CellBranch CoSMoS model : increments 1, 2 and 3

Richard B. Greaves[1], Sabine Dietmann[2], Austin Smith[2],

Susan Stepney[1], and Julianne D. Halley[1]

[1]York Centre for Complex Systems Analysis, University of York, UK
[2]Wellcome Trust-Medical Research Council Cambridge Stem Cell Institute,
University of Cambridge, UK

February 2016

University of York Technical Report YCS-2016-501

Contents

List of Figures

Preface

The principal barrier to gaining understanding of embryonic stem (ES) cell regulatory networks is their complexity. Reductionist approaches overlook much of the complexity inherent in these networks and treat the ES cell regulatory system as more or less equivalent to the sum of its component parts, studying them in relative isolation. However, as we learn more about regulatory components it becomes increasingly difficult to integrate complex layers of knowledge and to develop more refined understanding. We seek better control of the complexity inherent in non-equilibrium ES cell regulatory networks undergoing lineage specification by developing computer simulations of self-organisation using the CoSMoS approach.

Simulation, together with the hypothesis that lineage computation occurs at the edge of chaos, should allow us to investigate the driving of gradual accumulation of network complexity 'from the bottom up'. Here, we present the first steps in this design process: use of the CoSMoS approach to develop a highly abstracted model and simulation of regulatory network activity driven by just pluripotent transcription factors (TFs), at genome-wide scales.

First, we investigate three TFs in isolation: Oct4, Nanog and Sox2, central elements of the core pluripotent network of mouse embryonic stem cells. Second, we allow these TFs to couple, so that activity of one TF can ignite activity in another. Third, we add a further TF, cMyc, to the mix, and examine its effects. This model provides a suitable basis for future modelling of more complex TF interactions. Finally we give information on how to access and use our simulation code.

Acknowledgments

This work was performed as part of the CellBranch project, funded by the UK's Biotechnology and Biological Sciences Research Council (BBSRC), project reference BB/L018705/1.

Chapter 1

Background

1.1 Introduction

Mathematical or computational frameworks and tools are indispensable in the study of cell regulatory networks [Bornholdt, 2005; Zandstra and Clarke, 2014] because functions, traits and pathologies are rarely caused by single genes [Hartwell et al., 1999; Weatherall, 2001; Bornholdt, 2005]. The principal challenge that prevents comprehensive understanding (and simulation) of regulatory networks is their complexity [Mesarovic et al., 2004]. In the era of systems biology, the icon for molecular biology is the 'hairball' graph, which illustrates how everything seems to interact with almost everything else [Ferrell, 2009; Lander, 2010]. High-throughput technologies generate such large volumes of data that there is concern about how to grasp the big picture [Bray, 2003; Howe et al., 2008; Driscoll, 2009] and most data sets are not being used to their full potential.

Here we present three increments of a novel computational framework to interrogate the complexity of stem cell regulatory networks, developed as partof the CellBranch project. We employ a previously-described theoretical framework based on the notion that the backbone of stem cell fate computation is provided by the critical-like self-organisation of transcription factor (TF) regulatory networks [Halley and Winkler, 2008; Halley et al., 2009, 2012].

We apply the CoSMoS (Complex Systems Modelling and Simulaiton) framework [Andrews et al., 2010; Stepney and Andrews, 2015a; Stepney et al., 2016], which is specifically designed to capture the emergent properties of complex systems, and to guide the engineering of trustworthy computer simulations: those that are scientifically valid, useful and credible to third parties.

The structure of this report follows the CoSMoS patterns as defined in the CoSMoS approach, outlined in §1.2. Each part documents a model and simulator increment, in terms of the three phases of discovery, development, and exploration. We conclude with some reflections on the process and discussion of further work in chapter 11. Information about obtaining and running the simulation code is given in the appendix.

1.2 The CoSMoS approach

The CoSMoS approach [Andrews et al., 2010; Stepney and Andrews, 2015a; Stepney et al., 2016] supports the construction and exploration of computer simulations for the purposes of scientific research. CoSMoS describes a series of models and other components that need to be specified, designed, and implemented in order to build and use a fit-for-purpose simulator. The approach is guided by considering the simulator to be a form of *scientific instrument* [Andrews et al., 2012] that needs to be carefully designed, built, calibrated and used in a manner appropriate to specific research questions.

The CoSMoS approach is encapsulated as a *pattern language* [Alexander et al., 1977]. The CoSMoS patterns provide guidance on what to do at the various stages of a CoSMoS simulation project [Stepney, 2012; Stepney et al., 2016]. In this report we structure the increments explicitly in terms of these patterns.

To guide the reader through the pattern structure, we reproduce in boxed text a brief overview of the pattern: the pattern name and *intent*, a short phrase describing what should be done; and, where applicable, any *components* (including sub-patterns) that can be used to decompose the intent. We use section subheadings to capture the specific pattern names (named with initial capitals, such as *Research Context*) and other components (named in lower case, such as *success criteria*) and their position in the overall pattern structure.

Each increment starts at the top level of a simulation project, which is formed of three phases: Discovery, Development, and Exploration.

Part I

Increment 1 : single TF

Chapter 2

CoSMoS Simulation Project: increment 1

The models and results in this part of the report document the first increment of the CellBranch project using the CoSMoS design cycle. Here, we design and calibrate simulations of single TFs in isolation. This single TF version of the full model is not biologically realistic; its purpose is to serve as a building block of complexity to serve as the basis of following increments.

This part is a minimally updated version of [Greaves et al., 2015], changed to correct small errors, to bring in line with the latest version of CoSMoS patterns, and to interface with the other parts of this report.

> CoSMoS pattern: *CoSMoS Simulation Project:* Develop a basic fit-for-purpose simulation of the complex scientific domain of interest.
>
> The components of a *CoSMoS Simulation Project* are:
>
> - carry out a *Discovery Phase*
> - carry out a *Development Phase*
> - carry out an *Explorations Phase*

Chapter 3

Discovery phase 1

CoSMoS pattern: *Discovery:* Decide what scientific instrument to build. Establish the scientific basis of the project: identify the domain of interest, model the domain, and shed light on scientific questions.

The components of the *Discovery* phase are:

- identify the *Research Context*
- define the *Domain*
- do *Domain Modelling*
- [Argue Appropriate Instrument Designed (omitted)]

3.1 Discovery > Research Context

CoSMoS pattern: *Research Context:* Identify the overall scientific context and scope of the simulation-based research being conducted.

The components needed to identify the *Research Context* are:

- provide a thumbnail *overview* of the research context
- document the *research goals* and project scope
- agree the *Simulation Purpose*, including criticality and impact
- identify the *Team* members, including the Domain Scientist, the Domain Modeller and the Simulation Engineer, their roles, and experience
- document *Assumptions* relevant to the research context
- note the available *resources*, timescales, and other constraints
- determine *success criteria*
- decide whether to proceed, or walk away

3.1.1 Discovery > Research Context > overview

The context of this research is the investigation of a conceptual approach: self-organisation at the edge of chaos. We have argued that if the activity of single transcription factors can be described as critical-like branching processes, their interplay should define a critical-like genome-wide interference pattern that captures in some way the nature of the entire pluripotency transcription factor regulatory network [Halley et al., 2012].

Here we build a simulation based on the representation of TFs as *branching processes*. The mathematical concept of a branching process (BP) is as follows. Consider a population of individuals. At time t each individual i produces a next generation of m_i offspring individuals, with the value of m_i drawn from some probability distribution. Let the average number of offspring produced be μ. If $\mu > 1$, then the process is supercritical and the number of individuals grows without bound. If $\mu = 1$ then the system is critical and can either give rise to more individuals in the next step or lead to dissipation of the process. If $\mu < 1$ then the process goes to extinction.

Our model of TF BPs builds on this idea, and also allows the TFs to *interact* in such a way as to cause the regulatory network to self-organise at the edge of chaos. We capture the activity of single TFs as BPs in order to predict the interplay of multiple TFs and the emergent nature of the entire TF regulatory network, hypothesised to operate in a critical-like state [Halley et al., 2012].

For a TF to be stably expressed, its BP must be supercritical [Halley et al., 2012]. Therefore, by modelling the activity of TFs known to be expressed in mouse embryonic stem cells, we link the perturbation of a TF's cistrome (portion of the genome in which the TF displays some activity) with a dynamic and distributed description of TF activity. This is a prerequisite to being able to simulate the entire TF regulatory network of an ES cell, as argued in [Halley et al., 2012]. The TFs called Oct4, Sox2 and Nanog are central elements of the core pluripotent network of mouse embryonic stem cells. In this first increment, we develop our simulation for these three TFs in isolation, and so characterise how their associated TF BPs propagate in the absence of interference or communication.

Our incremental approach to the development of the full simulation commences with the simplest possible system: the operation of one transcription factor at genome-wide scales. We later add layers of further complexity, testing and calibrating as we go.

A model of a single pluripotent TF in isolation is far from complete and is not biologically realistic. It is only when multiple communicating TF BPs are simulated in parallel that we can expect to generate the interference patterns predicted to underpin circuitry self-organisation. As greater numbers of pluripotency TFs are included in the model, we anticipate that our simulations will become increasingly biologically realistic. In future increments we will augment the complexity of the computational model in a stepwise manner, adding detail and refining assumptions as we progress, and increasingly be able to provide insights not accessible by other means.

3.1.2 Discovery > Research Context > research goals

The overall research goals of this work are:

1. to create a simulation of Branching Process Theory (BPT) as applied to embryonic stem cell differentiation
2. to use this simulation to validate the application of BPT in this context
3. to make the simulation available for more general use

In this part, we document the first increment, of a single TF branching process.

3.1.3 Discovery > Research Context > Simulation Purpose

CoSMoS pattern: *Simulation Purpose:* Agree the purpose for which the simulation is being built and used, within the *Research Context.*

The components of the *Simulation Purpose* are:

- define the role of the simulation
- determine the criticality of the simulation results

Simulation role

The role of the simulation is exploratory: to provide evidence of the usefulness of BPT as a model of decision making in stem cell differentiation. The simulation will be used to investigate which values of the average branching ratio are required to set up a sustainable TF branching process.

Simulation criticality

The simulation work is being used to explore the suitability of a particular xonceptual modelling approach, BPT, in the domain. The simulation results are not safety, security, or financially critical: they will not be used directly in the development of any products.

3.1.4 Discovery > Research Context > team

The three main CoSMoS roles are fulfilled by the team members in the following way:

- *Domain Scientist*: Halley, an expert on BPT as applied to stem cell differentiation, backed up by a domain expert in ES cell biology (Smith), and a data collection expert (Dietmann)
- *Domain Modeller*: Greaves, with CoSMoS domain modelling experience, backed up by a further CoSMoS modelling expert (Stepney)
- *Simulation Engineer*: Greaves, with agent based simulation engineering experience

3.1.5 Discovery > Research Context > Assumptions

CoSMoS pattern: *Document Assumptions:* Ensure assumptions are explicit and justified, and their consequences are understood.

The components of *Document Assumptions* are:

- identify that an assumption has been made, and record it
- for each assumption, determine its nature and criticality
- for each assumption, document the reason it has been made
- for each reason, document its justification, or flag it as “unjustified” or “unjustifiable”
- for each assumption, document its connotations and consequences
- for each critical assumption, determine the connotations for the scope and fitness-for-purpose of the simulation
- for each critical assumption, achieve consensus on the appropriateness

of the assumption, and reflect this in fitness for purpose arguments
- revisit the simulation scope in light of the assumption, as appropriate

A.1 Cistrome data can be provided by processed ChIP-Seq data

reason It is the data we have

justification This is one standard use for ChIP-Seq data

consequence ChIP-Seq data is variable across measurements, so we will need to check the robustness of our results to this variation

A.2 It is sufficient to consider only the key pluripotency transcription factors: Nanog, Oct4, Sox2

reason As a first step in providing insight, we consider the three TFs widely acknowledged to be central components of the core pluripotent network

justification See for example [Boyer et al., 2005]

consequence We will not be able to determine the effect of further TFs. However, it should be straightforward to incorporate further TF data into the multi-cistrome model.

A.3 We can use mouse data as a suitable proxy for data from human ES cells

reason Suitable mouse data is more readily available; mouse ES cells have an unambiguous ‘ground state’; so mouse data is a good basis for evaluating the TF BP model

justification Although effective manipulation of human ES cells is a long term goal, here we are only assessing the TF BP model

consequence We cannot extrapolate results to the human system

3.1.6 Discovery > Research Context > resources, timescales, other constraints

The project has a one year duration. The Domain Scientist is employed full time, and Simulation Engineer part time.

The work has access to a local computer cluster, for running simulations and gathering performance metrics.

The team members are split between York (Halley, Greaves, Stepney) and Cambridge (Smith, Dietmann).

3.1.7 Discovery > Research Context > success criteria

1. a single-cistrome simulator that exhibits the expected behaviours, and can be used as the basis for multi-cistrome simulator development
2. a single-cistrome simulator that can justify the use of the TF BP model to analyse stem cell fates

3.2 Discovery > Domain

CoSMoS pattern: *Domain:* Identify the subject of simulation: the real-world biological system, and the relevant information known about it.

The components are:

- draw an explanatory *Cartoon*
- provide an *overview* description of the domain
- provide a *Glossary* of relevant domain-specific terminology
- Document *Assumptions* relevant to the domain
- define the *scope and boundary* of the domain – what is inside and what is outside
- identify relevant *sources*: people, literature, data, models, etc

3.2.1 Discovery > Domain > Cartoon

CoSMoS pattern: *Cartoon:* Sketch an informal overview picture of the *Domain.*

Figure 3.1 is a cartoon of the regulatory process. A single gene regulation and its expression is conceptually relatively straightforward; the complex interplay of multiple interacting regulatory processes is not.

3.2.2 Discovery > Domain > overview: embryonic stem (ES) cell biology

Modern, high-throughput laboratory techniques routinely provide large-scale datasets including complete genome sequences, dynamic measurements of gene expression, extensive lists of regulatory proteins and RNAs, and *in vivo* occupancy of DNA by TFs, cofactors and nucleosomes [Ay and Arnosti, 2011]. Such datasets facilitate the investigation of ES cell regulatory networks. To create a complete multi-layered model of a stem cell network one should exploit these big data to bridge gaps between the phenotypic behaviour of whole cells and key regulatory molecules [Xu et al., 2010].

We need to capture the results of multiple high-throughput experiments within a logical and transparent conceptual and computational framework in order to facilitate the interrogation of multiple layers of complex regulatory information. Our initial model is based on the complete genome sequence of mouse embryonic stem cells and on ChIP-Seq data that capture the density of TF binding sites throughout the genome. TFs operate in parallel, influencing each other; according to our hypothesis, they produce genome-wide interference patterns that capture in some way the predicted nature of the entire pluripotent circuitry.

Embryonic stem (ES) cells have the potential to produce all of the different cell types within the body, but this behaviour cannot yet be efficiently exploited *in vitro*. We have considerable knowledge of the component parts of the regulation of ES cells maintained under precise external conditions [Martello and Smith, 2014], but during normal development many different types of regulatory factors interact, enabling cells to respond flexibly to changing environments.

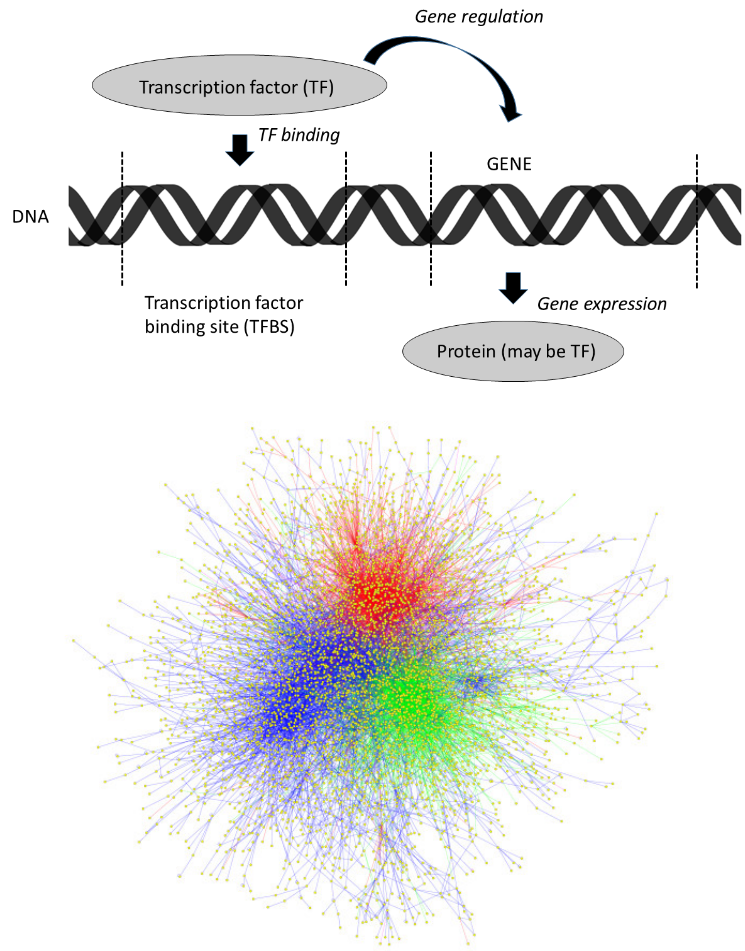

Figure 3.1: Domain > Cartoon: (top) The regulatory process: a TF protein binds to DNA at the binding site, thereby regulating production of protein (which may be a TF) from the corresponding gene (gene expression). (bottom) Expressed proteins may include other TFs that can regulate expression of other genes: a 'hairball graph' of the human proteome and its binding interactions [Ferrell, 2009, fig.1]

The regulatory network of single ES cells is therefore some function of both cell intrinsic and cell extrinsic variables.

Here we assume that pluripotency is a state of individual ES cells. ES cells exit pluripotency via a transient 'primed' state that facilitates cell fate computation [Nichols and Smith, 2009]. Our knowledge of this exit process and the transient primed state is incomplete, partly because it is difficult to obtain data from transient cell states [Teles et al., 2013]. The process of pluripotency exit itself is intrinsically disorganised and/or chaotic in order for it to integrate intrinsic and extrinsic information and compute cell fate. According to our conceptual framework, regulatory circuitries compute cell fate trajectories via 'critical-like dynamics' at the edge of chaos [Halley et al., 2012].

Nanog, Oct4 and Sox2 form part of the core pluripotency circuitry of ES cells [Boyer et al., 2005]. Oct4 in particular seems central to understanding pluripotency. Oct4 expression level is closely regulated, with deviations either above or below a certain expression range resulting in differentiation [Niwa et al., 2000]. It has been suggested that protein complexes, in which Oct4 is involved, help to establish a dynamic competition between individual elements, serving to buffer the differentiation-promoting activity of Oct4 [Muñoz Descalzo et al., 2013].

