the concepts of the SPiM language.

The SPiM graphical representation of the model can be seen in Figure 3.12. Each component of that graph is a stochastic pi-calculus process and edges between them indicate that one one of them can evolve into the other. The nature of the model is different from the equivalent model given in the previous section. The evolution of every species represented by a stochastic pi-calculus process is defined independently. We start the computation by running 100 `Acyl_CoA` processes and 30 `Malonyl_CoA` processes:

```
run(100 of Malonyl_CoA() | 30 of Acyl_CoA())
```

The initial reaction of the system creating the C_4 product is represented as communication by channel `form6` between `Acyl_CoA` and `Malonyl_CoA` processes. Once the communication happens the `Acyl_CoA` process evolves into the `Acyl_CoA_4` process while `Malonyl_CoA` process goes to the empty process(gets consumed). After that all the processes corresponding FAs between C_6 and C_{10} only have one communication option, communicate via the appropriate channel with a `Malonyl_CoA` process and then evolve into a process corresponding to a longer FA. For example process `Acyl_CoA_8`:

```
and Acyl_CoA_8() = ?form10; Acyl_CoA_10()
```

After that all the processes corresponding to intermediaries for longer than C_{10} FAs have two communication options corresponding to the familiar stay-continue stochastic decision that we have seen in the previous section, they can either communicate with a `Malonyl_CoA` process and evolve into a process corresponding to a longer FA while the `Malonyl_CoA` process terminates or do a delay and evolve into their corresponding FA product. The FA products which are modelled as empty processes could have been ignored since they do not interact further with any other processes but they were kept there intentionally to keep track of their numbers.

```
let Malonyl_CoA() =
(
do !form6; ()
```

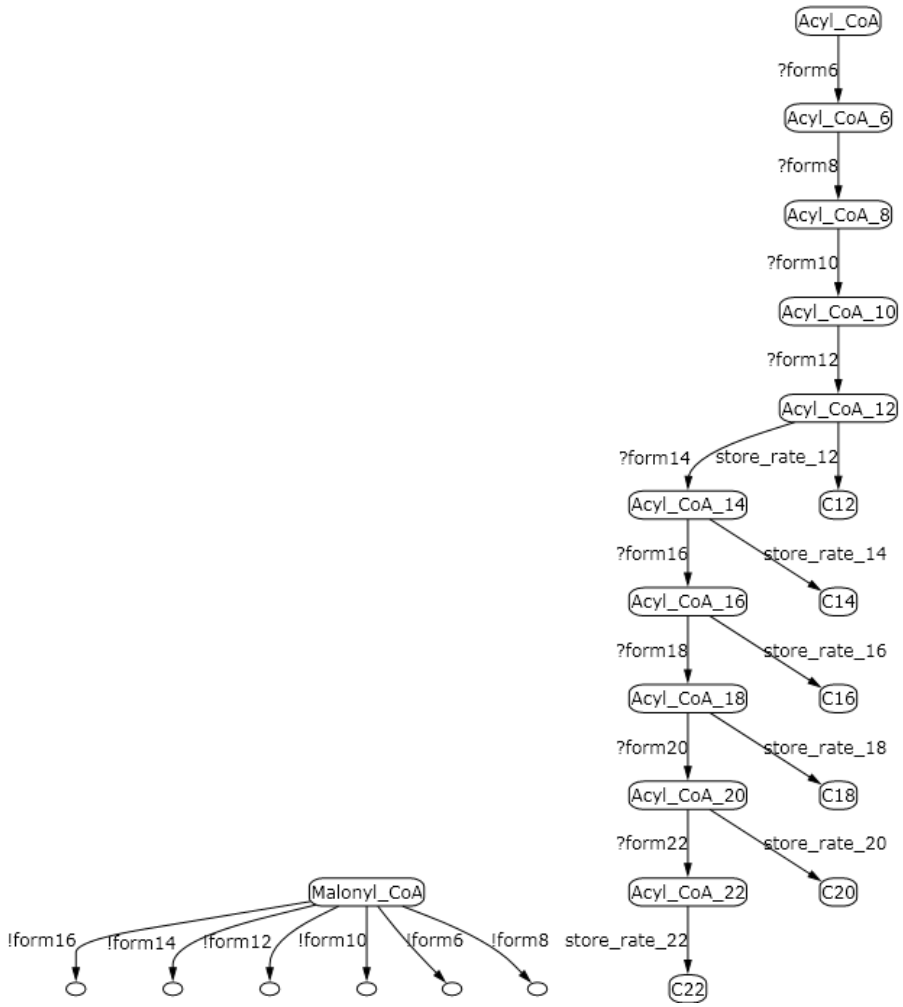


Fig. 3.12 SPiM graph

```
or !form8; ()
or !form10; ()
or !form12; ()
or !form14; ()
or !form16; ()
)
and Acyl_CoA() = ?form6; Acyl_CoA_6()
and Acyl_CoA_6() = ?form8; Acyl_CoA_8()
and Acyl_CoA_8() = ?form10; Acyl_CoA_10()
and Acyl_CoA_10() = ?form12; Acyl_CoA_12()
and Acyl_CoA_12() =
(
do ?form14; Acyl_CoA_14()
or delay@store_rate_12; C12()
)
and Acyl_CoA_14() =
(
do ?form16; Acyl_CoA_16()
or delay@store_rate_14; C14()
)
and Acyl_CoA_16() =
(
do ?form18; Acyl_CoA_18()
or delay@store_rate_16; C16()
)
and Acyl_CoA_18() =
(
do ?form20; Acyl_CoA_20()
or delay@store_rate_18; C18()
)
and Acyl_CoA_20() =
(
do ?form22; Acyl_CoA_22()
or delay@store_rate_20; C20()
)
and Acyl_CoA_22() = delay@store_rate_22; C22()
```

```
and C12() = ()
and C14() = ()
and C16() = ()
and C18() = ()
and C20() = ()
and C22() = ()

run(100 of Malonyl_CoA() | 30 of Acyl_CoA())
```

3.2 Extended model of FA synthesis

In the previous section I presented Petri Net and stochastic pi-calculus implementations of a simplified view of FA elongation/synthesis. In this section I will present a Petri net implementation of an extended model which captures one of the main control regulatory mechanisms between FA biosynthesis and other general metabolic machinery of the cell. The first and commited step in the FA biosynthesis pathway is the conversion of acetyl-CoA to Malonyl-CoA through the action of acetyl-CoA carboxylase. Acetyl-CoA carboxylase is inactivated by AMPK protein kinase in mammals. In the well-fed state AMPK is itself inactivated which leads to an increase in the flow of Acetyl-CoA to Malonyl-CoA and since this is an irreversible reaction to the FA biosynthesis pathway. Also in conditions of high glucose concentration isocitrate dehydrogenase is inhibited so Acetyl-CoA only does the first step of the Krebs cycle when Citrate is produced. Since Citrate cannot go any further in the cycle it starts to accumulate until it starts to diffuse from the mitochondrion into the cytosol where it becomes Acetyl-CoA again and acts as precursor for FA biosynthesis. The net effect of these two factors is that more Acetyl-CoA will flow towards the FA biosynthesis than go towards the Kreb cycle and the Respiratory chain to produce energy because the immediate energy requirements are low (high ATP).

Nielsen [23] experimentally confirmed that this regulatory mechanism is conserved in yeast through the action of Snf1 which the yeast analog of the mammalian AMPK and they also produced Flux Balance models to capture the change in flows in the system throught the action of protein kinase Snf1. Here we provide a Petri Net implementation of the above regulatory scheme as an extension to the basic model we presented in the previous section.

3.2.1 Petri Net implementation

The Petri Net implementation of the above mentioned regulatory mechanism as an extension to the basic model presented before can be seen in Figure 3.13. The net starts with an empty transition feeding into glucose which can be thought of as an interface to the environment. That could for example be used to model glucose intake from the diet. Then after that the action of every pathway is condensed into a single transition except FA biosynthesis/elongation for which we use the basic model developed previously. After the initial glucose intake transition, glucolysis is represented by a transition from Glucose to Pyruvate and then Acetyl-CoA which plays the central role in this regulatory mechanism. Acetyl-CoA has a decision of whether to flow towards FA biosynthesis represented by transitions to place `FAsyn` or flow towards the TCA cycle and further energy production which is represented by a transition to the `TCA` place (red box in Figure 3.13). The `TCA` place represents the combined effect of the TCA cycle and the respiratory chain which has as an ouput 30 molecules of ATP. ATP can then be consumed by taking part in a transition without any post-places. ATP consumption could for example be used to model various energy requirements. FA biosynthesis/elongation process is represented as before. The only difference is that it now only has a single input which then leads to the two inputs used in the basic version of the model (green box in Figure 3.13). The main regulatory mechanism comes by making ATP a pre-place for the Acetyl-CoA transition to `FAsyn` so that ATP levels can influence the rate of that transition and hence the flow towards FA biosynthesis.