Fluctuations are inevitable in any system that has many degrees of freedom. At static equilibrium, such fluctuations ultimately disappear but under non-equilibrium conditions, fluctuations are often great enough to drive reorganisation toward new dynamic states [Nicolis and Prigogine, 1977; Chaisson, 2004]. If continual driving is experienced, complex spatiotemporal patterning usually results and systems are said to have 'self-organised' [Nicolis and Prigogine, 1977; Gollub and Langer, 1999; Ball, 2001].

In biology, the growth and development of organisms occurs far from equilibrium. The stem cell regulatory networks that facilitate these processes are replete with positive and negative feedback loops and nonlinear interactions. When faced with overwhelming complexity, the natural tendency of humans is to either reduce, simplify or ignore it. Reductionist thinking makes systems (a) easier to think about, (b) easier to consider manipulating, and (c) easier to predict, provided non-equilibrium driving is minimal.

Over the last few decades, there has been increasing awareness of the limitations of the reductionist approach [Crutchfield et al., 1986; Farmer and Packard, 1986; Bak and Paczuski, 1993; Parisi, 1993; Kauffman, 1995] and it has become clear that some laws of nature cannot be deduced by resolving more detail [Vicsek, 2002]. This so called 'new era of physics' focuses on developing complex behaviour out of simplicity, instead of the traditional reductionist approach that reduced complexity to its simplest possible form [Kadanoff, 1987; Anderson, 1991; Parisi, 1993]. Non-equilibrium driving can have profound consequences on system behaviour, a realisation that contrasts with our natural tendency to assume systems are near equilibrium or at least show some steady state behaviour. Equilibrium and reductionist thinking pervades most scientific disciplines [Bak and Paczuski, 1995; Ball, 1999, 2001; Ekeland, 2002], including molecular and stem cell biology.

The differentiation of pluripotent cells in the early embryo is a fascinating non-equilibrium process that results in the production of numerous specialised cell types. More than 600 different proteins have been implicated in exit from a naïve pluripotent state and control of early state transitions in the mouse [Kalkan and Smith, 2014]. As our focus shifts from individual components to complex communication networks, experimental studies have become more difficult. Not only do central features of complex networks, such as robustness, prevent straight forward analysis and interpretation of network behaviours, but many experiments cannot be performed because of ethical reasons surrounding the use of human embryos.

Computer simulation sidesteps the ethical, moral and political issues surrounding use of human embryos. It therefore represents an alternative route to gaining new insight in to this promising field of regenerative medicine. Our overarching aim is to gain sufficient understanding so that any cell type of therapeutic interest can be generated effectively at will.

3.2.3 Discovery > Domain > Glossary: terms and acronyms

> CoSMoS pattern: *Glossary:* Provide a common terminology across the simulation project.

The main biological terms used in the various models are:

binding site : section of DNA that binds a given TF and influences transcription of associated genes

branching process (BP) : the mathematical model underlying inspiration of the TF BP framework being investigated here

ChIP-Seq : a technique to identify the binding sites of transcription factors on DNA

cistrome : the portion of the genome associated with a specific TF; a pattern of genome-wide binding sites to which the TF displays some activity

pluripotent stem cell : a cell capable of generating all the cell types present in the adult body

segment : the genome data is segmented, into say 10k or 50k base-pair sequences, in order to apply the TFBP framework

transcription factor (TF) : a protein that binds to DNA to influence transcription of the associated gene

3.2.4 Discovery > Domain > Assumptions

See §3.1.5 for the *Assumptions* pattern requirements.

A.4 The genome can be modelled as a set of overlapping TF cistromes without needing epigenetic factors

reason We are looking only at TF segments, and the pluripotent state can be induced by TFs alone

justification See, for example, [Kim et al., 2008]

consequence Behaviours facilitated by other factors, such as epigenetics, will be unseen in the model

A.5 a TF binding site is either bound or unbound, there is no partial TF binding

reason not enough data to say otherwise

A.6 a segment can be either active or inactive; there are no differing amounts of activation

reason Simplification: the data does say whether a segment has one or more binding sites

justification This is the first increment; we may revisit the necessity and impact of this assumption in later increments

consequence We will not be able to separate out behaviours of groups of genes in a segment. In order to do so, we could use smaller segments. But segments cannot be made too small, else we would lose correlations between related TFs.

A.7 we can investigate cell decision making by modelling an individual cell, not a population

reason cells have internal decision making, although they can also be influenced by their environment

justification See, for example, [Loh et al., 2006]

consequence We will not be able to investigate population-level decision making

3.2.5 Discovery > Domain > scope

- single cell model
- single transcription factor model
- later increments will add more, coupled TFs, and more interacting cells

3.2.6 Discovery > Domain > sources

- Domain scientists
- Biological literature, as referenced in the various overviews
- Chip-seq data for various cistromes (source: Dietmann)

3.3 Discovery > Domain Modelling

CoSMoS pattern: *Domain Modelling:* Produce an explicit description of the relevant domain concepts.

The components of *Domain Modelling* are:

- *collaborate* with the identified Domain Scientist
- draw an explanatory *Cartoon*
- discuss and choose the *Modelling Approach* and level of abstraction
- build the *Domain Model* using the chosen modelling approach
- build the *Data Dictionary*
- build the *Domain Experiment Model*
- define the *Expected Behaviours*
- document *Assumptions* relevant to the domain model
- [Argue Domain Model Appropriate (omitted)]

3.3.1 Discovery > Domain Modelling > collaborate

The lead domain scientist (Halley) and the domain modellers (Greaves, Stepney) collaborated closely throughout the development of the domain model, translating and abstracting the conceptual TF BP model into a form suitable for simulation.

The domain scientists (Halley, Smith, Dietmann) collaborated on refining the research context.

The simulation engineer (Greaves) collaborated with the the data collection expert (Dietmann) on the form and content of the biological data provided.

3.3.2 Discovery > Domain Modelling > Cartoon

See §3.2.1 for the *Cartoon* pattern.

Due to the structure of our Domain Model description, the *Domain Modelling Cartoon* is presented in the section on the TF BP model (figure 3.4), and should be read in that context.

3.3.3 Discovery > Domain Modelling > Modelling Approach

CoSMoS pattern: *Modelling Approach:* Choose an appropriate modelling approach and notation.

A central part of this design process is to develop the simplest possible working model at each stage of the modelling process. This 'agile' approach ensures that simulation code is not unnecessarily complicated. It also helps to ensure that if a coding problem is found, it is simple matter to backtrack to the last working model.

The domain model is captured using UML, in anticipation of an agent-based, object-oriented design and implementation of the simulator.

3.3.4 Discovery > Domain Modelling > Domain Model

Our domain modelling gives rise to several models at different levels of abstraction: a specifically biological stem cell model of regulatory networks, a model simplifying detailed transcription regulatory networks using branching process theory, and a generic abstract model, which we refer to as the 'sparking posts' model.

Note that the sparking posts model could also be used as a domain model for other biological phenomena as captured by branching process theory, such as patterns of information flow in the human brain.

Regulatory network

We have mouse genome data including the suite of binding sites within it. For convenience and simplicity, we divide this sequence in to 50 kilobase (kb) segments, any of which may or may not contain binding sites for a particular TF of interest. If a 50kb segment contains a binding site for our transcription factor, X, then the segment is said to be part of the X cistrome.

Data about the locations of the transcription factor binding sites, in relation to the gene segments in the model, is provided experimentally by ChIP-Seq data. Figure 3.2 is a representation of ChIP-Seq data.

The regulatory network components can be captured in a model such as that shown in figure 3.3. However, we abstract away from many of these 'hairball' inducing details, and consider the system instead in terms of the TF BP model.

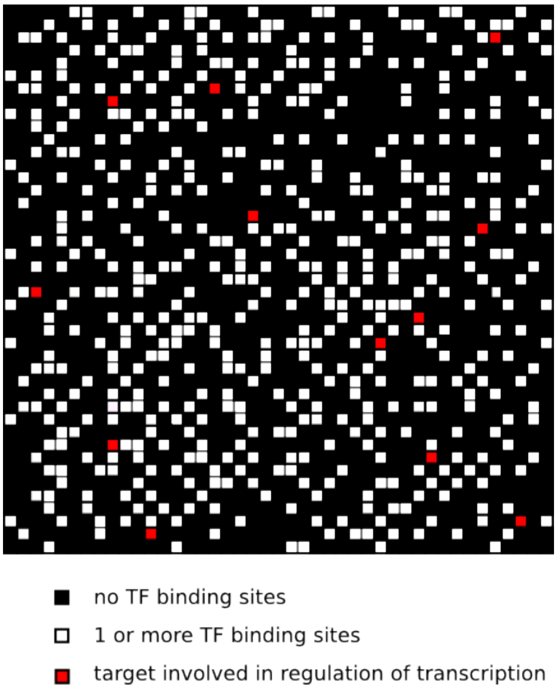

Figure 3.2: A representation of a set of ChIP-Seq data for a cistrome (part of the genome relevant to a specific TF). Each square represents a 50kb segment of DNA. A white square is a segment that contains at least one binding site site for a product that is not a TF. A red square is a segment that contains at least one binding site site for a product that is a TF. A black square is a segment that does not belong to this cistrome.

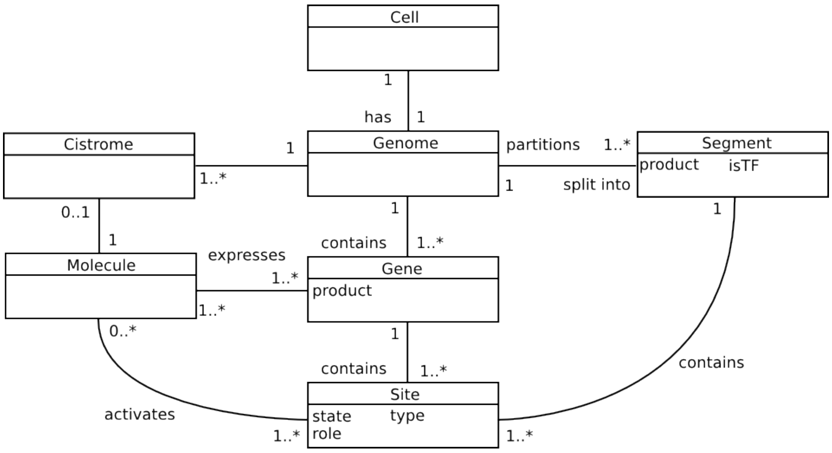

Figure 3.3: Stem cell pluripotency regulatory network model: class diagram. The stem cell has a genome comprised of genes, which can alternatively be described as a cistrome (or set of cistromes), each being comprised of segments of gene which may or may not contain transcription factor binding sites.

Transcription Factor Branching Process (TF BP) model

A common approach to understanding cell regulatory processes is the application of concepts, tools and techniques developed in mathematics, physics or computer science [MacArthur et al., 2008]. Network representations, for example, can accommodate multiple types of data within a single visual illustration that provides an overview of regulatory pathways and components [Gallagher and Appenzeller, 1999; MacArthur et al., 2008]. Empirically-derived interaction networks can be difficult to interpret, often appearing as a 'hairball' graph as regulatory mechanisms are increasingly dissected.

We use here a novel way to visualise and simulate genome-wide regulatory network interactions. Our coarse-grained approach does not require details of binding constants prerequisite for most ODE models of stem cell regulation. In many previous computational or mathematical models of stem cell regulatory networks, TFs are represented as single nodes with binary (*on/off*) behaviour. Here, we use a different approach that captures TF activity as a dissipative branching process that propagates within the bounds imposed by the TF's unique cistrome.

Unlike reductionist models that capture TF activity using single variables in an equation, in our model we explicitly represent a background delocalisation of TF activity throughout the genome. We can visualise the activity of each TF's BP as a kind of gateway through which regulatory information pertaining to the TF passes over time.

The TF BP model allows a decoupling between details of binding site constants and the emergent effect of TF activity throughout the genome. Instead of struggling with countless (often unknown) binding constants, we consider the overall flow of regulatory information at genome-wide scales. It is thus more suitable for attempts to discover how the ES cell regulatory network behaves as a whole during computation of lineage choice. Through this more coarse-grained methodology, we hope to discover complex interactions that can easily be overlooked by studies that focus on only a handful of key regulatory components at a time.

The potential binding of a TF to target regions throughout the genome is determined by ChIP-Sequencing. The data set or 'footprint' for a given TF comprises a unique pattern of TF-DNA interactions that is somewhat dependent upon the precise methods used to infer interactions. The precise footprint for a specific TF may vary between different experimental datasets. Such 'fuzziness', rather than being a nuisance, is intrinsic to the TF BP model.

If we understand the activity of any given TF as a branching process of regulatory information propagating through time, it makes sense for there to be some correlation between observed TF expression and the saturation of target sites influenced by TF activity. The significance of this point should become clearer in later increments, when we simulate multiple cistrome data sets. In this increment, we focus on simulating a single TF's BP to introduce the groundwork for our approach.

Figure 3.4 presents a Cartoon of the TF BP model. Each square in the figure corresponds to a 50kb segment of the mouse genome. Black squares represent segments that contain no binding sites for the TF of interest, while red and white squares represent segments with at least one binding site for the TF of

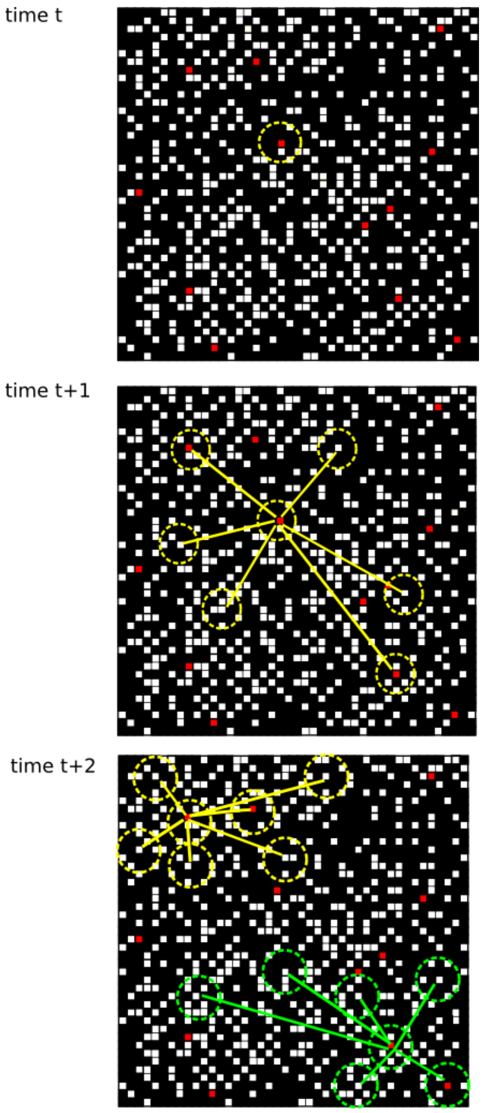

Figure 3.4: Domain Modelling > Cartoon: A branching process representation of the overall flow of regulatory information, which serves as the basis of our simulation. At t, assume the circled red segment is active. At time $t+1$ this activates m further randomly chosen segments (arrows), and itself deactivates. At time $t+2$, all of these newly active segments that are themselves red each activate a further m randomly chosen segments, and deactivate.

interest. The difference between a red and white segment lies in their products. A red segment has products that include TFs, whereas none of the products of a white segment is a TF. Henceforth, when we refer to a ‘red segment’ we mean a gene segment that can bind TF and thus become stimulated into transcribing further TFs.

We capture the countless (ill-defined or unknown) cascades of gene activation via TF production and feedback as a branching process in which TFs produce other TFs while also regulating the remainder of the genome. There are potentially three qualitatively different types of behaviour for any $\mathrm{TF_X}$ branching process. Firstly, the cistrome X is saturated and the $\mathrm{TF_X}$ gene is continually and stably expressed. Alternatively, there is the opposite type of emergent behaviour, with $\mathrm{TF_X}$ expression occurring at a very low noisy level that is not sustainable unless $\mathrm{TF_X}$ is supported by continual activation of the $\mathrm{TF_X}$ gene via some external signal. Finally there is a dynamic intermediate between these extremes where a branching process only just percolates through the $\mathrm{TF_X}$ cistrome. In all cases, the targets of $\mathrm{TF_X}$ are divided in to two types: (1) dissipative targets that do not propagate information back in to the $\mathrm{TF_X}$ cistrome, and (2) amplifying targets that are either TFs themselves and capable of propagating information or code for signalling molecules that are involved in signal transduction.

We define an average branching ratio, called m, for our gene regulation branching process. That is to say that once transcribed, a gene (or gene segment in our case) will produce m product molecules (in this single cistrome model these will all be the TF that binds to binding sites within the cistrome of interest). If the active site is associated with TF products then new TFs are produced and these can bind to other TF binding sites in the system. In this way, up to m segments are activated in the next time step of the algorithm. In the time step after this each of the active segments can go on to activate m further segments and so on as illustrated in Figure 3.4.

This TF BP model is built on the classical BP theory outlined in section 'Domain > overview', and is adapted in the following ways:

- m is related to the BP branching factor μ, but is not the same, because here the m 'offspring' include both white and red segments, yet only red segments go on to produce further 'offspring'.
- In the supercritical case, the number of offspring cannot increase without bound, but only up to the number of relevant segments in the cistrome.
- The individuals are segments, and do not 'die' at the end of a generation; rather they can be reused (reselected) in subsequent generations.

Domain Model: Sparking Posts

In order to model a branching process, we produce our domain model in terms of a metaphor. To capture the nature of critical-like self-organisation hypothesised to underpin lineage computation, we have reduced the system to a 'sparking posts model'. This computational model is used to define the backbone of critical-like self-organisation upon which other layers of complexity are elaborated.

The TF BP representation of our system is modelled as a 'sparking posts' representation of the cistrome in which each segment is modelled as a metal 'post' which emits 'sparks' once it has been activated by an incoming spark emitted by another post in the previous timestep. The sparks represent the TF products of the genes contained within a given segment and are therefore the principal mode of communication between cistromes, the genome being effectively the sum of all cistromes in the system.