3.2.2 Model Parameters

In this section I present a way to construct the rate functions for the transitions out of Acetyl-CoA in order to get the required regulatory behaviour based on ATP levels and a way to find the rate parameters for the transitions from the `FAsyn` place to the two main input points for the FA synthesis/elongation process.

Since we have not modelled enzymatic activity explicitly but only through the transition rates, the effect that ATP has on the enzyme responsible for the flow of Acetyl-CoA to FA synthesis will have to become an effect on the rate of the corresponding transition. In order to introduce that effect the rate function for the transition taking Acetyl-CoA to FA biosynthesis becomes an increasing function of ATP. That by itself is not enough since at high ATP levels the reaction rate not only needs to be high but crucially higher than the TCA transition rate in order to make the flow towards FA biosynthesis more favorable when Acetyl-CoA is faced with the TCA/FA biosynthsesis decision. So, again, the important fact here, since we do not consider the timeframe of the process, is the ratio between these

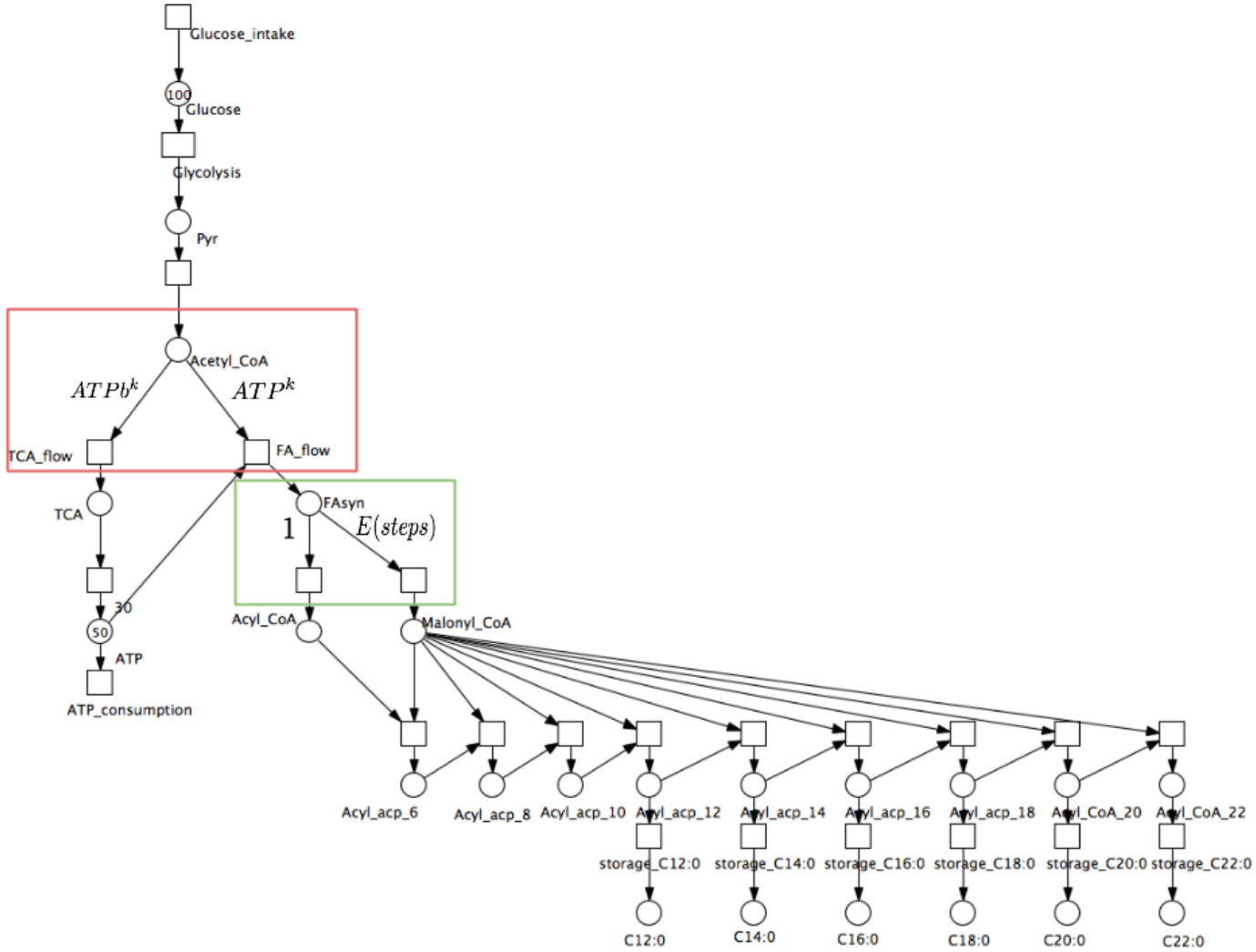


Fig. 3.13 An extended version of the model

parameters. In order to achieve that we can set up a basal rate for the two alternatives for a base level of ATP for which they are both equally likely to happen. Any deviations from that balanced level and the flow towards FA biosynthesis becomes more or less likely. Let us consider a constant base level of $ATPb$ ATP molecules and a variable ATP for the number of ATP molecules at any given time during the execution of the model. An example of rate functions that meet our requirements are: $\lambda_{TCA} = ATPb^k$ and $\lambda_{FAsyn} = ATP^k$. When $ATP = ATPb$ the transition rates are equal so the Acetyl-CoA flows with equal likelihood to TCA and FA biosynthesis. The strength of the response to deviations is then:

$$\left(\frac{ATP}{ATPb}\right)^k$$

We can therefore control the strength of the response by tuning the k parameter. For linear responses for example we can set $k = 1$ which will mean that a 2-fold increase in the ATP level from its base rate results in a doubling of the rate of the transition towards FA biosynthesis while the rate towards TCA remains constant.

For the transition rates from `FAsyn` we can make a simple stoichiometric argument to find the correct ratio between them that will guide the flow towards those two input points of the FA synthesis/elongation process. For every C_2 concatenation step we need one Malonyl-CoA so for example if the process does 7 steps to reach the C_{16} then we would need 7 Malonyl-CoAs and 1 Acetyl-CoA. Since it is a stochastic process however we do not know in advance how many steps each token will take through the net. We can however find the expectation of the number of steps:

$$E(steps) = \sum_{4 \leq i \leq 9} i \times P(i)$$

So on average we need $E(steps)$ Malonyl-CoAs for every one Acetyl-ACP. We can set the transition rates towards these two starting points to reflect that:

$$\frac{\lambda_{Malonyl\_CoA}}{\lambda_{Acetyl\_CoA}} = E(steps)$$

Chapter 4

Discussion

This section is divided into two logical parts. The first two sections are commentary and critique on firstly the created model and secondly the languages we used to capture the model. The last section takes the form of perspectives and future directions of both the language methodology and our developed model.

4.1 Simplified model and accuracy of description

In this work we created an abstract simplified reaction-centric model of the elongation process that takes place in the cell to grow the CH tails of FAs. The model we developed captures the defining high-level characteristics of this process accurately. The developed model allows us to observe the series of elongation steps and get distributions of outputs of FA products that can be compared with real experimental datasets as we have done in section Results. We have also shown how we can tune the parameters of the model in the presence of such datasets . The assumptions we made however for the contrusction of the model make it biochemically inaccurate. So while it might provide a high level view of the model in a conceptually clear way it does not reflect the actual biochemical process that takes place inside the cell. All the reaction rate functions were assumed to be constant and treated as probabilities without regarding enzymatic activity and numbers of molecules of the reactants. In fact all the concatenation steps of the process involve the activity of the same enzymes so the basal reaction rates (probability of reaction between two reactant molecules) should be the same for all elongation steps. There must be then some other underlying biochemical mechanism that gives rise to the probabilistic interpretation we provided through our model. Each concatenation step was also considered as a single reaction instead of the four that it actually takes. This assumption does not pose a big problem for the biochemical validity of the model if the stoichiometries and enzyme activities participating

in the reactions are considered carefully. The other big simplification was the combination of the two pathways acting together for the elongation of FAs, FA biosynthesis in the cytosol for FA lengths up to 16 and FA elongation in ER for longer FAs. The transportation between the cytsol was not considered.

In the extended model we considered the relation between different pathways in the metabolic machinery of the cell. We only wanted to investigate the communication between the interfaces of the different pathways so we represented pathways by one transition and explicitly only the metabolites that provide the communication interface with other pathways and most importantly the FA biosynthesis/elongation process. So for example the entire TCA cycle and respiratory chain responsible for the production of energy was represented as one transition with only ATP as output as that is the point of communication with the FA biosynthesis process. Also this interaction was represented as a single step while in reality it goes through the AMPK protein kinase. Again transportation between the compartments that in this case also inlude the mitochondrion is not included and Acetyl-CoA is represented by a single place. An interesting aspect of our model is that we can also capture the communication between the pathways and the environment for example by setting the rate for glucose intake or the rate of ATP consumption.

In a way our models do not completely fall into the bottom-up or top-down approaches to model construction. We started from some knowledge of the system to construct the model in the first place (top-down) but then experimental data guided some of the decisions in the model simplification. In the end the FA biosynthesis/elongation simplified model can provide accurate description and a mechanistic model for the reproduction of measurable quantities but not a mechanistic model for the underlying biochemical process. We still consider this a reaction-centric model though despite the fact that the Petri Net transitions do not exactly correspond to accurate biochemcical reactions. It is still reaction-centric because the main events of the model are the transformations of species from one form to the other.

4.2 Description languages

One of the project goals was also to investigate modelling methodology for a suitable language to describe the reaction-centric view of lipid metabolism. We particularly focused on Petri Nets that were the main language used but we also provided an alternative stochastic pi-calculus (SPiM) implementation of the basic model.

Petri Nets have an attractive graphical notation language that captures the main characteristics we identified in lipid metabolism that served as motivation for this reacton-centric

view: iterative conversion processes and probabilistic decisions. There is also a very natural correspondence between Petri Net transitions and reactions. The behaviour of the net is described with formal operational semantics that captures this reaction-centric view since the transition is the main component in the evolution of the computation. Because of the similarity with the traditional biochemical view of the system and the static diagrams used to described it we can think of Petri Nets as a dynamic executable version of the KEGG diagrams. Petri Nets are also expressive enough to allow us to move up or down the abstraction scale. For example in the extended net model we had places corresponding to entire pathways or places that were placeholders for the flow towards a specific pathway (`FAsyn` place in Figure 3.13). The formal operational semantics have a graphical equivalent in the form of the token game that can provide intuition in the workings of a pathway or network. There is a long tradition of using Petri Nets so there exist many tools to 'draw' nets and play the token game. In this work we mainly focused on quantitative analysis of stochastic Petri Nets but the dynamic picture is only part of the strength of Petri Nets. Their formal specification allows for qualititative static topological analyses that yield the same insides as standard techniques in the field like Elementary Nodes analysis. Another thing that we have not touched here but it is of particular interest especially as the models grow in size is the model checking ability we have with the use of Petri Nets. We can use model checking techniques for example to explore qualitative and quantitive static and dynamic properties of systems. Gilbert et al. [11] go as far as to provide a unifying framework for both qualitative and quantitative analysis of biochemical networks captured with the Petri Net language. It is interesting that we can use the same language to analyse and talk about features of systems that are usually captured with different approaches. One big drawback of Petri Nets is that they are monolithic. Models cannot be decomposed to smaller parts in order for example to abstract away non-important details of parts of the process that are not of immediate interest. We did this informally in the extended version of the model by defining entire pathways as transitions and explicitly only the metabolites that provided the link to other pathways. This is not captured formally though in the language. The inherent coupling with the chemical reaction view makes the constructed models grow linearly with the number of reactions. This is not a big problem for metabolic pathways because of the iterative chain of conversion that usually takes place. For larger systems however with combinatorial interactions between species there are scaling issues (See Future perspectives for some thoughts on decomposability and scalability).

Stochastic pi-calculus and specifically SPiM are more detached from the standard biochemical graphical notation. Each component of the system is defined indepedently in contrast to the Petri Nets where the main unit of definition are the transitions corresponding

to the reactions. Their operational reduction semantics are howerer defined in terms of communication between the components that correspond to the reactions. It is particularly suited to metabolism because metabolic pathways usually involve linear series of conversions. These linear series of conversion can be encoded with the sequential operator of the calculus that defined state changes of a component afer communication actions(reactions). The other characterstic we are interested in, decisions, is also captured nicely with the choice operator of the calculus that provides non-determinism. Another advantage of stochastic pi-calculus over Petri Nets is the decomposition that is inherent in the syntax/semantics. This coupled with the fact that the model size grows linearly with the number of species (instead of reactions) could make them more attractive for larger networks.