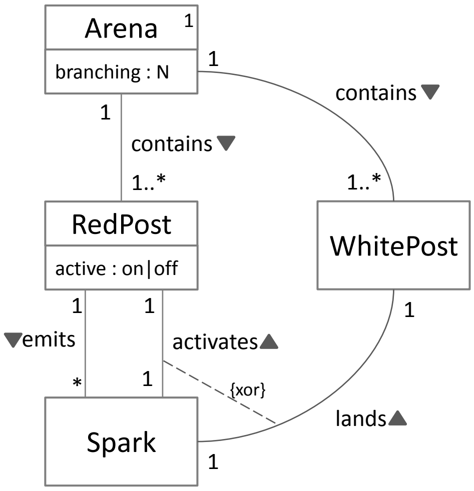

Figure 3.5: Sparking posts model for a single arena: class diagram. There is one arena, which has a branching factor. The arena contains multiple red posts, which can be on or off, and multiple white posts. A red post can emit several sparks; each spark is emitted by a particular post. A particular spark either activates a red post or lands on white post, but not both.

So the Domain Model is as follows.

An *arena* contains metal *posts*, some *red*, some *white*. The arena is an abstraction of a particular cistrome; the posts are abstractions of the segments containing binding sites (red and white squares in figure 3.2); red posts are abstractions of segments that express TFs (red squares in figure 3.2).

Posts may be *active* (on) or *inactive* (off). In a timestep, an active red post emits m *sparks*. A post being active is an abstraction of a gene in a segment being active; a red post sparking is an abstraction of an active gene expressing a TF.

Posts become inactive after they have sparked. A spark lands on a random post in the arena (that is, the model is aspatial), and activates it.

Continued propagation of sparks relies on the activation of sufficient red posts at each timestep.

Figures 3.5 and 3.6 capture aspects of this Domain Model.

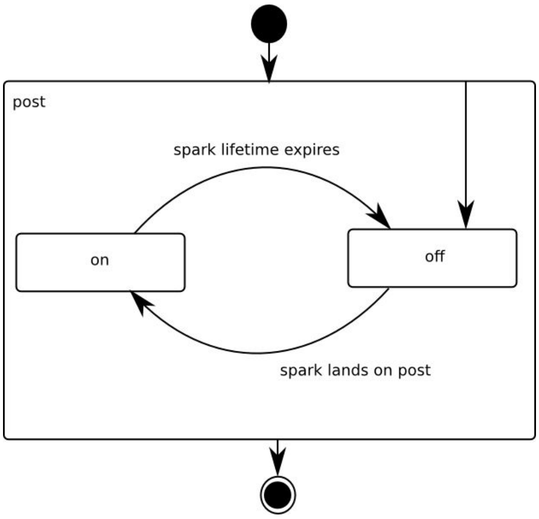

Figure 3.6: RedPost state diagram. RedPosts are initially inactive (off); become active (on) if a spark lands; then become inactive in the next timestep.

p	total number of posts in the arena
r	number of red posts
m	sparks emitted per active red post
s_0	number of red posts active initially
t	timestep
s_t	number of red posts active at timestep t

Figure 3.7: Data Dictionary: (top) parameters, constant during a simulation run; (bottom) variables, changing during a simulation run

	Nanog	Sox2	Oct4
p	4310	3330	2540
r	631	542	466
r/p	0.146	0.163	0.183
$p/r = m_c$	6.8	6.1	5.5

Figure 3.8: Data Dictionary: parameter values for p (number of posts, or segments in the cistrome); r (the number of red posts, or red segments in the cistrome); derived value r/p, the proportion of posts that are red; derived value m_c, the critical branching factor in the infinite arena limit.

3.3.5 Discovery > Domain Modelling > Data Dictionary

CoSMoS pattern: *Data Dictionary:* Define the modelling data used to build the simulation, and the experimental data that is produced by domain experiments and the corresponding simulation experiments.

The sparking post model's parameters and variables are shown in Figure 3.7. Figure 3.8 shows the values of some of these parameters for the cistromes of interest here.

3.3.6 Discovery > Domain Modelling > Domain Experiment Model

The Domain Model is sufficiently abstracted from the Domain that the Simulation Experiments in this increment do not mirror any Domain Experiments. Hence it is unnecessary to build a Domain Experiment Model in this increment.

3.3.7 Discovery > Domain Modelling > Expected Behaviours

CoSMoS pattern: *Expected Behaviours:* Describe the expected emergent behaviours of the underlying system.

The 'sparking posts' domain model forms the basis for subsequent simulation development.

We can form a much simpler version of the model, which will help to understand the effect of noise. Since there are a finite number of posts, stochastic fluctuations will occur, and sparks might occasionally miss many or all of the red posts. Here we instead assume that posts are always hit the average number of times. This is the case when $p \rightarrow \infty$ whilst keeping r/p constant. We are interested in the proportion of red posts active in the 'steady state', in limit of large time.

At time t there are s_t red posts active. Each of these active post emits m sparks, so a total of $s_t \times m$ sparks are emitted. Let each of these sparks be absorbed by a separate post, of which a fraction r/p are red. So at the next timestep, there are $s_{t+1} = s_t mr/p$ red posts active.

The number of active red posts reduces with time if $m < p/r$, and so the arena is extinguished, with $s_\infty = 0$.

The number of active red posts steadily grows with time if $p/r < m$, until there are more sparks emitted than there are posts in total (moving outside our assumption of each spark being absorbed by a separate post), and so the arena saturates with $s_\infty = r$.

The critical value, m_c, where this change of behaviour happens is $m_c = p/r$. Values for m_c for the TFs of interest are shown in figure 3.8.

Hence the expected behaviour of the single cistrome simulation is to quench for low values of m, saturate for high values of m, and have a tipping point around m_c.

3.3.8 Discovery > Domain Modelling > Assumptions

See §3.1.5 for the *Assumptions* pattern requirements.

First, we have some assumptions related to the TF BP model, which we note as they have an impact on the sparking posts model.

A.8 the product of a TF producing segment is the TF whose cistrome we are modelling

reason An assumption underlying use of the TF BP model

justification The TF may not be directly produced; there may be a cascade of production, but the TF BP model collapses this cascade. We are investigating this model.

consequence This is an abstraction from the biology, made to allow us to model the highly complex processes. If it works, this abstraction could also provide an approach to include other features such as epigenetics and mRNAs in a tractable model.

A.9 the identity of the TFs produced during transcription is irrelevant in the single cistrome model

reason An assumption underlying use of the TF BP model

justification The TF BP model assumes that the relevant scale of computation is the cistrome level, abstracted from specific details of the individual TFs

Assumptions directly related to the sparking posts model are:

A.10 a spark from a post can hit any post with equal probability: there is no notion of a ‘distance’ between posts

reason an aspatial model

justification the TF BP model collapses a potential cascade of TFs into a single ‘proxy’ TF. This cascade would lose any spatial dependence in the DNA.

A.11 a post cannot be hit by more than one spark per timestep: there is no notion of different ‘capacity’ posts

reason follows from assumption A.6

Chapter 4

Development phase 1

CoSMoS pattern: *Development:* Build the scientific instrument: produce a simulation platform to perform repeated simulation, based on the output of the *Discovery* phase.

The components of the development phase are:

- *revisit* the Research Context
- do *Platform Modelling*
- develop a *Simulation Platform*
- [Argue Instrument Built Appropriately (omitted)]

4.1 Development > revisit

The research context is unchanged in the light of *Discovery* phase activities. The TF concepts need to be reinterpreted in terms of the sparking posts model.

4.2 Development > Platform Modelling

CoSMoS pattern: *Platform Modelling:* From the *Domain Model*, develop a platform model suitable to form the requirements specification for the *Simulation Platform.*

The relevant components of *Platform modelling* are:

- choose a *Modelling Approach* for the platform modelling
- develop the *Platform Model* from the Domain Model
- develop the *Simulation Experiment Model* from the Domain Experiment Model
- document *Assumptions* relevant to the platform model

4.2.1 Development > Platform Modelling > Modelling Approach

We use the same approach as for domain model, assisting seamless development.

4.2.2 Development > Platform Modelling > Platform Model

The emergent tipping point behaviour is not part of the platform model. The rest of the ‘sparking posts’ model carries over from the domain model unchanged.

4.2.3 Development > Platform Modelling > Simulation Experiment Model

In this case we do not have a relevant Domain Experiment Model to use as the basis for design. The kinds of Simulation Experiments we will do will require the following input/output and instrumentation:

- derive parameters p, r from ChIP-Seq data
- input and set parameters p, r, m, s_0
- run the simulation for T timesteps
- output s_T, the number of active posts at time T
- perform multiple runs with different random seeds

4.2.4 Development > Platform Modelling > Assumptions

A.12 the sparks emited by an active post last for one simulation time step

reason simplicity

justification first increment

consequence half lives and decay rates are not modelled; they may be added in later increments

4.3 Development > Simulation Platform

CoSMoS pattern: *Simulation Platform:* Develop the executable simulation platform that can be used to run the *Simulation Experiment.*

The relevant components of developing the simulation platform are:

- choose an *Implementation Approach*
- implement Platform Model (details omitted here)
- implement Simulation Experiment Model (details omitted here)
- perform calibration (details omitted here)
- document *Assumptions* relevant to the simulation platform

4.3.1 Development > Simulation Platform > Implementation Approach

The simulation is implemented as an object-oriented Java application using the MASON simulation environment to handle such things as time-stepping the simulation and on screen graphics (when running in graphical mode).

4.3.2 Development > Simulation Platform > Assumptions

There are no further relevant assumptions made in the simulation platform development.

Chapter 5

Exploration phase 1

CoSMoS pattern: *Exploration:* Use the simulation platform resulting from *Development* to explore the scientific questions established during *Discovery*.

The components are:

- *revisit* the Research Context
- perform *Results Modelling*
- perform a *Simulation Experiment*
- [Argue Instrument Used Appropriately (omitted)]

5.1 Exploration > revisit

The research context is unchanged in the light of *Discovery* and *Development* phase activities.

5.2 Exploration > Results Modelling

CoSMoS pattern: *Results Modelling:* Develop a results model suitable for interpreting simulation experiment data in Domain Model terms.

The relevant components of results modelling are:

- build a *Visualisation Model*
- build a *Results Model*
- [Argue Results Model Appropriate and Consistent (omitted)]

5.2.1 Exploration > Results Modelling > Visualisation Model

CoSMoS pattern: *Visualisation Model:* Visualise the simulation experiment results of the *Data Dictionary* in a manner relevant to the users.

- The visualisation mimics the cistrome data in figure 3.2
- The output data is presented as plots of activation at $T = 1000$ against m

5.2.2 Exploration > Results Modelling > Results Model

The results model is the number of active posts (cistrome activity) at time $T = 1000$.

5.3 Exploration > Simulation Experiment

CoSMoS pattern: *Simulation Experiment:* Use the simulation as a scientific instrument to explore the behaviour of the system.

The relevant components of a simulation experiment are:

- design the experiment
- perform the experiment
- analyse the results

5.3.1 Exploration > Simulation Experiment > design

The parameters p (number of posts) and r (number of red posts) are effectively fixed for any given set of experimentally derived cistrome data (figure 3.8). We can also generate synthetic data to create systems with a range of p and r values to explore general behaviours.

We identify 4 experiments to perform on the single-arena simulation:

Experiment E.0: Effect of m. For each cistrome, create an arena with the relevant p and r values, and $s_0 = r$. Explore the effect of m by locating those values of m for which the system remains fully saturated: all red posts are active at all time steps. Compare this with the expected m_c value (figure 3.8) for a noiseless system.

E.0	Nanog	Sox2	Oct4
p	4310	3330	2540
r	631	542	466
m	0–50	0–50	0–50
s_0	r	r	r

Experiment E.1: Effect of s_0, sensitivity to initial conditions. Repeat E.0 with a smaller value of s_0.

E.1	Nanog	Sox2	Oct4
p	4310	3330	2540
r	631	542	466
m	0–50	0–50	0–50
s_0	$r/2$	$r/2$	$r/2$

Experiment E.2: Effect of r. Create an arena with the Nanog p value, and a range of r values. At each value of r, determine the values of m for which the system remains saturated throughout the simulation.

E.2	Nanog
p	4310
r	200, 400, 600, 800
m	0–50
s_0	r

Experiment E.3: Effect of noise. Keeping the value of p/r fixed at the Nanog value of 4310/631, investigate the effect of reducing p. This gives some insight into whether we can use smaller arenas in experiments to improve simulation performance, without affecting the results.

E.3	Nanog	Nanog
p	2000	1000
r	293	146
m	0–50	0–50
s_0	r	r

5.3.2 Exploration > Simulation Experiment > perform

Number of simulation runs

We are not performing any statistical analyses at this stage of the project, merely inspecting behaviour. However, the simulation is essentially stochastic, and when we do come to perform statistics, we will need to choose the number of runs based on the significance, power, and effect size of interest. For consistency, we make that choice now, and use the relevant number of runs.

We require a statistical significance of 99% (a 1% false positive rate), a statistical power of 99% (a 1% false negative rate), and a 'medium' effect size (Cohen's $d = 0.5$, the ability to distinguish a difference in means of 0.5 of a standard deviation). Calculating the required sample size for these experimental parameters[1] gives 192.

We round this up, and take the number of runs to be $N = 200$.

Protocol

One simulation run comprises the p and r values of a particular arena (chosen for example to match Nanog, Sox2, Oct4 data), an m value (0–50), and a starting activity s_0 , ad detailed in the experiment design tables.

For each simulation run, we record the proportion of active red posts at the final timestep, $T = 1000$.

For each parameter set (p, r, m, s_0), we run the simulation $N = 200$ times.

[1] using, for example, the calculator at http://powerandsamplesize.com/Calculators/Compare-2-Means/2-Sample-Equality

5.3.3 Exploration > Simulation Experiment > analyse results

Experiments E.0 and E.1

See figure 5.1 for the results of the simulation runs.

The observed values of m where the system 'switches on', and can maintain saturation, are close to the calculated m_c values, see figure 5.2. However, m has to be somewhat higher than this to saturate the finite-sized arena.

Starting with only half the posts active makes little difference to the results.

Experiment E.2

For experiment E.2, we take $p = 4310$ (as in Nanog), vary r, and examine how the value of m_c changes. We use $s_0 = r$ throughout.

See figures 5.3–5.4 for the results of the simulation runs.

Recall that the theoretical tipping point value is $m_c = p/r$. So as r increases, m_c should decrease. This is observed (figure 5.4).

The smaller the value of r, the noisier the behaviour (visible as more extended are the boxplots in figures 5.3). This demonstrates how stochastic effects are more prominent when there are fewer red posts available.

Experiment E.3

For experiment E.3, we take $p/r = 4310/631$ (as in Nanog), and vary p keeping p/r constant (mimicking a different sized arena but with the same density of red posts). We use $s_0 = r$ throughout.

See figure 5.5 for the results of the simulation runs; compare with figure §5.1(top) for the 'full' arena.

The systems tip at the same point, but the behaviour gets noisier as p (and hence r) decreases, and stochastic effects become more pronounced.

Summary

Having exercised the single cistrome simulation, increment 1, we are satisfied that the simulation returns results qualitatively in line with what we expect: simulations run with a higher branching parameter, m, exhibit more sustained activity in that cistrome, and the proportion of the cistrome that contains TF binding site also affects simulation behaviour. Thus we have been able to show that Nanog, Oct4 and Sox2 cistrome branching processes all behave differently, and that each has its own value of m, m_c, at which the simulation runs started to show sustained activity. This observed m_c is marginally higher than the theoretical value, due to noise and finite size effects.

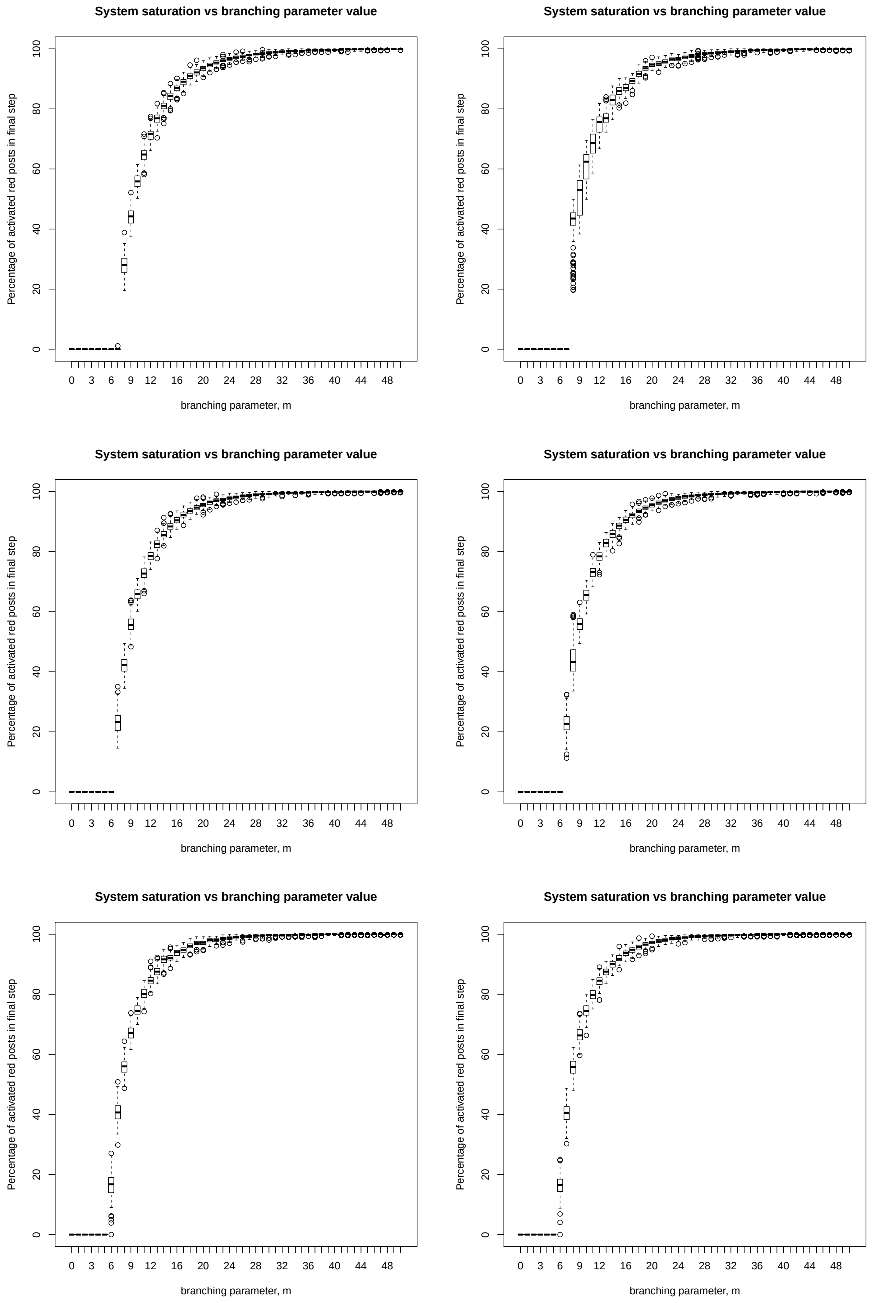

Figure 5.1: p and r corresponding to (top row) Nanog data, calculated $m_c = 6.8$; (middle row) Sox2 data, calculated $m_c = 6.1$; (bottom row) Oct4 data, calculated $m_c = 5.5$. (left) E.0: $s_0 = r$; (right) E.1: $s_0 = r/2$.

	Nanog	Sox2	Oct4
$p/r = m_c$ calc	6.8	6.1	5.5
m_c observed	8	7	6

Figure 5.2: Theoretical value m_c, the critical branching factor in the infinite arena limit, compared to the observed value in the finite-sized arena simulation.

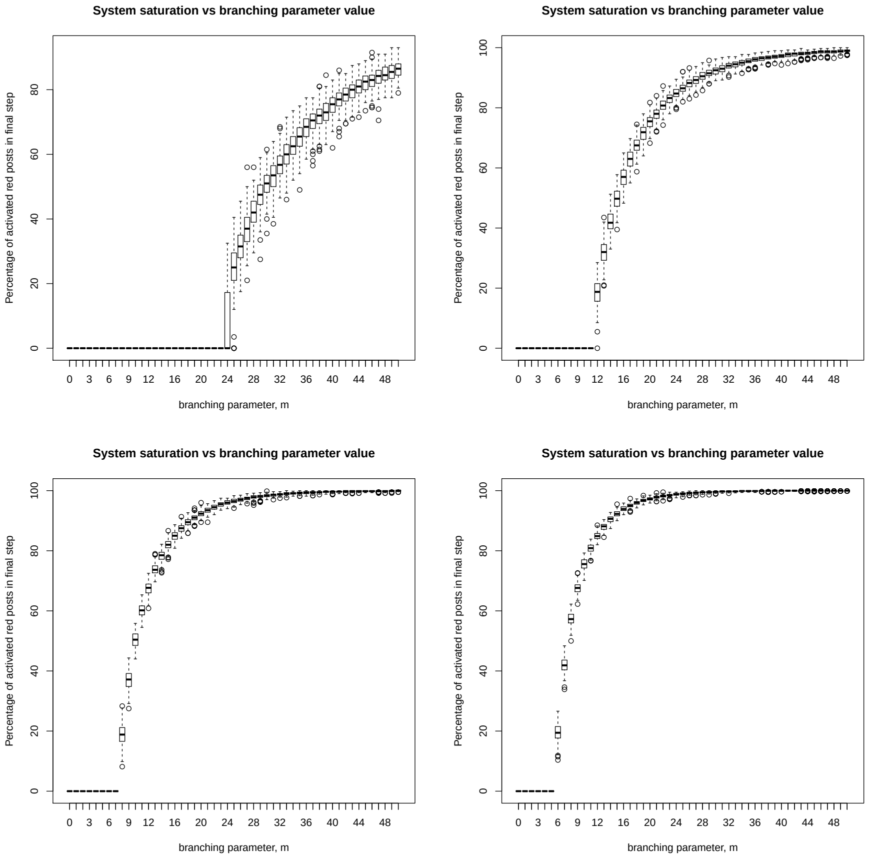

Figure 5.3: E.2: varying r; here $p = 4310$ (Nanog). (top left) $r = 200$; (top right) $r = 400$; (bottom left) $r = 600$; (bottom right) $r = 800$

r	m obs	m_c calc
200	23–24	21.6
400	11–12	10.8
600	7–8	7.2
800	5–6	5.4

Figure 5.4: E.2: observed value of m at tipping point for different r (with $p = 4310$, Nanog), versus calculated value m_c

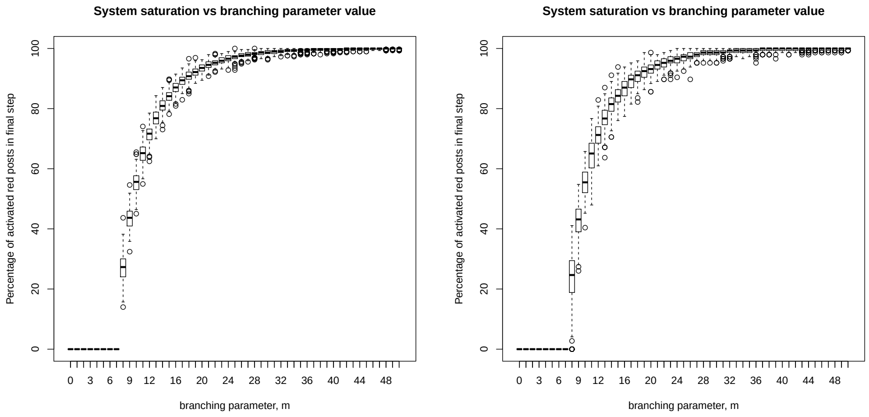

Figure 5.5: E.3: varying p with constant p/r: (left) $p = 2000, r = 293$; (right) $p = 1000, r = 146$

Part II

Increment 2 : multiple TFs

Chapter 6

CoSMoS Simulation Project: increment 2

The models and results in this part document the second increment of the CellBranch project using the CoSMoS approach. Here, we increment the model and simulation with multiple interacting TF branching processes, and perform new simulation experiments.

CoSMoS pattern: *CoSMoS Simulation Project:* Develop a basic fit-for-purpose simulation of the complex scientific domain of interest.

The components of a *CoSMoS Simulation Project* are:

- carry out a *Discovery Phase*
- carry out a *Development Phase*
- carry out an *Explorations Phase*

Chapter 7

Discovery phase 2

CoSMoS pattern: *Discovery:* Decide what scientific instrument to build. Establish the scientific basis of the project: identify the domain of interest, model the domain, and shed light on scientific questions.

The components of the *Discovery* phase are:

- identify the *Research Context*
- define the *Domain*
- do *Domain Modelling*
- [Argue Appropriate Instrument Designed (omitted)]

7.1 Discovery > Research Context

CoSMoS pattern: *Research Context:* Identify the overall scientific context and scope of the simulation-based research being conducted.

The components needed to identify the *Research Context* are:

- provide a thumbnail *overview* of the research context
- document the *research goals* and project scope
- agree the *Simulation Purpose*, including criticality and impact
- identify the *Team* members, including the Domain Scientist, the Domain Modeller and the Simulation Engineer, their roles, and experience
- document *Assumptions* relevant to the research context
- note the available *resources*, timescales, and other constraints
- determine *success criteria*
- decide whether to proceed, or walk away

7.1.1 Discovery > Research Context > overview

The overall research context remains much as it was in increment 1: the investigation of the conceptual branching process approach. We have argued that if the activity of single transcription factors can be described as critical-like branching processes, their interplay should define a critical-like genome-wide interference pattern that captures in some way the nature of the entire pluripotency transcription factor regulatory network [Halley et al., 2012].

We now develop the models and simulation resulting from the first increment, and increase their biological relevance by permitting the system to consist of two or more interacting transcription factor branching processes (TF BPs) , thus permitting us to gain understanding of the behaviour of constructively interfering branching processes. We defer inclusion of destructive interference between TF BPs until a later increment, to permit us to more fully understand this simpler representation of the system prior to the addition of another layer of complexity.

This second increment of model development will permit us to characterise the behaviour of the central elements of the core pluripotent network of mouse embryonic stem cells, that is, to characterise the associated TF BPs and how they propagate in the presence of cross-cistrome communication.

Our incremental approach to the development of the full simulation continues with the simplest possible augmentation of the system: the co-operation of two or more transcription factors at genome-wide scales.

This model of multiple interacting pluripotent TF BPs is still far from complete, and not biologically realistic. It is only when multiple TF BPs are simulated in parallel, generating branching process interference patterns via both constructive and *destructive* interference, that we can expect to generate the interference patterns predicted to underpin circuitry self-organisation. This increment allows the simulation of multiple cistromes interacting constructively. As greater numbers of pluripotency TFs are included in the model, we expect that our simulations will become increasingly biologically realistic.

7.1.2 Discovery > Research Context > research goals

Our research goals remain essentially unchanged from those stated in increment 1: to create a simulation of Branching Process Theory (BPT) applied to embryonic stem cells and to use this simulation to validate use of BPT in this context. Here we report on the second increment of the TF BP simulation, a multi-TF branching process.

7.1.3 Discovery > Research Context > Simulation Purpose

CoSMoS pattern: *Simulation Purpose:* Agree the purpose for which the simulation is being built and used, within the *Research Context.*

The components of the *Simulation Purpose* are:

- define the role of the simulation
- determine the criticality of the simulation results

Simulation role

The role of the simulation remains exploratory: to provide evidence of the usefulness of BPT as a model of decision making in stem cell differentiation. The simulation will be used to investigate which values of the average branching ratio are required to set up a sustainable TF branching process and how different TF BPs interact with each other.

Simulation criticality

The simulation results are not safety, security, or financially critical.

7.1.4 Discovery > Research Context > team

The main CoSMoS roles as defined in increment 1 remain unchanged throughout this second increment.

7.1.5 Discovery > Research Context > Assumptions

CoSMoS pattern: *Document Assumptions:* Ensure assumptions are explicit and justified, and their consequences are understood.

The components of *Document Assumptions* are:

- identify that an assumption has been made, and record it
- for each assumption, determine its nature and criticality
- for each assumption, document the reason it has been made
- for each reason, document its justification, or flag it as “unjustified” or “unjustifiable”
- for each assumption, document its connotations and consequences
- for each critical assumption, determine the connotations for the scope and fitness-for-purpose of the simulation
- for each critical assumption, achieve consensus on the appropriateness of the assumption, and reflect this in fitness for purpose arguments
- revisit the simulation scope in light of the assumption, as appropriate

The key assumptions made in the first increment of the simulation development remain relevant, as do their justifications and consequences. In addition:

A2.1 It is sufficient to consider only constructive interference between cistromes

reason As part of an incremental development of providing insight.

justification This is a sensible increment that will provide further insight.

consequence The branching processes can interfere both constructively and destructively [Halley et al., 2012], so the results of this increment will lack full biological relevance.

7.1.6 Discovery > Research Context > resources, timescales, other constraints

The project has a one year duration; this increment comprises the final 4 months. The Domain Scientist is employed full time, and Simulation Engineer part time.

The work has access to a local computer cluster, the YARCC, for running simulations and gathering performance metrics.

The team members are split between York (Halley, Greaves, Stepney) and Cambridge (Smith, Dietmann).

7.1.7 Discovery > Research Context > success criteria

1. a multi-cistrome simulator development
2. a multi-cistrome simulator that can justify the use of the TF BP model to analyse stem cell fates

7.2 Discovery > Domain

CoSMoS pattern: *Domain:* Identify the subject of simulation: the real-world biological system, and the relevant information known about it.

The components are:

- draw an explanatory *Cartoon*
- provide an *overview* description of the domain
- provide a *Glossary* of relevant domain-specific terminology
- Document *Assumptions* relevant to the domain
- define the *scope and boundary* of the domain – what is inside and what is outside
- identify relevant *sources*: people, literature, data, models, etc

7.2.1 Discovery > Domain > Cartoon

Figure 3.1 is a cartoon of the regulatory process and is still a relevant description of the biological domain as we understand it.

7.2.2 Discovery > Domain > overview: embryonic stem (ES) cell biology

We present an overview of the relevant biology in increment 1. We continue to seek to exploit the big data available to understand the phenotypic behaviour of entire cells in terms of the behaviour of key regulatory molecules [Xu et al., 2010] via creation of a multi-layered model of a stem cell regulatory network.

7.2.3 Discovery > Domain > Glossary: terms and acronyms

CoSMoS pattern: *Glossary:* Provide a common terminology across the simulation project.

The main biological terms used in the various models in this increment are described in the Glossary provided in increment 1.

7.2.4 Discovery > Domain > Assumptions

See §7.1.5 for the *Assumptions* pattern requirements.

The assumptions presented in §3.2.4 remain valid for this increment.

7.2.5 Discovery > Domain > scope

- a single cell model
- multiple TFs interfering constructively
- later increments may add destructive interference
- later increments may add more biological detail and build towards a model of interacting cells

7.2.6 Discovery > Domain > sources

Our sources remain as described in increment 1:

- Domain scientists
- Biological literature, as referenced in the various overviews
- ChIP-Seq data for various cistromes (source: Dietmann)

7.3 Discovery > Domain Modelling

CoSMoS pattern: *Domain Modelling:* Produce an explicit description of the relevant domain concepts.

The components of *Domain Modelling* are:

- *collaborate* with the identified Domain Scientist
- draw an explanatory *Cartoon*
- discuss and choose the *Modelling Approach* and level of abstraction
- build the *Domain Model* using the chosen modelling approach
- build the *Data Dictionary*
- build the *Domain Experiment Model*
- define the *Expected Behaviours*
- document *Assumptions* relevant to the domain model
- [Argue Domain Model Appropriate (omitted)]

7.3.1 Discovery > Domain Modelling > collaborate

The project team continued to collaborate in the manner described in increment 1, in order to develop a robust and appropriate model of our enhanced system.

7.3.2 Discovery > Domain Modelling > Cartoon

See §3.2.1 for the *Cartoon* pattern.

Due to the structure of our Domain Model description, the *Domain Modelling Cartoon* (figure 3.4) should be read in the context of the TF BP model.

7.3.3 Discovery > Domain Modelling > Modelling Approach

CoSMoS pattern: *Modelling Approach:* Choose an appropriate modelling approach and notation.

We continue to capture the domain model using UML in anticipation of continuing our agent-based design and implementation of the simulator.

7.3.4 Discovery > Domain Modelling > Domain Model

As discussed in increment 1, our domain modelling gives rise to several models at different levels of abstraction: a specifically biological stem cell model of regulatory networks, a model simplifying detailed transcription regulatory networks using branching process theory, and a generic abstract model, which we refer to as the 'sparking posts' model. Here we describe that changes to the increment 1 model from allowing multiple constructively interfering cistromes.

Regulatory network

We have mouse genome data including the suite of binding sites within it. For convenience and simplicity, we divide this sequence in to 50 kilobase (kb) segments, any of which may or may not contain binding sites for a particular TF of interest. If a 50kb segment contains a binding site for our transcription factor, X, then the segment is said to be part of the X cistrome.

Data about the locations of the transcription factor binding sites, in relation to the gene segments in the model, is provided experimentally by ChIP-Seq data as in the work described in increment 1. Figure 3.2 is a representation of ChIP-Seq data.

Transcription Factor Branching Process model

We continue to use the Transcription Factor Branching Process model described in increment 1. This novel, coarse-grained approach does not require details of binding constants prerequisite for most ODE models of stem cell regulation. As in the original implementation of the simulation, the refined simulation will also explicitly represent a background delocalisation of TF activity throughout the genome.

Domain Model: Sparking Posts

In increment 1, in order to model a branching process, we produced our domain model in terms of a metaphor.

To capture the nature of critical-like self-organisation hypothesised to underpin lineage computation, we reduced the system to a 'sparking posts' model. This computational model was used to define the backbone of critical-like self-organisation upon which this and other layers of complexity are elaborated.

So, to re-iterate the Domain Model used as the basis for our simulation implementation: The TF BP representation of our system is modelled as a 'sparking posts' representation of the cistrome in which each segment is modelled as a metal 'post' which emits 'sparks' once it has been activated by an incoming spark emitted by another post in the previous timestep. The sparks represent the TF products of the genes contained within a given segment and are therefore the principal mode of communication between cistromes, the genome being effectively the sum of all cistromes in the system.

The Domain Model with multiple TFs is as follows.

An *arena* contains metal *posts*, some *red*, some *white*. There are several arenas; there are some red posts that appear in the same position in different arenas: these are called *shared* posts. An arena is an abstraction of one particular cistrome; the posts are abstractions of the segments containing binding sites (red and white squares in figure 3.2); red posts are abstractions of segments that express TFs (red squares in figure 3.2); shared red posts are abstractions of the same segment expressing multiple TFs related to different cistromes.

Posts may be *active* (on) or *inactive* (off). In a timestep, each active red post emits m *sparks*. A post being active is an abstraction of a gene in a segment being active; a red post sparking is an abstraction of an active gene expressing a TF.

Posts become inactive after they have sparked.

The emitted sparks lands on random posts in the arena (that is, the model is aspatial). If a spark lands on an unshared red post, it activates it. If a spark lands on a shared red post, then the spark is transferred to any inactive corresponding post in another arena; if all the shared posts in other arenas are active, the spark activates the post in the original arena. That is, the spark is transferred to another arena where possible. No post can accept more than one spark, whether it shares it or not.

Continued propagation of sparks in an arena relies on the activation of sufficient red posts at each timestep. Sharing sparks between arenas allows an arena to become or stay active even if it does not produce enough sparks itself.

Figure 7.1 illustrates spark sharing; figure 7.2 shows the updated class diagram (compare figure 3.5).

7.3.5 Discovery > Domain Modelling > Data Dictionary

> CoSMoS pattern: *Data Dictionary:* Define the modelling data used to build the simulation, and the experimental data that is produced by domain experiments and the corresponding simulation experiments.

The sparking post model's parameters and variables are as in increment 1, see figure 3.7. Figure 3.8 shows the values of some of these parameters for the cistromes of interest here. Figure 7.3 shows the posts shared between the arenas.

7.3.6 Discovery > Domain Modelling > Domain Experiment Model

This is not needed in increment 2.

7.3.7 Discovery > Domain Modelling > Expected Behaviours

> CoSMoS pattern: *Expected Behaviours:* Describe the expected emergent behaviours of the underlying system.

With multiple interconnected arenas, we expect sparking behaviour in any arena to be affected by the behaviour in other arenas with shared posts.

We could at this point develop a coupled ODE model, in terms of the number of shared posts, giving an indication of the behaviour as cistrome size tends to infinity. However, we omit that development at this time.