4.3 Future perspectives

4.3.1 Modelling and experimental validation

In this section we present some possible future directions for our contstructed models and experimental validation.

Using the already contstructed models as a scaffold we could expand them to include interactions with other pathways. In this work we concentrated on even-chain saturated FAs but these are not the only FA products. FAs can get modified after creation not only but the elongation pathway we considered but by adding double bonds at one (monounsatureate) or many (polyunsaturated) places in their CH tails. Then these FA products of different lengths and numbers of double bond combine as building blocks to form more complex lipid products like triacylglycerols (TAGs) and diacylglycerols (DAGs). DAGs and TAGs contain a headgroup and two or three FAs respectively. Following from our high level view of FA biosynthesis/elongation we could add probabilistic further modifications and probabilistic complex formation. Our current modelling methoodology is suited for this kind of systems. For example if we also had some experimental data about the frequencies of the triplets or pairs of FAs found in TAGs and DAGs respectively we could extend the model by making the current sinks of the system feed into transitions for complex formation using the frequencies to infer the probabilities. Perhaps we could start from yeast that only has mono-unsaturated FAs to limit the combinatorial space in complex formation [23]. Production and modelling the net forward direction is only part of the story however and the corresponding catabolic processes are equally important and the balance between them one of the major goals of the interplay and control between different parts of the metabolic machine. For example we could and should definitely add degradation of FAs. Finally as an

extension to our model with control we could add other control mechanisms. The control of of Acetyl-CoA flow to FA biosynthesis through ATP and AMPK is only one point of control. The production of C_{16} also inhibits the flow to prevent the accumulation of FAs for example.

In order for our models to be valid in their representation of some aspect of the natural world we need to be able to experimentally validate them by comparing their execution to real data. We have done that for our basic model for which we have output real experimental data. It would be interesting to get experimental data to confirm the control mechanism between ATP and Acetyl-CoA flow to FA formation. For example we could use ^{13}C metabolite labelling to get time-series data of the flows. This would allow us to calculate the strength of the response to ATP change in Acetyl-CoA flow towards FA biosynthesis and would allow us to tune the k parameter we used in our extended model for the transition from Acetyl-CoA to the `FAsyn` place. Having experimental data across different conditions, like we had for example for our basic model, could provide another useful dimension to our models. In the Results section we calculated the model parameters for different clusters of lipid product profiles. By observing changes in the parameters across conditions we could pinpoint areas or structures that show behavioural changes in the model. This could lead to mechanistic understanding of changes that lead to different conditions and disorders. This would be far more attractive with the time-series data and the control mechanisms that appear to be the reasons behind several metabolic disorders. Even with our high level models we could for example calculate the strength response parameter k across conditions.

4.3.2 Language methods

As we have mentioned in previous sections Petri Nets may not be suited to larger systems. Even for smaller system they lack the expressiveness to capture different parts of models at different levels of abstraction. **(author?)** [l Sobocinski and Stephens] propose Petri Nets with Boundaries that can decompose Petri Nets and compositional operators to define the interconnections between different parts. This approach has a graphical notation where each part is enclosed in box representing its boundaries. Each part is itself a Petri Net consisting of an internal structure of places and transitions as we have seen before. Every box has output ports that connect it to the outside world. These ports can be connected with places inside the box via normal transitions (Figure 4.1). The main operator of the algebra is the synchronisation between these parts for communication. The researchers have also provided a nice programming language borrowing elements from functional programming to use along with the algebra for Petri Nets with boundaries that I think it is important in order for these formalisms to be adopted by practitioners. Now this approach is quite

attractive because it allows for the decomposition of net models in an intuitive way and then communication between the components through the synchronisation operator. In a way it borrows some of the advantages of process-algebra models while keeping the vivid graphical notation of Petri Nets. For our purposes of biological modelling that would mean the different interacting parts could different pathways or different compartments. If for example we are only interested in one pathway we could define that part in detail while define only the interfaces interacting with the outside world for other pathways. This is what we have done informally in our extended model but this approach makes it formal. This approach however has as its main motivation behind the decomposition the minimisation of state-space for model checking. It will need some further work for it to be executable and useful for biological modelling. I am just presenting it here because I think it is a step in the right direction.