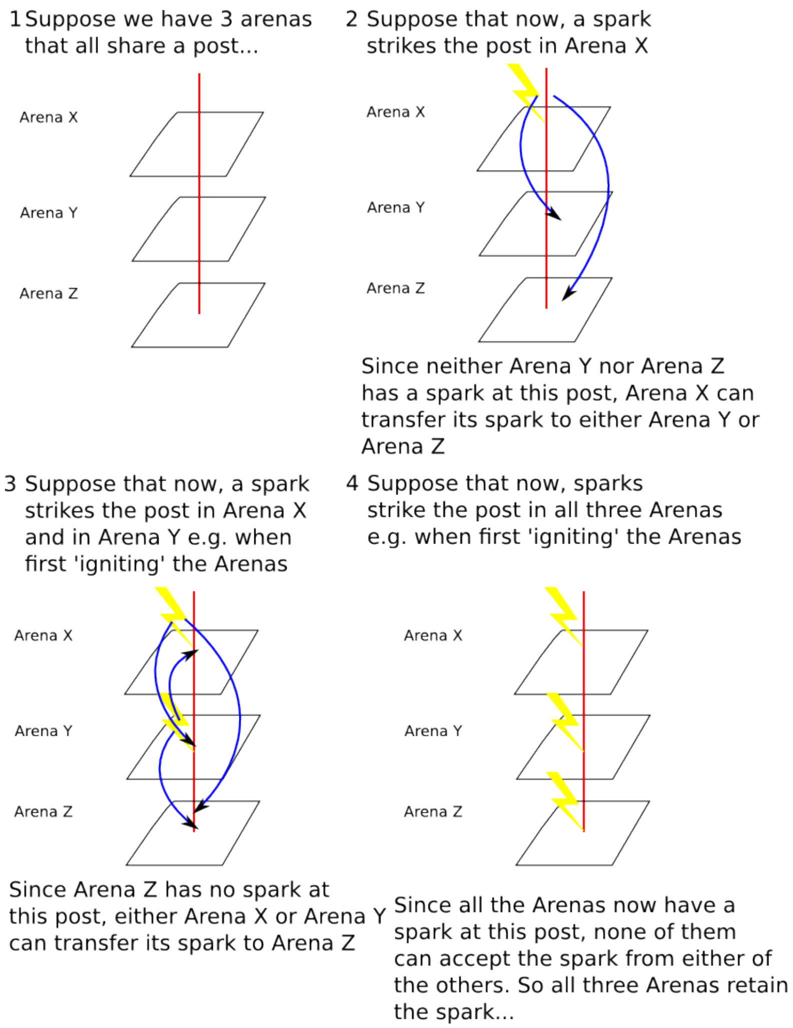
1 Suppose we have 3 arenas that all share a post...
Arena X
Arena Y
Arena Z
2 Suppose that now, a spark strikes the post in Arena X
Arena X
Arena Y
Arena Z
Since neither Arena Y nor Arena Z has a spark at this post, Arena X can transfer its spark to either Arena Y or Arena Z
3 Suppose that now, a spark strikes the post in Arena X and in Arena Y e.g. when first 'igniting' the Arenas
Arena X
Arena Y
Arena Z
Since Arena Z has no spark at this post, either Arena X or Arena Y can transfer its spark to Arena Z
4 Suppose that now, sparks strike the post in all three Arenas e.g. when first 'igniting' the Arenas
Arena X
Arena Y
Arena Z
Since all the Arenas now have a spark at this post, none of them can accept the spark from either of the others. So all three Arenas retain the spark...

Figure 7.1: Domain Model: illustration of communication between arenas. (1) a shared red post: an abstraction of a TF binding site common to two or more cistromes; (2) a shared post is struck by one spark, in arena X; it will activate the post in arena Y or Z; (3) a shared post is struck by two sparks, in arenas X and Y; they will activate the post in arena Z (by transferring the spark from X or Y) and in arena X or Y (the other spark cannot be transferred); (4) a shared post is struck by a spark in each arena, and the post is activated in each arena.

7.3.8 Discovery > Domain Modelling > Assumptions

See §7.1.5 for the *Assumptions* pattern requirements.

The assumptions in increment 1 are still relevant to the increment 2 model. We add a further assumption to allow for communication between cistromes in the model:

A2.2 in any given timestep, a post in a cistrome can gain at most one spark from being hit and from sharing sparks from shared posts.

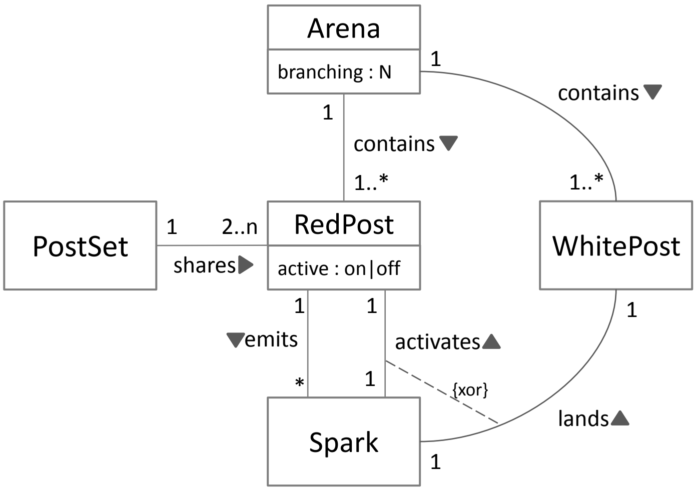

Figure 7.2: Sparking posts model components for multiple arenas with shared posts: class diagram. Each arena has a branching factor. Each arena contains multiple red posts, which can be on or off, and multiple white posts. A red post can emit several sparks; each spark is emitted by a particular post. A particular spark either activates a red post (figure 7.1 illustrates which arena this red post is in) or lands on white post, but not both. A shared set of posts comprises two to n red posts, where n is the number of arenas (every red post in a shared set is in a distinct arena; not shown in the diagram); a shared post is in a particular shared set.

arena	shared red posts
Nanog–Oct4	194
Nanog–Sox2	287
Oct4–Sox2	237

Figure 7.3: Data Dictionary: The number of red posts shared by the arenas

reason follows from assumption A.6 (‘no differing amounts of activation’) and assumption A.11 (‘a post cannot be hit by more than one spark per timestep’)

Chapter 8

Development phase 2

CoSMoS pattern: *Development:* Build the scientific instrument: produce a simulation platform to perform repeated simulation, based on the output of the *Discovery* phase.

The components of the development phase are:

- *revisit* the Research Context
- do *Platform Modelling*
- develop a *Simulation Platform*
- [Argue Instrument Built Appropriately (omitted)]

8.1 Development > revisit

The research context is unchanged in the light of *Discovery* phase activities.

8.2 Development > Platform Modelling

CoSMoS pattern: *Platform Modelling:* From the *Domain Model*, develop a platform model suitable to form the requirements specification for the *Simulation Platform.*

The relevant components of *Platform modelling* are:

- choose a *Modelling Approach* for the platform modelling
- develop the *Platform Model* from the Domain Model
- develop the *Simulation Experiment Model* from the Domain Experiment Model
- document *Assumptions* relevant to the platform model

8.2.1 Development > Platform Modelling > Modelling Approach

We use the same approach as for the domain modelling, assisting seamless development.

8.2.2 Development > Platform Modelling > Platform Model

The increment 2 platform model is developed from the increment 1 version by allowing multiple arenas, adding the concept of shared posts, and the spark sharing algorithm.

8.2.3 Development > Platform Modelling > Simulation Experiment Model

In this case we do not have a relevant Domain Experiment Model to use as the basis for design. The kinds of Simulation Experiments we will do will require the following input/output and instrumentation:

- derive parameters p, r, and shared r from ChIP-Seq data, for each arena
- input and set parameters p, r, m, s_0, and shared r for each arena
- run the simulation for T timesteps
- output s_t, the number of active posts at each timestep up to time T
- perform multiple runs with different random seeds

8.2.4 Development > Platform Modelling > Assumptions

We continue to assume that the sparks generated from an active post last for one simulation timestep as discussed in the Platform Model assumptions for increment 1.

8.3 Development > Simulation Platform

CoSMoS pattern: *Simulation Platform:* Develop the executable simulation platform that can be used to run the *Simulation Experiment.*

The relevant components of developing the simulation platform are:

- choose an Implementation Approach
- implement Platform Model (details omitted here)
- implement Simulation Experiment Model
- perform calibration (details omitted here)
- document *Assumptions* relevant to the simulation platform

8.3.1 Development > Simulation Platform > implementation approach

The modified simulation is again implemented as an object-oriented Java application using the MASON simulation environment to handle such things as time-stepping the simulation.

8.3.2 Development > Simulation Platform > Simulation Experiment Model

The behaviour of the increment 2 simulation experiments should reduce to that of increment 1 in the case of a single arena. As part of the testing process, we have re-run some of the increment 1 experiments, requiring the same results.

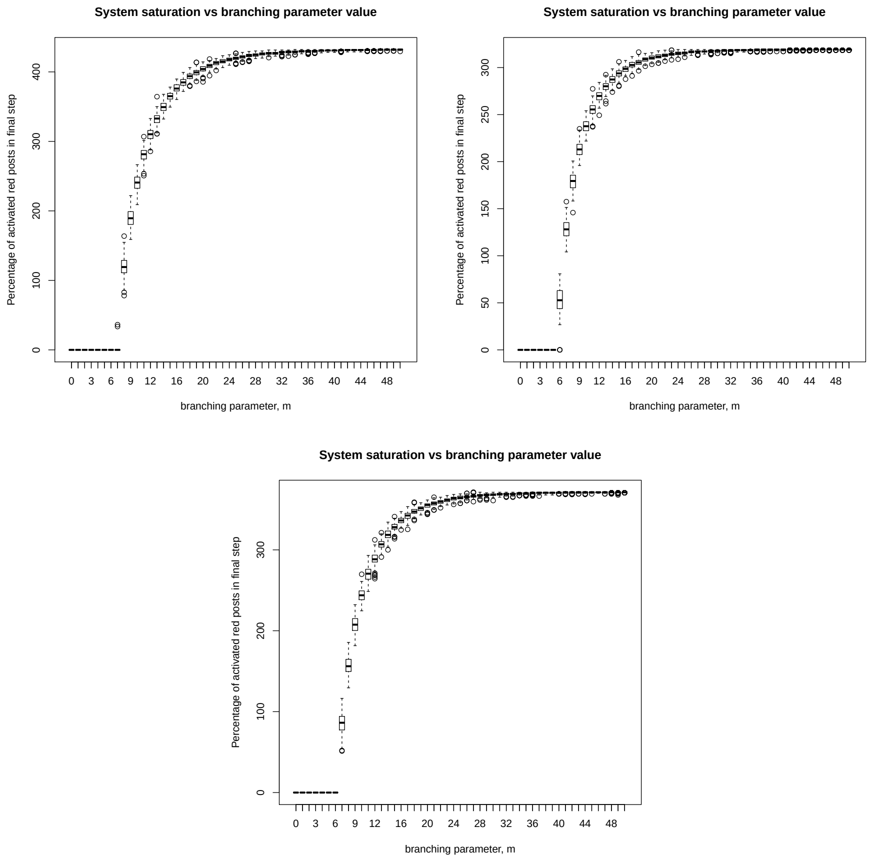

Figure 8.1: Replication of E.1. Determination of the critical value of the branching parameter, m, for arenas constructed from the cistromes for the three core pluripotency transcription factors. The upper left hand panel shows the result for the Nanog arena, the upper right for the Oct4 arena and the lower panel that for the Sox2 arena.

Testing experiment E.1

We repeat experiment E.1 from increment 1 with each of the three core pluripotency cistromes to verify that the modified simulation returns results consistent with those of the previous increment. As before, these simulations commence with $s_0 = 0.5r$ and m is varied between 0 and 50 in an attempt to locate the critical value of m, m_c, at which we first start to observe sustainable branching in the cistrome of interest.

The results obtained are shown in figure 8.1, and are unchanged from figure 5.5

Testing experiment E.3

We partially repeat experiment E.3 from increment 1 to show that the effects of altering the values of p and r in the input cistrome can be reproduced by

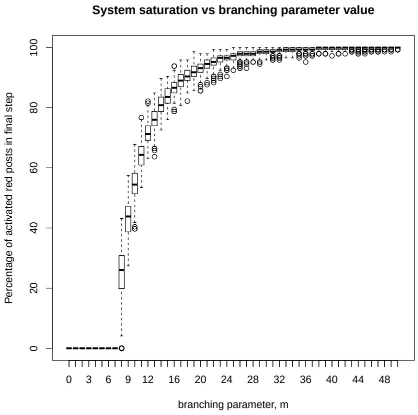

Figure 8.2: Replication of E.3. $p = 1000$ and $r = 146$. $s_0 = r$ and m is varied from 0 to 50.

increment 2. This experiment uses a synthetic, generated cistrome with $p = 1000$ (i.e. 1000 posts) and $r = 146$ (i.e. 146 red posts). The synthetic cistrome created is in effect a scaled down Nanog cistrome – for Nanog p=4310 and r=631.

The experiment starts with $s_0 = r$, that is, with all red posts active. The results are shown in figure 8.2, and are unchanged from figure 5.5 (right hand panel).

[Further details of Simulation Experiment Model development and testing omitted here.]

8.3.3 Development > Simulation Platform > Assumptions

There are no further relevant assumptions made in the simulation platform development.

Chapter 9

Exploration phase 2

CoSMoS pattern: *Exploration:* Use the simulation platform resulting from *Development* to explore the scientific questions established during *Discovery*.

The components are:

- *revisit* the Research Context
- perform *Results Modelling*
- perform a *Simulation Experiment*
- [Argue Instrument Used Appropriately (omitted)]

9.1 Exploration > revisit

The research context is unchanged in the light of *Discovery* and *Development* phase activities.

9.2 Exploration > Results Modelling

CoSMoS pattern: *Results Modelling:* Develop a results model suitable for interpreting simulation experiment data in Domain Model terms.

The relevant components of results modelling are:

- build a *Visualisation Model*
- build a *Results Model*
- [Argue Results Model Appropriate and Consistent (omitted)]

9.2.1 Exploration > Results Modelling > Visualisation Model

- The visualisation employed in increment 1 mimics the single cistrome data in figure 3.2. It removed from increment 2, as it did not prove useful in increment 1.
- Some output data is presented as plots of activation at $T = 1000$ against m, for multiple arenas
- Some output data is presented as plots of activation against time, for multiple arenas

9.2.2 Exploration > Results Modelling > Results Model

The results model remains the cistrome activity (number of active posts) as a function of time.

9.3 Exploration > Simulation Experiment

CoSMoS pattern: *Simulation Experiment:* Use the simulation as a scientific instrument to explore the behaviour of the system.

The relevant components of a simulation experiment are:

- design the experiment
- perform the experiment
- analyse the results

9.3.1 Exploration > Simulation Experiment > design

The parameters p (number of posts) and r (number of red posts) are effectively fixed for any given set of experimentally derived cistrome data (figure 3.8). We can also generate synthetic data to create systems with a range of p and r values to explore general behaviours.

We identify several experiments to perform on the multiple-arena simulation:

Experiment E2.0 : Preliminary investigation of system behaviour with the modified simulation. In each case the arenas have their critical value of the branching parameter as determined in increment 1.

- Simulate pairs of coupled arenas (Nanog/Oct4, Nanog/Sox2, Oct4/Sox2).
- Simulate a three coupled arena (Nanog/Sox2/Oct4).

E2.0	Nanog	Sox2	Oct4
r	631	542	466
m_c	8	7	6
m	m_c	m_c	m_c
s_0	r	r	r

Experiment E2.1 : Test if an individually sustainable Oct4 arena can drive an initially totally dissipated Nanog arena.

E2.1	Nanog	Oct4
r	631	466
m_c	8	6
m	m_c	m_c
s_0	0	r

Experiment E2.1rev : Test if an individually sustainable Nanog arena can drive an initially totally dissipated Oct4 arena.

E2.1rev	Nanog	Oct4
r	631	466
m_c	8	6
m	m_c	m_c
s_0	r	0

Experiment E2.2 : Test system behaviour when the Nanog arena has $m = 2 < m_c$, and the Oct4 arena has $m = 12 \gg m_c$. Both arenas have all red posts initially active.

E2.2	Nanog	Oct4
r	631	466
m_c	8	6
m	2	12
s_0	r	r

Experiment E2.3 : Test 16 different combinations of branching parameter, m, for the Nanog and Oct4 arenas, to see when Oct4 can ignite Nanog.

E2.3	Nanog	Oct4
r	631	466
m_c	8	6
m	$0.5m_c, m_c - 1, m_c, 2m_c$	$0.5m_c, m_c - 1, m_c, 2m_c$
s_0	0	r

Experiment E2.3rev : As for experiment E2.3, but with reversed initial conditions, to see when Nanog can ignite Oct4.

E2.3rev	Nanog	Oct4
r	631	466
m_c	8	6
m	$0.5m_c, m_c - 1, m_c, 2m_c$	$0.5m_c, m_c - 1, m_c, 2m_c$
s_0	r	0

Experiment E2.4 : Test system behaviour when the Nanog arena has no branching, and the Oct4 arena has high branching, to see if Oct4 can sustain a non-branching Nanog. Both arenas have all red posts initially active.

E2.4	Nanog	Oct4
r	631	466
m_c	8	6
m	0	12
s_0	r	r

Experiment E2.5 : Determine the minimum value of m for which sustainable Nanog arena can be ignited via activity in the Oct4 arena.

E2.5	Nanog	Oct4
r	631	466
m_c	8	6
m	0–m_c	m_c
s_0	0	r

Experiment E2.5rev : As for experiment E2.5, but with the roles of Oct4 and Nanog reversed: determine the minimum value of m for which sustainable Oct4 arena can be ignited via activity in the Nanog arena.

E2.5rev	Nanog	Oct4
r	631	466
m_c	8	6
m	m_c	0–m_c
s_0	r	0

Experiment E2.6 : Determine the minimum value of m for which sustainable Nanog arena can be ignited via combined activity in the Sox2 and Oct4 arenas.

E2.6	Nanog	Sox2	Oct4
r	631	542	466
m_c	8	7	6
m	0–m_c	m_c	m_c
s_0	0	r	r

Experiment E2.7 : Determine the minimum value of m for which sustainable Oct4 arena can be ignited via combined activity in the Nanog and Sox2 arenas.

E2.7	Nanog	Sox2	Oct4
r	631	542	466
m_c	8	7	6
m	m_c	m_c	0–m_c
s_0	r	r	0

9.3.2 Exploration > Simulation Experiment > perform

Number of simulation runs

Following the same argument applied in increment 1, we elect to run sample batches of 200 simulation runs.

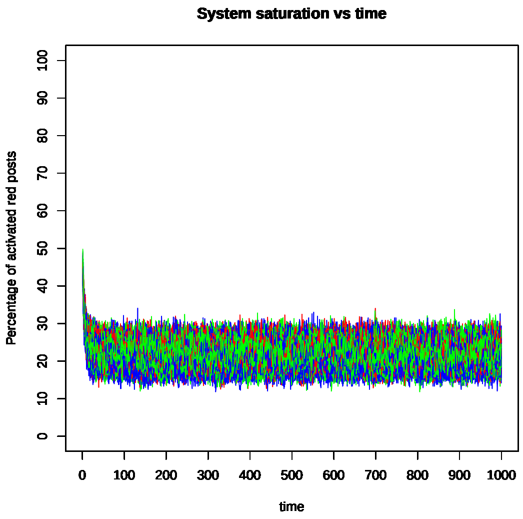

Figure 9.1: E2.0: Coupling three arenas. The level of activation of three core pluripotency TF arenas in a simulation of cistrome communication. The activity traces are all similar: (red) for Nanog, (blue) for Oct4, and (green) for Sox2. $N = 200$ runs overlaid.

Protocol

Each simulation run comprises the p and r values of a particular arena (chosen to match Nanog, Sox2, Oct4 data), with an m value and a starting activity as stated in each individual experiment.

For each simulation run, we record the proportion of active red posts at each timestep, producing a timeseries of the extent of cistrome activation.

For each parameter set (p, r, m, s_0), we run the simulation $N = 200$ times.

9.3.3 Exploration > Simulation Experiment > analyse results

Experiment E2.0

Figure 9.1 shows the behaviour of a three cistrome (arena) model, with the three cistromes in different colours, for clarity. The results for the two arena simulations are not presented here, but all the possible arena pairings show similar behaviour with average activation around 25% for both arenas throughout the duration of the simulation.

Experiment E2.1

Figure 9.2 shows that the Oct4 cistrome critical branching process can drive an initially dissipated Nanog cistrome branching process, through activation of red posts that the two arenas share.

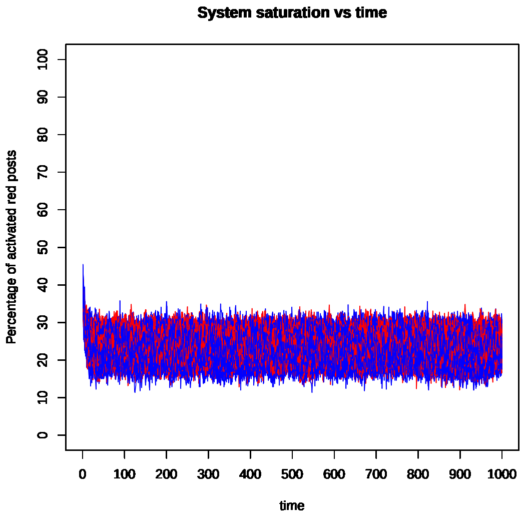

Figure 9.2: E2.1: Oct4 igniting Nanog. Simulation of the interacting Nanog and Oct4 branching processes when both BPs have $m = m_c$, the Oct4 BP has $s_0 = r$ and the Nanog BP, $s_0 = 0$. The activity traces are all similar: (red) for Nanog, (blue) for Oct4. $N = 200$ runs overlaid.

Experiment E2.1rev

Figure 9.3 shows that the Nanog cistrome critical branching process can drive an initially dissipated Oct4 cistrome branching process, through activation of red posts that the two arenas share.

Experiment E2.2

The behaviour of a subcritically branching Nanog arena interacting with a supercritically branching Oct4 arena can be seen in Figure 9.4. If the branching parameter is high enough in the Oct4 arena it can still drive activity in the Nanog arena even with very low values of the branching parameter in this arena.

Experiment E2.3

From Figure 9.5 we can see that the branching process in both arenas is only sustainable if at least one of them has branching parameter m set to a value equal to, or greater than the critical value, m_c.

Experiment E2.3rev

As with Experiment E2.3, we can see from Figure 9.6 we can see that the branching process in both arenas is only sustainable if at least one of them has branching parameter m set to a value equal to, or greater than the critical value, m_c.

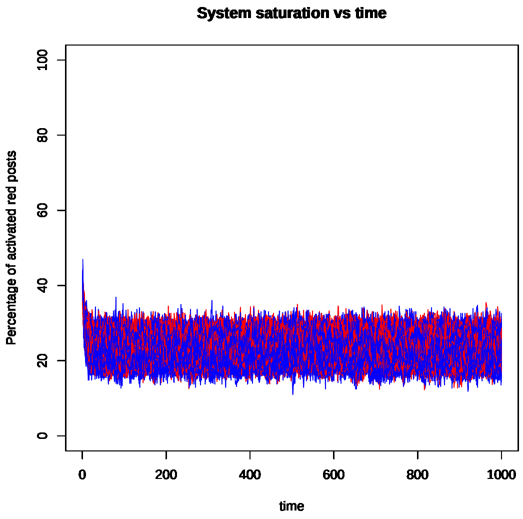

Figure 9.3: E2.1rev: Nanog igniting Oct4. Simulation of the interacting Nanog and Oct4 branching processes when both BPs have $m = m_c$, the Nanog BP has $s_0 = r$ and the Oct4 BP, $s_0 = 0$. The activity traces are all similar: (red) for Nanog, (blue) for Oct4. $N = 200$ runs overlaid.

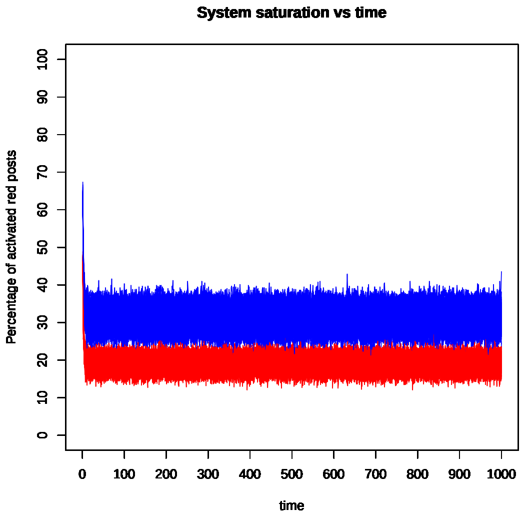

Figure 9.4: E2.2: Oct4 sustaining Nanog. Simulation of the interacting Nanog and Oct4 branching processes when both BPs have $s_0 = r$. The Nanog BP has m set much lower than m_c for the Nanog arena and the Oct4 BP has m set much higher than m_c for the Oct4 arena. The activity traces are well separated: (red) for Nanog, (blue) for Oct4. $N = 200$ runs overlaid.

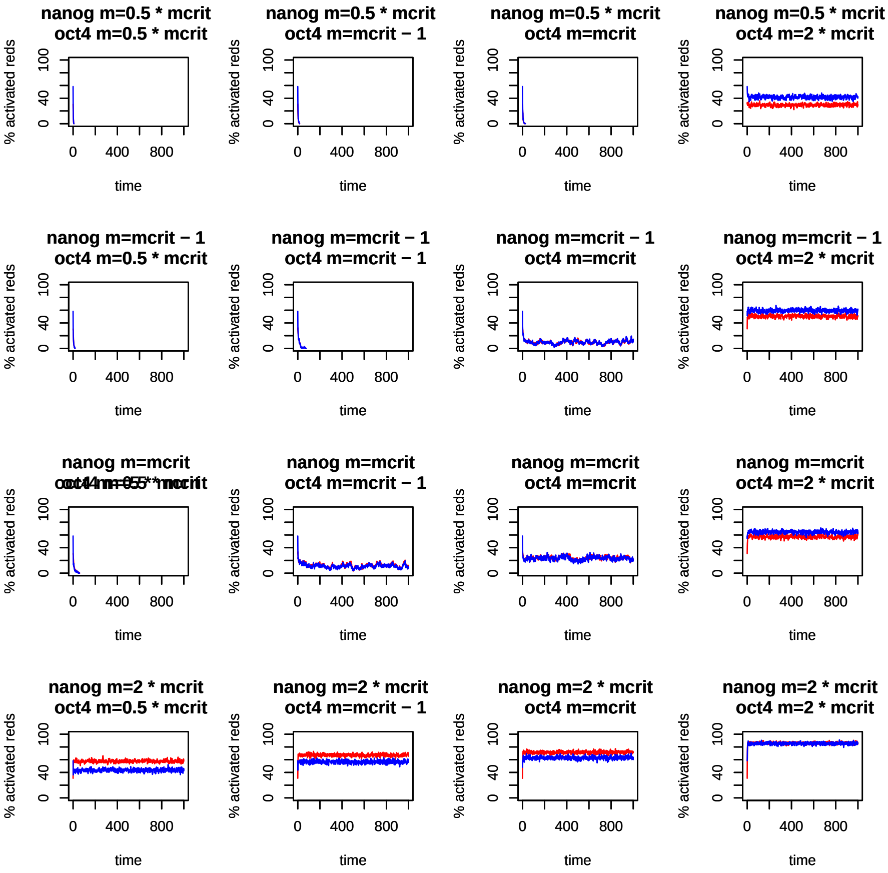

Figure 9.5: E2.3: Oct4 igniting Nanog, varying m. The grid shows the effects of starting the simulation with $s_0 = 0$ for the Nanog arena and $s_0 = r$ for the Oct4 arena. The m values used for each arena are $m_c/2$, $m_c - 1$, m_c and $2m_c$; the combination of values for the branching parameter are indicated above the panel for each set of results in the figure.

Experiment E2.4

The system behaviour when an initially fully activated, but non-branching Nanog arena ($s_0 = r$ and $m = 0$) interacts with a fully activated and super-critically branching Oct4 arena ($s_0 = r$ and $m = 12$, twice the value of m_c determined for the Oct4 arena) is shown in Figure 9.7. The result is comparable to that in Experiment E2.2, see Figure 9.4. The Oct4 arena is still capable of driving short-lived activity in the Nanog cistrome, but this is all derived from the activation of shared red posts.

Experiment E2.5

Varying the value of the branching parameter, m, for the Nanog arena while it is coupled to a critically branching Oct4 arena reveals that the coupled Nanog arena can now sustain branching when $m = 7$, which represents an effective

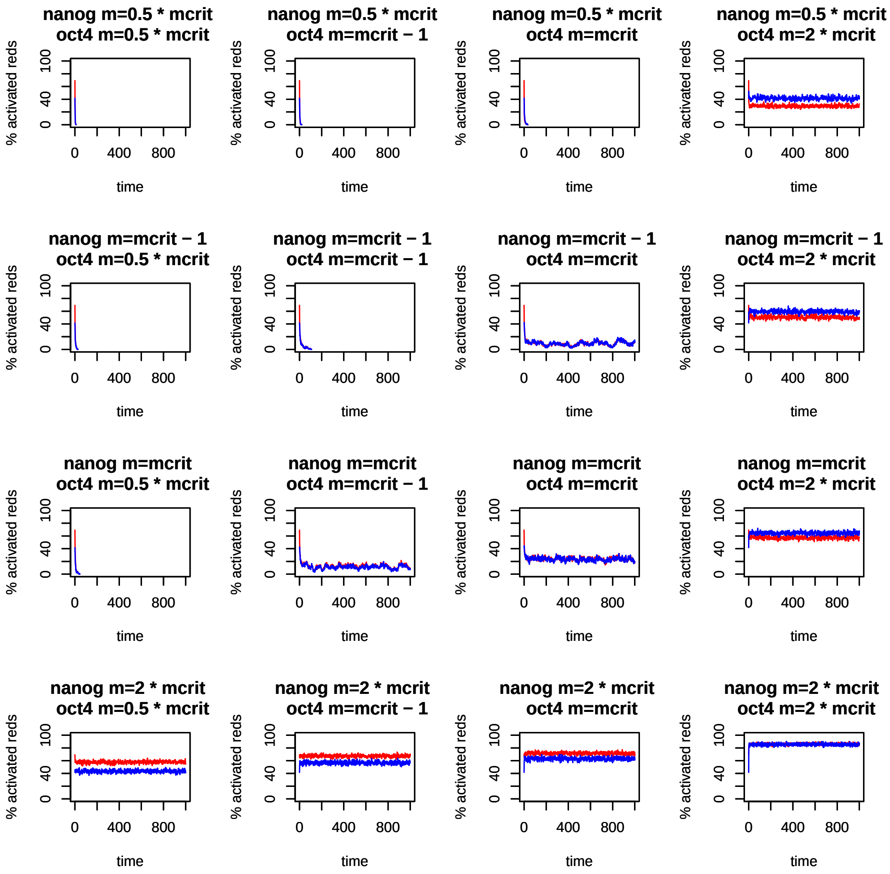

Figure 9.6: E2.3rev: Nanog igniting Oct4, varying m. The grid shows the effects of starting the simulation with $s_0 = r$ for the Nanog arena and $s_0 = 0$ for the Oct4 arena. The m values used for each arena are $m_c/2$, $m_c - 1$, m_c and $2m_c$; the combination of values for the branching parameter are indicated above the panel for each set of results in the figure.

reduction of m_c from 8 to 7 as illustrated in Figure 9.8.

Experiment E2.5rev

The experiment is essentially reversing the roles of the two arenas in Experiment E2.5. Now it is the Nanog arena that is driving the Oct4 arena. From Figure 9.9 we can see that the effect of coupling the Oct4 arena to the Nanog arena is to lower the effective value of m_c from 6 to 5 for the Oct4 BP.

Experiment E2.6

Varying the value of the branching parameter, m, for the Nanog arena while it is coupled to critically branching Oct4 and Sox2 arenas reveals that the coupled Nanog arena can now sustain branching when $m = 6$, which represents an

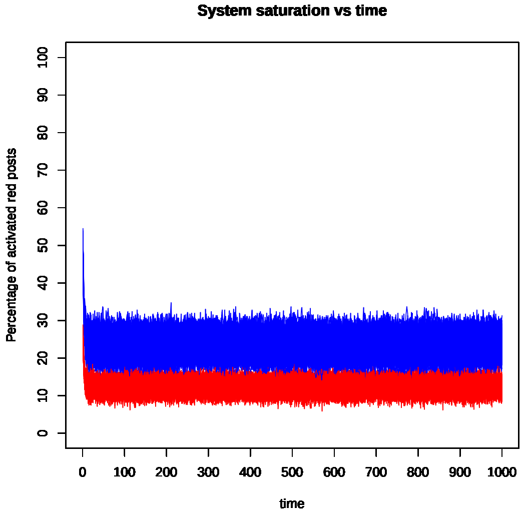

Figure 9.7: E2.4: Oct4 sustaining nonbranching Nanog. Simulation of the interacting Nanog and Oct4 branching processes when both BPs have $s_0 = r$. The Nanog arena has $m = 0$ and the Oct4 arena has $m \gg m_c$. The activity traces are well separated: (red) for Nanog, (blue) for Oct4. $N = 200$ runs overlaid.

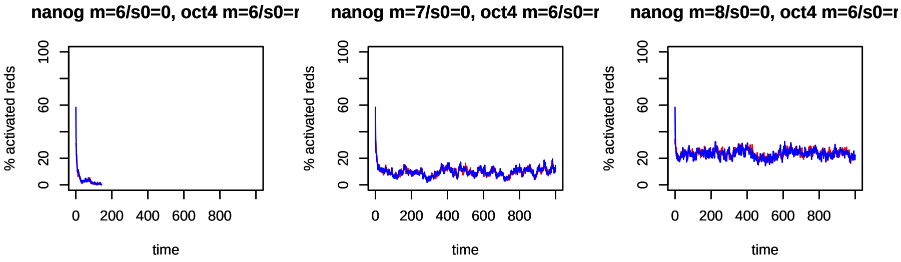

Figure 9.8: E2.5: minimum m for Oct4 to sustain Nanog. Simulation of the interacting Nanog and Oct4 branching processes when the Nanog arena has $m = 0, \ldots, m_c = 8$ and the Oct4 arena has $m = m_c = 6$. (Only $m = 6, 7, 8$ shown; lower m dissipate essentially instantly.) The Nanog arena starts off dissipated ($s_0 = 0$) and the Oct4 arena saturated ($s_0 = r$). The Nanog results are red and the Oct4 results are blue.

effective reduction of m_c from 8 to 6 as illustrated in Figure 9.10.

Experiment E2.7

Varying the value of the branching parameter, m, for the Nanog arena while it is coupled to a critically branching Oct4 arena reveals that the coupled Nanog arena can now sustain branching when $m = 7$, which represents an effective reduction of m_c from 8 to 7 as illustrated in Figure 9.11.

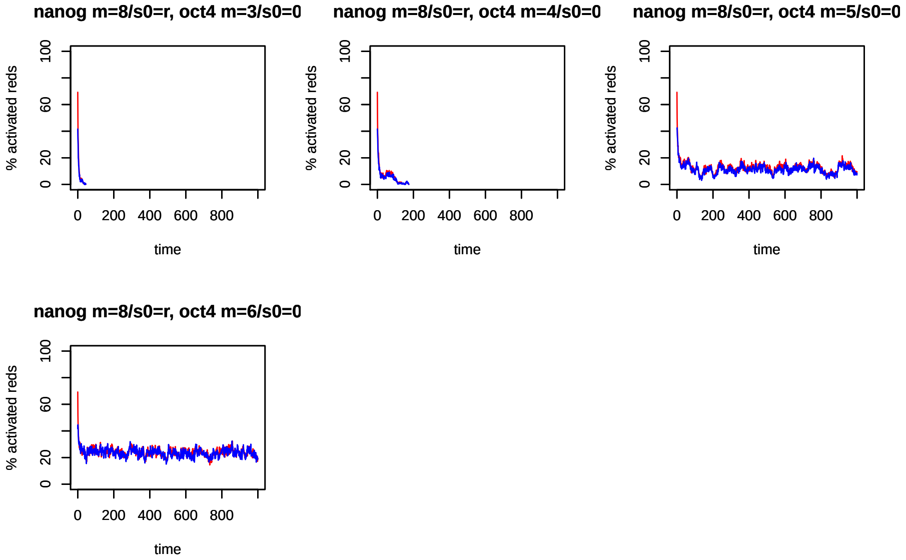

Figure 9.9: E2.5rev: minimum m for Nanog to sustain Oct4. Simulation of the interacting Nanog and Oct4 branching processes when the Nanog arena has $m = m_c = 8$ and the Oct4 arena has $m = 0, \ldots, m_c = 6$. (Only $m = 3, 4, 5, 6$ shown; lower m dissipate essentially instantly.) The Nanog arena starts off saturated ($s_0 = r$) and the Oct4 arena dissipated ($s_0 = 0$). The Nanog results are red and the Oct4 results are blue.

Summary

We observe that coupling cistromes permits an effective lowering of the critical value of the branching parameter, m_c. Coupling the Nanog arena to either the Oct4 or the Sox2 arena lowers its effective m_c by 1 if the other arena is branching at its critical rate m_c. Coupling the Nanog arena to both the Oct4 and Sox2 arenas, with both critically branching, lowers its effective m_c by 2, from 8 to 6.