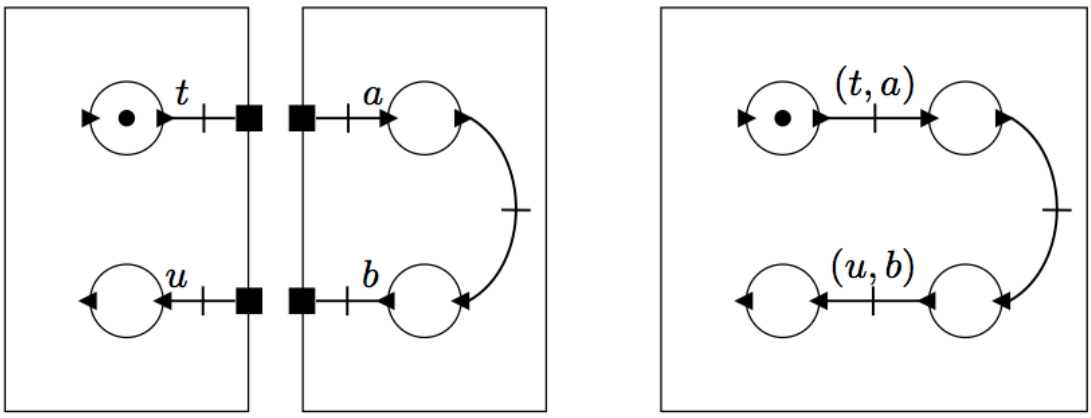


Fig. 4.1 Petri Nets with boundaries. On the left two composing parts of the net. Each has its own internal net structure and transitions. Some of the places can communicate via transitions withe the outports of that part(squares at the edge of the box). Communication between the two parts happens through synchronisation of the outports of the two(picture on the right).

This is part of a more general attempt to capture hierarchy in these minimilastic languages for distributed computing. In Petri Nets with boundaries we can talk at two levels, the local level of the transitions and state changes inside a box and then the communication through synchronisation between boxes. Another attempt that falls in the same category is bigraphs that also has a graphical notation and the ability to nest structures [16]. Degasperi and Calder [8] proposed a process algebra with hooks that attempts to provide modelling capabilities for multi-level biological systems and communication operators between the

levels. I think that these are steps in the right direction because we are interested in qualities of biological systems that naturally fall on different levels. We should be able to express the higher level qualities in terms of the lower level qualities. That way we could possible have multi-level stratified design of biological systems akin to computer programs organisation and abstraction [1].

Chapter 5

Conclusions

In this work, we have used the Petri Net and stochastic pi-calculus formal languages to capture a an abstract stochastic reaction-centric view of Fatty Acid biosynthesis and elongation and their control.

In the first part we presented a full model of the elongation process and motivated through our modelling goals the reduction to a more simple version of the model that we build and worked on for the rest of the study. This simpler model was elegantly described as a series of Bernoulli trials that guide the C2 concatenation steps making up the process. We then presented two methods to infer the parameters of the simplified model, first using local likelihood functions for each of the binomials representing the decisions and then using a global likelihood function for the entire process. The validity of our model to capture the defining iterative elongation procedure was verified with a real experimental dataset. The methods for parameter estimation were used successfully to infer the parameters of the model. We have also shown how we can observe parameter changes over different groups in the data, something that could be useful for identification of the mechanisms underlying conditions.

In the second part we extended the model to include one of the control mechanisms for the flow of the Acetyl-CoA precursor into the FA biosynthesis-elongation process. The model was again a high level one with entire pathways being represented by single transitions and only the metabolites at the interfaces between pathways being explicitly represented. The main regulatory mechanism was encoded with a transition from ATP to the transition taking Acetyl-CoA into FA biosynthesis. We then gave a possible parameterisation of the rate functions of the transitions out of Acetyl-CoA that it is able to capture the regulatory element of interest.

5.1 Future work

We provided some directions for future directions in both the description methodology and the models themselves. As for the next priority

References

[1] Abelson, H. and Sussman, G. J. (1987). Lisp: A language for stratified design.

[2] Baldan, P., Cocco, N., Marin, A., and Simeoni, M. (2010). Petri nets for modelling metabolic pathways: a survey. *Natural Computing*, 9(4):955–989.

[3] Berry, G. and Boudol, G. (1989). The chemical abstract machine. In *Proceedings of the 17th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 81–94. ACM.

[4] Boden, G. and Shulman, G. (2002). Free fatty acids in obesity and type 2 diabetes: defining their role in the development of insulin resistance and β-cell dysfunction. *European journal of clinical investigation*, 32(s3):14–23.