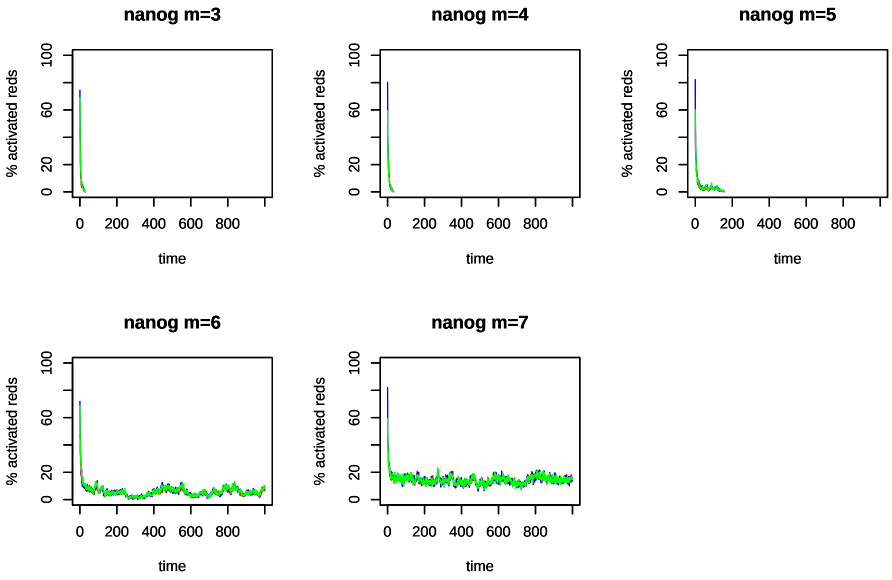

Figure 9.10: E2.6: minimum ms for Oct4 and Sox2 to jointly sustain Nanog. Simulation of the interacting Nanog, Oct4 and Sox2 branching processes when the Nanog arena has $m = 0, \ldots, 7$, the Oct4 arena has $m = m_c = 6$ and the Sox2 arena has $m = m_c = 7$. (Only $m = 3, 4, 5, 6, 7$ shown; lower m dissipate essentially instantly.) The Nanog arena starts off dissipated ($s_0 = 0$) and both the Oct4 and the Sox2 arenas are saturated ($s_0 = r$). The Nanog results are red, the Oct4 results are blue, and the Sox2 results are green.

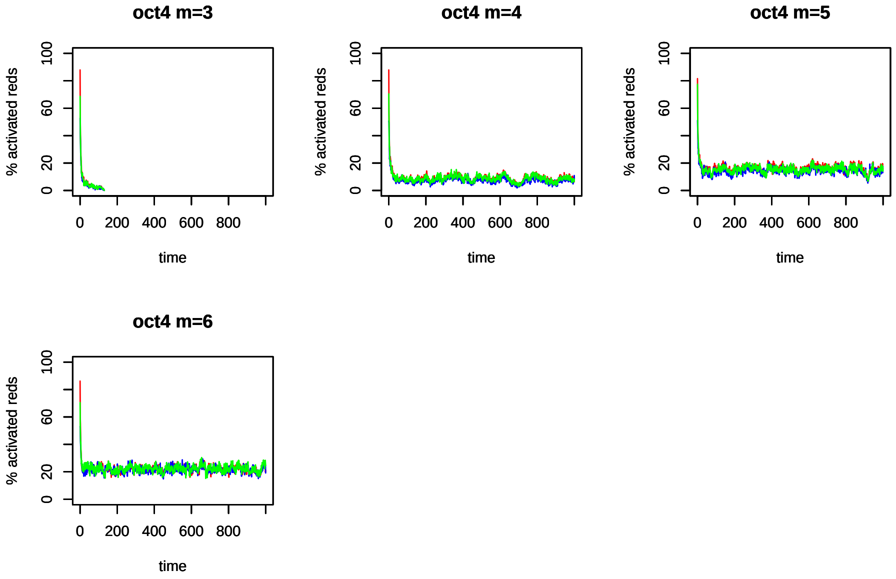

Figure 9.11: E2.7: minimum ms for Nanog and Sox2 to jointly sustain Oct4. Simulation of the interacting Nanog, Oct4 and Sox2 branching processes when the Oct4 arena has $m = 0, \ldots, m_c = 6$, the Nanog arena has $m = m_c = 8$ and the Sox2 arena has $m = m_c = 7$. (Only $m = 3, 4, 5, 6$ shown; lower m dissipate essentially instantly.) The Oct4 arena starts off dissipated ($s_0 = 0$) and both the Nanog and the Sox2 arenas are saturated ($s_0 = r$). The Nanog results are red, the Oct4 results are blue and the Sox2 results are green.

Part III

Increment 3: fourth cistrome

Chapter 10

Increment 3: fourth cistrome

10.1 CoSMoS Simulation Project: increment 3

In this third increment, we further explore the potential of our simulation. The results of increment 2, that coupling arenas can lower the effective m_c, prompts us to investigate whether arenas that share a greater or lesser number of red posts can exert a greater or lesser effect on each others' value of m_c. We experiment with coupling the core pluripotency cistromes, Nanog, Oct4 and Sox2, with a further cistrome that is known to overlap extensively with these: cMyc.

CoSMoS pattern: *CoSMoS Simulation Project:* Develop a basic fit-for-purpose simulation of the complex scientific domain of interest.

The components of a *CoSMoS Simulation Project* are:

- carry out a *Discovery Phase*
- carry out a *Development Phase*
- carry out an *Explorations Phase*

10.2 Discovery phase 3

cMyc was chosen as the additional TF because cMyc is connected with approximately 30,000 target areas throughout the genome. The Domain Scientist believes that this indicates that cMyc represents an evolutionarily very ancient undercurrent that now underpins circuitry self-organization and cell behaviour in very many different ways [Nie et al., 2012; Rothenberg et al., 2010].

cMyc overlaps with the core pluripotency TF cistromes to a similar extent to which they overlap each other. However, the calculated critical value for its branching parameter, m_c, is half that of the Nanog cistrome and lower than that for both the Oct4 and Sox2 cistromes.

We therefore test the extent to which overlaps between the cMyc arena with those of the core pluripotency transcription factors serve to lower m_c for the core TFs when coupled with the cMyc cistrome.

	Nanog	Sox2	Oct4	cMyc
p	4310	3330	2540	2961
r	631	542	466	864
r/p	0.146	0.163	0.183	0.292
m_c calc	6.8	6.1	5.5	3.4
m_c obs	8	7	6	4

Figure 10.1: Data Dictionary: the values of the parameters p (number of posts, or segments in the cistrome) and r (the number of red posts, or red segments in the cistrome) for the TFs investigated in this study. The observed value of m at tipping point, versus calculated value m_c, are shown.

arena	shared red posts
Nanog–Oct4	194
Nanog–Sox2	287
Oct4–Sox2	237
Nanog–cMyc	265
Oct4–cMyc	268
Sox2–cMyc	252

Figure 10.2: Data Dictionary: the number of red posts shared by the arenas

The main activity needed in the Discovery phase is to check that none of the assumptions is invalidated by the inclusion of a fourth, previously unconsidered, cistrome, cMyc. The only assumption that mentions specific cistromes is A.2:

A.2 *It is sufficient to consider only the key pluripotency transcription factors: Nanog, Oct4, Sox2.* It *is* sufficient: we have gained understanding from consideration of these three. We are now moving to larger systems to investigate their behaviour.

There is also an update to the data dictionary, with details of the new cistrome. See figures 10.1 and 10.2

10.3 Development phase 3

None of the development phase assumptions is invalidated by the inclusion of a fourth, previously unconsidered, cistrome, cMyc.

No changes to the models or implementation are needed. Only new data, on cMyc segments, is required. Dietmann provided the necessary ChIP-Seq data for the cMyc cistrome in a suitable format.

10.4 Exploration phase 3

CoSMoS pattern: *Exploration:* Use the simulation platform resulting from *Development* to explore the scientific questions established during *Discovery.*

The components are:
- *revisit* the Research Context
- perform *Results Modelling*
- perform a *Simulation Experiment*
- [Argue Instrument Used Appropriately (omitted)]

10.4.1 Exploration > revisit

The research context is extended to more cistromes, but this has no effect on the assumptions.

10.4.2 Exploration > Results Modelling

There is no change to the Results model in this increment.

10.4.3 Exploration > Simulation Experiment

CoSMoS pattern: *Simulation Experiment:* Use the simulation as a scientific instrument to explore the behaviour of the system.

The relevant components of a simulation experiment are:
- design the experiment
- perform the experiment
- analyse the results

Exploration > Simulation Experiment > design

To run simulations with the cMyc cistrome branching at the critical rate, m_c, we first need to determine the observed value of m_c.

Experiment E3.0 : As in Experiment E.0, determine the critical value, m_c, of the branching parameter m, at which we first observe sustainable activity in the isolated cMyc arena. All red posts initially active, $s_0 = r$.

E3.0	cMyc
p	2961
r	864
m	0–50
s_0	r

Experiment E3.1 : Effect of coupling the Nanog and cMyc cistromes on m_c for the Nanog cistrome.

E3.1	Nanog	cMyc
r	631	864
m_c obs	8	4
m	0–m_c	m_c
s_0	0	r

Experiment E3.2 : Effect of coupling the Oct4 and cMyc cistromes on m_c for the Oct4 cistrome.

E3.2	Oct4	cMyc
r	466	864
m_c obs	6	4
m	0–m_c	m_c
s_0	0	r

Experiment E3.3 : Effect of coupling the Sox2 and cMyc cistromes on m_c for the Sox2 cistrome.

E3.3	Sox2	cMyc
r	542	864
m_c obs	7	4
m	0–m_c	m_c
s_0	0	r

Exploration > Simulation Experiment > perform

The simulation runs and experimental protocol are unchanged for this increment.

Exploration > Simulation Experiment > analyse results

Experiment E3.0 The results of running simulations at values of m between 1 and 50 are shown in Figure §10.3. Figure 10.3 shows that the critical value for sustained activity in the cMyc cistrome, m_c, is 4.

Experiment E3.1 Coupling of the Nanog cistrome with the cMyc cistrome has the effect of reducing m_c from 8 (Nanog cistrome in isolation) to 6 (see Figure 10.4 and Figure 5.1(top)).

Experiment E3.2 Coupling of Oct4 with cMyc has the effect of reducing m_c from 6 (Oct4 in isolation) to 4 (see Figure §10.5 and Figure 5.1(bottom)).

Experiment E3.3 Coupling of Sox2 with cMyc has the effect of reducing m_c from 7 (Sox2 in isolation) to 5 (see Figure 10.6 and Figure 5.1(middle)).

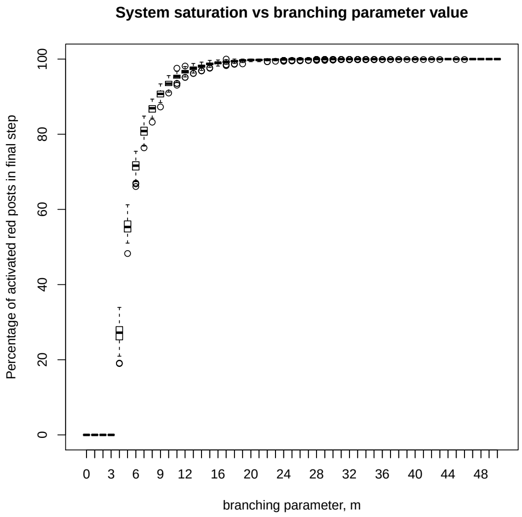

Figure 10.3: E3.0: mCyc observerd m_c. p and r corresponding to cMyc data. $s_0 = r$. Recall m_c calc = 3.4

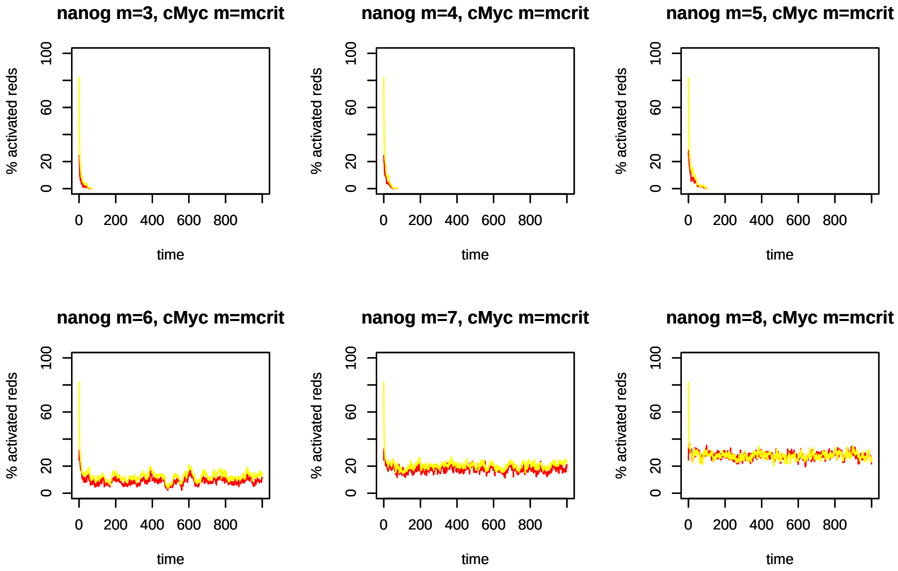

Figure 10.4: E3.1: coupling Nanog and cMyc. Investigating the effect of coupling the Nanog and cMyc arenas on the value of m_c for the Nanog arena. The cMyc arena has $s_0 = r$ and $m = m_c = 4$. The activity of the Nanog arena is red, the cMyc arena is yellow.

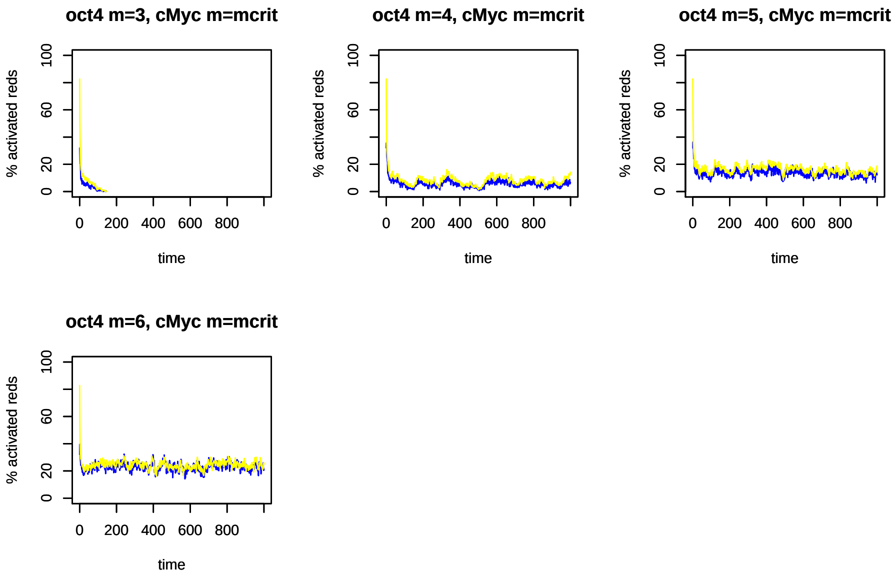

Figure 10.5: E3.2: coupling Oct4 and cMyc. Investigating the effect of coupling the Oct4 and cMyc arenas on the value of m_c for the Oct4 arena. The cMyc arena has $s_0 = r$ and $m = m_c = 4$. The activity of the Oct4 arena is blue, the cMyc arena is yellow.

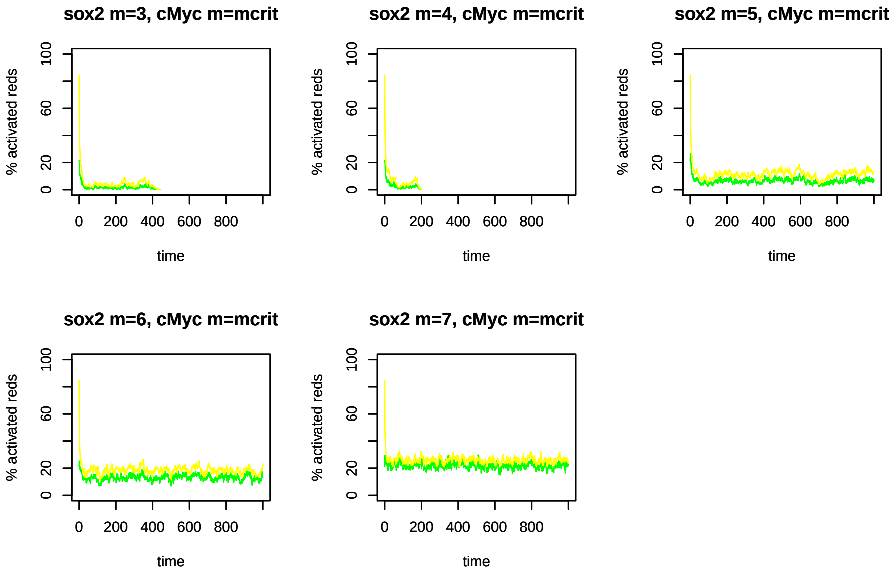

Figure 10.6: E3.3: coupling Sox2 and cMyc. Investigating the effect of coupling the Sox2 and cMyc arenas on the value of m_c for the Sox2 arena. The cMyc arena has $s_0 = r$ and $m = m_c = 4$. The activity of the Sox2 arena is green, the cMyc arena is yellow.

Summary

Similar to the other cistormes, we observe that coupling with cMyc cistrome permits an effective lowering of the critical value of the branching parameter, m_c.

Part IV

Discussion and conclusions

Chapter 11

Discussion and conclusions

11.1 CoSMoS lessons

This report documents and illustrates the use of CoSMoS patterns to perform a CoSMoS simulation project, from initial discovery, through development, to exploration, over three increments. There were several lessons learned about the use of CoSMoS, which are summarised here.

It is not always clear whether information should be included in the Domain, or Domain Model, sections, particularly relating to assumptions. Similarly, some of the preliminary experiments to determine m_c might be considered to be calibrations. What is important, however, is to document the information, rather than to agonise over precisely which section to document it in.

Not all patterns are applicable. For example, here the Domain Model Cartoon had to be presented within the Domain Model section, rather than as a prior illustration. Additionally, the TF BP model is so abstracted from the Domain, that aspects such as the Domain Experiment Model [Andrews and Stepney, 2015] are not relevant. Again, it is more important to follow the spirit of the CoSMoS approach rather than the letter of every pattern.

Not every aspect of the CoSMoS approach needs to be performed with complete rigour. This simulation is not safety critical, so some aspects have been omitted (such as justification of every assumption, and argumentation of fitness-for-purpose). The extra effort needed to complete all aspects should be expended only if it gives benefit. So while increment 2 followed through all the patterns, increment 3 moved lightly over the Discovery and Development patterns, as it was clear they were mostly not needed.

Although the presentation is sequential and hierarchical, the historical process was not. We spent many short iterations, and considerable backtracking (for example, see figure 3.3), before finally fixing on the 'sparking posts' model in increment 1. The CoSMoS patterns define what information should be recorded by the end of the project, but not the order it needs to be produced. Some uses of CoSMoS can apply the patterns in significantly different orders, for example [Andrews and Stepney, 2014].

We might not have arrived at the conceptual sparking posts model without taking an incremental approach. The need to have just a single-cistrome model for this first increment revealed a fundamental misunderstanding that the modellers were having about the background TF BP model.

We were taking an agile approach, producing minimal simulation models and code, and collaboration meetings would often generate interesting but out of current scope ideas. We invented the concept of the "to don't" list: a place to record the ideas for future reference, in a manner that made it clear they were not to be included in the current increment. Some of these ideas also prompted the recognition of assumptions in the current increment.

The Domain Scientist (Halley) was new to the CoSMoS approach at the start of the project, but had previous experience working with modellers using different approaches on other projects. Halley reports that CoSMoS is a flexible tool to produce objective scientific simulations, and allows progress without being funnelled into preconceptions imposed by a specific toolset or implementation approach.

11.2 Summary, Conclusions, Future Work

This work has run through three complete CoSMoS project increments, with differening focuses, producing the first increment of the system: a single cistrome model; a second increment: the multi-cistrome model; and a third increment: extended experiments.

The results from the first increment demonstrate that the single-cistrome model exhibits its tipping point close to the predicted value of m_c (the value of the branching parameter required for branching process activity to be sustained in a particular cistrome), but the tipping is not particularly sharp, and for values of m close to m_c, there is a lot of noise in the system.

The results from the the second increment simulation indicate that cistromes can interact in such a way that they can modify each other's value of m_c and that the extent of the effect is in some non-linear way dependent on the amount of 'overlap' between those coupled cistromes and their respective critical values for the branching parameter. cMyc has twice the effect of Sox2 or Oct4 on m_c of the Nanog cistrome whilst having similar overlap with the Nanog cistrome as Oct4 and Sox2, and an m_c half that for the core pluripotency TFs.

In order to generate results that have genuine biological relevance, it will be necessary to create a simulation of two or more cistromes interacting with each other via the TFs that each produces, with both excitory and inhibitory interactions. Given the groundwork developed in the first two increments, the modelling and simulation work for future developments, such as including inhibitory interactions between TF Branching Processes, should be relatively straightforward.

Beyond this, future increments could include:

- More complex connections within networks of cistromes, including inhibition and negative feedback, combinatorial binding of TFs, and indicators of 3D genomic or chromosomal architecture. The inclusion of inhibition of gene expression is particularly relevant to the process of pluripotency exit, as batteries of differentiation genes are suddenly expressed.
- A Domain Specific Language with which we can describe the network
- TF half life variability

- Epigenetic histone marks that may help to shape circuitry self-organisation
- Combinatorial binding of TFs to enhancer sites that impart transcriptional synergy [Struhl, 2001]
- Multicellular model incorporating cell-cell signalling

The model presented here represents a novel example of self-organisation that may apply to other complex systems. It is of interest from a purely theoretical perspective because it helps to demonstrate how distributed interactions among units result in higher ordered emergent behaviours. Such complexity could provide dynamic templates of organisation upon which natural selection builds additional elaborations [Halley and Winkler, 2008].

Part V

Appendix

Appendix A

Using the Simulation Code

The code for the simulation, batch scripts for running the simulation on an SGE enabled compute cluster, Python scripts for generating real or synthetic cistromes, and example R scripts for processing simulation results into graphical form, are all available on GitHUB at: https://github.com/CellBranch/CellBranch

The project files are made available in several directories. The top level directory contains:

cosmos2015-proceedings.pdf – the Proceedings for the 2015 CoSMoS workshop, held in York, UK. The proceedings include a paper about the CellBranch simulation [Greaves et al., 2015] which is reporduced here in part I.

v1.0b_simulation/ – a directory containing the source code for the CellBranch multi-cistrome simulation, and an executable jar file for use on a compute cluster

cluster_scripts/ a directory of scripts to run batch jobs on the local SGE enabled cluster

Python_data_generation/ – a directory of Python scripts to generate artificial cistrome data and experimental cistrome data from ChIP-Seq data

R_data_analysis/ a directory of R scripts to process the resulting batches of simulation outputs and produce output graphs

The contents of these directories are now described in more detail.

A.1 v1.0b_simulation/

CBSimv1.0b.jar is the executable jar for the simulation and is needed to execute the simulation from the command line or from a cluster batch script. The actual code resides in the directory /src/simv0/ and its subdirectories.

A.2 cluster_scripts/

This directory contains the following files used to execute the simulation on an SGE-enabled cluster:

- do_batch_of_200.sh
- n8_o0_parameters.xml
- do_exp1_batch.sh

A.2.1 n8_o0_parameters.xml

The file n8_o0_parameters.xml is an example of the XML formatted input parameters used to describe the system to be simulated – principally the cistromes to include the simulation, their branching parameters, and how activated they are at step 0. An additional run parameter is the path to the directory where the simulation output should be written.

The <resultFilePath> tag contains a location at which the results directory will be created. On the YARCC at the University of York, this is the users area on the scratch disk /scratch. You need to replace this with the path of a directory you can write to from your compute cluster.

A.2.2 do_batch_of_200.sh

This script submits a batch of 200 jobs using the specified parameters to a Sun Grid Engine-enabled cluster. The simulation expects two command line parameters from the user:

```
qsub ./do_batch_of_200.sh nnn uuu
```

where nnn is the number of runs to perform and uuu can be any unique number. For example, uuu could be the m value for the first cistrome specified in the parameters file. nnn, uuu and the run number $SGE_TASK_ID are used to make up a directory path where the run results are stored.

Run results are stored as space-separated values in a plain text file. The file has a set of 6 columns per cistrome in the model, where the columns contain:

1. the number of red posts unique to this cistrome that are active
2. the cumulative number of red posts unique to this cistrome that are active
3. the number of white posts unique to this cistrome that are active
4. the cumulative number of white posts unique to this cistrome that are active
5. the number of red posts shared by this cistrome and another that are active
6. the cumulative number of red posts shared by this cistrome and another that are active

The script issues a shell command to run a java JAR and so you should replace the $path_to_local_java/ variable with the path to the java that is installed on your local cluster (and which should be the version of java used to compile the JAR too).

A.2.3 do_exp1_batch.sh

This is a script that dumps all the batches of 200 runs needed to complete a given multi-batch experiment onto the cluster.

A.3 Python_data_generation

This directory contains the following files:

- segments.50kb.txt and segments_new.50kb.txt
- gen_cistromes_from_ChIP-Seq.py
- real_nanog_cistrome.xml
- real_oct4_cistrome.xml
- real_sox2_cistrome.xml
- real_cmyc_cistrome.xml
- data_dimensions.xml
- gen_data.py
- gen_overlapping_cistromes.py

A.3.1 segments.50kb.txt and segments_new.50kb.txt

The text files contain experimental ChIP-Seq data for the cistromes we are modelling. Nanog, Oct4 and Sox2 data is in segments.50kb.txt and cMyc data is in segments_new.50kb.txt. Each file is a space separated value file which is read in and processed one line at a time by gen_cistromes_from_ChIP-Seq.py.

A.3.2 gen_cistromes_from_ChIP-Seq.py

This Python script accepts segments.50kb.txt as input, and outputs the requested cistrome in XML format. The script prompts the user with an instruction on how to use the script. The command line arguments are the name of the file containing the ChIP-Seq data, the column containing the number of TF binding sites for the desired TF, the column giving details of the gene products, the name of the output file for the cistrome data in XML format.

A.3.3 The cistrome data files, real_XXX_cistrome.xml

The cistrome data used in our experiments is contained in the files real_nanog_cistrome.xml, real_oct4_cistrome.xml, real_sox2_cistrome.xml and real_cmyc_cistrome.xml.

The XML formatted cistrome data is produced from segments.50kb.txt or segments_new.50kb.txt by gen_cistromes_from_ChIP-Seq.py. Each is a 234 by 234 grid of Segments, each described by an XML node in the cistrome file.

A.3.4 data_dimensions.xml

This XML file contains named parameters for input to gen_data.py. The parameters describe such things as grid size, the number of segments in the cistrome, the number of segments which are 'red' and 'white', etc

A.3.5 gen_data.py

This Python script generates a synthetic cistrome that satisfies the parameters specified in data_dimensions.xml. The script prompts the user with an instruction on how to use the script.

A.3.6 data_dimensions_overlap.xml

This XML file contains named parameters for input to gen_overlapping_cistromes.py. The parameters describe such things as grid size, the number of segments in the cistrome, the number of segments which are 'red' (shared and unshared) and 'white' (shared and unshared), etc

A.3.7 gen_overlapping_cistromes.py

This Python script reads in the parameters in data_dimensions_overlap.xml and generates the multiple cistromes that satisfy these parameters. The script prompts the user with an instruction on how to use the script.

A.4 R_data_analysis

This directory contains the following R scripts used for analysis of simulation outputs.

- do_timeseries.r and do_timeseries_2.r
- do_run_grid.r
- do_three_cistromes_run1_grid.r
- do_boxplot_grid.r
- compile_results.sh

A.4.1 do_timeseries.r and do_timeseries_2.r

These scripts collate the numbers of active posts at each timestep across 200 simulation runs. do_timeseries.r overlays all 200 timeseries from one experiment. do_timeseries_2.r produces 200 separate plots per experiment.

A.4.2 do_run_grid.r

This script produces a grid of timeseries plots – in this instance the first of the 200 runs in each experiment.

A.4.3 do_three_cistromes_run1_grid.r

This script produces a grid of timeseries plots – in this instance the first of the 200 runs in each experiment run on a 3 cistrome system.

A.4.4 do_boxplot_grid.r

This script produces a grid of boxplots. In this instance the boxplots are of the levels of activation of two TFs in the final step of each of 200 simulation runs. These final steps are collated using compile_results.sh.

References

Alexander, C. et al. (1977). *A Pattern Language: towns, buildings, construction.* Oxford University Press.

Anderson, P. W. (1991). Is complexity physics? is it science? what is it? *Physics Today*, 44(7):9.

Andrews, P. S., Polack, F. A. C., Sampson, A. T., Stepney, S., and Timmis, J. (2010). The CoSMoS process, version 0.1: A process for the modelling and simulation of complex systems. Technical Report YCS-2010-453, Department of Computer Science, University of York.

Andrews, P. S. and Stepney, S. (2014). Using CoSMoS to reverse engineer a domain model for Aevol. In *Proceedings of the 2014 Workshop on Complex Systems Modelling and Simulation, New York, USA, July 2014*, pages 61–79. Luniver Press.

Andrews, P. S. and Stepney, S. (2015). The CoSMoS Domain Experiment Model. In Stepney and Andrews [2015b].

Andrews, P. S., Stepney, S., and Timmis, J. (2012). Simulation as a scientific instrument. In Stepney et al. [2012], pages 1–10.

Ay, A. and Arnosti, D. N. (2011). Mathematical modeling of gene expression: a guide for the perplexed biologist. *Critical Reviews in Biochemistry and Molecular Biology*, 46(2):137–151.

Bak, P. and Paczuski, M. (1993). Why nature is complex. *Physics World*, 6(12):39–43.

Bak, P. and Paczuski, M. (1995). Complexity, contingency, and criticality. *Proceedings of the National Academy of Science USA*, 92:6689–6696.

Ball, P. (1999). Transitions still to be made. *Nature*, 402:C73–C76.

Ball, P. (2001). *The Self-Made Tapestry.* Oxford University Press.

Bornholdt, S. (2005). Less is more in modeling large genetic networks. *Science*, 310:449–451.

Boyer, L. A., Lee, T. I., Cole, M. F., Johnstone, S. E., Levine, S. S., Zucker, J. P., Guenther, M. G., Kumar, R. M., Murray, H. L., Jenner, R. G., Gifford, D. K., Melton, D. A., Jaenisch, R., and Young, R. A. (2005). Core transcriptional regulatory circuitry in human embryonic stem cells. *Cell*, 122:947–956.

Bray, D. (2003). Molecular networks: the top-down view. *Science*, 301:1864–1865.

Chaisson, E. (2004). Complexity: An energetics agenda. *Complexity*, 9(3):14–21.

Crutchfield, J. P., Farmer, J. D., Packard, N. H., and Shaw, R. S. (1986). Chaos. *Scientific American*, 255(6):38–49.

Driscoll, M. E. (2009). Is big data at a tipping point? http://www.analyticbridge.com/profiles/blogs/is-big-data-at-a-tipping-point. [Accessed: 2015-05-01].

Ekeland, I. (2002). In the balance. *Nature*, 417:385.

Farmer, J. D. and Packard, N. H. (1986). Evolution, games, and learning: Models for adaptation in machines and nature. An introduction to the proceedings of the CNLS Conference, Los Alamos, May 1985. *Physica D*, 22:vii–xii.

Ferrell, J. (2009). Q&A: Systems biology. *Journal of Biology*, 28.

Gallagher, R. and Appenzeller, T. (1999). Beyond reductionism. *Science*, 284(5411):79.

Gollub, J. P. and Langer, J. S. (1999). Pattern formation in nonequilibrium physics. *Reviews of Modern Physics*, 71(2):S396–S403.

Greaves, R. B., Dietmann, S., Smith, A., Stepney, S., and Halley, J. D. (2015). Genome-wide mouse embryonic stem cell regulatory network self-organisation: a big data CoSMoS computational modelling approach. In Stepney and Andrews [2015b], pages 31–66.

Halley, J. D., Burden, F. R., and Winkler, D. A. (2009). Stem cell decision making and critical-like exploratory networks. *Stem Cell Research*, 2(3):165–177.

Halley, J. D., Smith-Miles, K., et al. (2012). Self-organizing circuitry and emergent computation in mouse embryonic stem cells. *Stem Cell Research*, 8(2):324–333.

Halley, J. D. and Winkler, D. A. (2008). Critical-like self-organization and natural selection: Two facets of a single evolutionary process? *BioSystems*, 92(2):148–158.

Hartwell, L. H., Hopfield, J. J., Leibler, S., and Murray, A. W. (1999). From molecular to modular cell biology. *Nature*, 402:C47–C52.

Howe, D., Costanzo, M., et al. (2008). Big data: The future of biocuration. *Nature*, 455(7209):47–50.

Kadanoff, L. P. (1987). Chaos: A view of complexity in the physical sciences. In *From Order to Chaos II Essays: Critical Chaotic and Otherwise*. World Scientific.

Kalkan, T. and Smith, A. (2014). Mapping the route from naive pluripotency to lineage specification. *Phil. Trans. R. Soc. B*, 369:20130540.

Kauffman, S. (1995). *At Home in the Universe*. Oxford University Press.

Kim, J., Chu, J., Shen, X., Wang, J., and Orkin, S. H. (2008). An extended transcriptional network for pluripotency of embryonic stem cells. *Cell*, 132(6):1049–1061.

Lander, A. D. (2010). The edges of understanding. *BMC Biology*, 8(1):40.

Loh, Y.-H., Wu, Q., Chew, J.-L., Vega, V. B., Zhang, W., Chen, X., Bourque, G., George, J., Leong, B., Liu, J., Wong, K.-Y., Sung, K. W., Lee, C. W. H., Zhao, X.-D., Chiu, K.-P., Lipovich, L., Kuznetsov, V. A., Robson, P., Stanton, L. W., Wei, C.-L., Ruan, Y., Lim, B., and Ng, H.-H. (2006). The Oct4 and Nanog transcription network regulates pluripotency in mouse embryonic stem cells. *Nature Genetics*, 38(4):431–440.

MacArthur, B. D., Maayan, A., and Lemischka, I. R. (2008). Toward stem cell systems biology: From molecules to networks and landscapes. *Cold Spring Harbor Symposia on Quantitative Biology*, 73:211–215.

Martello, G. and Smith, A. (2014). The nature of embryonic stem cells. *Annual Review of Cell and Developmental Biology*, 30:647–675.

Mesarovic, M. D., Sreenath, S. N., and Keene, J. D. (2004). Search for organising principles: understanding in systems biology. *Systems Biology*, 1(1):19–27.

Muñoz Descalzo, S., Rué, P., et al. (2013). A competitive protein interaction network buffers Oct4-mediated differentiation to promote pluripotency in embryonic stem cells. *Molecular Systems Biology*, 9:694.

Nichols, J. and Smith, A. (2009). Naive and primed pluripotent states. *Cell Stem Cell*, 4(6):487–492.

Nicolis, G. and Prigogine, I. (1977). *Self-Organization in Nonequilibrium Systems*. John Wiley & Sons.

Nie, Z., Hu, G., Wei, G., Cui, K., Yamane, A., Resch, W., Wang, R., Green, D., Tessarollo, L., Casellas, R., Zhao, K., and Levens, D. (2012). Naive and primed pluripotent states. *Cell*, 151(1):68–79.

Niwa, H., Miyazaki, J.-i., and Smith, A. G. (2000). Quantitative expression of Oct-3/4 defines differentiation, dedifferentiation or self-renewal of ES cells. *Nature Genetics*, 24(4):372–6.

Parisi, G. (1993). Statistical physics and biology. *Physics World*, 6:42–47.

Rothenberg, M., Clarke, M., and Diehn, M. (2010). The myc connection: Es cells and cancer. *Cell*, 143:184–186.

Stepney, S. (2012). A pattern language for scientific simulations. In Stepney et al. [2012], pages 77–103.

Stepney, S. and Andrews, P. S. (2015a). CoSMoS special issue editorial. *Natural Computing*, 14:1–6.

Stepney, S. and Andrews, P. S., editors (2015b). *Proceedings of the 2015 Workshop on Complex Systems Modelling and Simulation, York, UK, July 2015*. Luniver Press.

Stepney, S., Andrews, P. S., and Read, M., editors (2012). *Proceedings of the 2012 Workshop on Complex Systems Modelling and Simulation, Orleans, France, September 2012*. Luniver Press.

Stepney, S. et al. (2016). *Engineering Simulations as Scientific Instruments*. Springer. [in prep].

Struhl, K. (2001). Gene regulation. a paradigm for precision. *Science*, 293:10541055.

Teles, J., Pina, C., et al. (2013). Transcriptional regulation of lineage commitment – a stochastic model of cell fate decisions. *PLOS Computational Biology*, 9(8):e1003197.

Vicsek, T. (2002). The bigger picture. *Nature*, 418:131.

Weatherall, D. J. (2001). Phenotype-genotype relationship in monogenic disease: lessons from the Thalassemias. *Nature Reviews Genetics*, 2:245–255.

Xu, H., Schaniel, C., Lemischka, I. R., and Ma'ayan, A. (2010). Toward a complete in silico, multi-layered embryonic stem cell regulatory network. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 2(6):708–733.

Zandstra, P. and Clarke, G. (2014). Computational modeling and stem cell engineering. In Nerem, R. M., Loring, J., et al., editors, *Stem Cell Engineering*, pages 65–97. Springer.