[5] Cardelli, L. (2005). Brane calculi. In *Computational methods in systems biology*, pages 257–278. Springer.

[6] Ciocchetta, F. and Hillston, J. (2009). Bio-pepa: A framework for the modelling and analysis of biological systems. *Theoretical Computer Science*, 410(33):3065–3084.

[7] Danos, V. and Laneve, C. (2004). Formal molecular biology. *Theoretical Computer Science*, 325(1):69–110.

[8] Degasperi, A. and Calder, M. (2013). A process algebra framework for multi-scale modelling of biological systems. *Theoretical Computer Science*, 488:15–45.

[9] Fisher, J. and Henzinger, T. A. (2007). Executable cell biology. *Nature biotechnology*, 25(11):1239–1249.

[10] Fontana, W. and Buss, L. W. (1996). *The barrier of objects: From dynamical systems to bounded organizations*. Citeseer.

[11] Gilbert, D., Heiner, M., and Lehrack, S. (2007). A unifying framework for modelling and analysing biochemical pathways using petri nets. In *Computational Methods in Systems Biology*, pages 200–216. Springer.

[12] Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The journal of physical chemistry*, 81(25):2340–2361.

[13] Heiner, M., Herajy, M., Liu, F., Rohr, C., and Schwarick, M. (2012). Snoopy–a unifying petri net tool. In *Application and Theory of Petri Nets*, pages 398–407. Springer.

[14] Ideker, T., Galitski, T., and Hood, L. (2001). A new approach to decoding life: systems biology. *Annual review of genomics and human genetics*, 2(1):343–372.

[15] Iverson, K. E. (2007). Notation as a tool of thought. *ACM SIGAPL APL Quote Quad*, 35(1-2):2–31.

[16] Jensen, O. H. and Milner, R. (2004). Bigraphs and mobile processes (revised). *Technical report, University of Cambridge Computer Laboratory*.

[17] Kitano, H. (2002). Systems biology: a brief overview. *Science*, 295(5560):1662–1664.

[l Sobocinski and Stephens] l Sobocinski, P. and Stephens, O. A programming language for spatial distribution of net systems.

[19] Lafont, Y. (1989). Interaction nets. In *Proceedings of the 17th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 95–108. ACM.

[20] Milner, R. (1980). A calculus of communicating systems.

[21] Milner, R., Parrow, J., and Walker, D. (1992). A calculus of mobile processes, ii. *Information and computation*, 100(1):41–77.

[22] Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580.

[23] Nielsen, J. (2009). Systems biology of lipid metabolism: from yeast to human. *FEBS letters*, 583(24):3905–3913.

[24] Olivier, B. G., Rohwer, J. M., and Hofmeyr, J.-H. S. (2005). Modelling cellular systems with pysces. *Bioinformatics*, 21(4):560–561.

[25] Orth, J. D., Thiele, I., and Palsson, B. Ø. (2010). What is flux balance analysis? *Nature biotechnology*, 28(3):245–248.

[26] Phillips, A. and Cardelli, L. (2007). Efficient, correct simulation of biological processes in the stochastic pi-calculus. *Computational Methods in Systems Biology*, 4695:184–199.

[27] Phillips, A., Cardelli, L., and Castagna, G. (2006). A graphical representation for biological processes in the stochastic pi-calculus. *Transactions in Computational Systems Biology*, 4230:123–152.

[28] Priami, C., Regev, A., Shapiro, E., and Silverman, W. (2001). Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Information processing letters*, 80(1):25–31.

[29] Salway, J. G. (2013). *Metabolism at a Glance*. John Wiley & Sons.

[30] Schneider, H.-C. and Klabunde, T. (2013). Understanding drugs and diseases by systems biology? *Bioorganic & medicinal chemistry letters*, 23(5):1168–1176.

Appendix A

Code

The code written for this project is available at a Github repository here. Following is a brief explanation of what is contained in the files in the Github repository.

Filename	Description
`fa.py`	Contains code for loading a model, executing it, and writing results to file (see Figure 3.9)
`fa.pyx`	Same as `fa.py` but in Cython for improved performance
`ml_est.R`	Contains code for the two methods for parameter inference for the basic model as described in Section 3.1.2. Also includes loading, filtering, and parameter calculation for the experimental dataset as presented in Section 3.1.3