

Odefying Kirkham

Simon Johanning

July 21, 2016

Contents

1	Introduction	1
1.1	Background	1
1.2	Kirkham paper	2
1.2.1	Motivation	2
1.2.2	Structure	2
2	Theory	3
2.1	Boolean Models	3
2.1.1	Formal description	4
2.2	General approach to make discrete models continuous	4
2.2.1	HillCube	5
2.3	Odefy	6
3	Results	6
3.1	Creation of the BM	7
3.1.1	Model creation	7
3.2	Simulation of continuous models	9
4	Discussion	10
5	Representation of gene regulatory networks	10

1 Introduction

1.1 Background

This course work has been written in the context of the course "Modellierung biologischer und molekularer Systeme" (WS 2015/16) at the University of

Leipzig.

It is based on a presentation of the paper *Early gene regulation of osteogenesis in embryonic stem cells* by G.R. Kirkham et. al. (see [2]) in the context of aforementioned course.

It will lay out the basic ideas of the paper (1.2), give an overview of the mathematical ideas the methodology is based upon (2), and will discuss results from the investigation of the boolean networks (3). It will close with a discussion of the paper and the derived results (4).

Since the course associated with this course work is about (mathematical) modeling, and the mathematical techniques used in the paper were merely sketched, this report will focus mostly on the mathematical ideas and modeling aspects employed in the paper, and will only treat the biological context of [2] on the side.

1.2 Kirkham paper

1.2.1 Motivation

Motivated by the knowledge gap of networks regulating the differentiation of mouse embryonic stem cells into bone cells, the authors of [2], are interested in understanding the regulatory mechanisms of osteogenesis. Since mathematical models of gene regulatory networks (GRNs) have proven to characterize GRNs well, as well as making novel predictions, especially in the case of incomplete data, the authors investigate mathematical models suited for this application.

Since boolean models (BMs) capture the qualitative dynamics of biological systems in a number of studies, the authors chose to start out modeling the GRN under investigation with boolean models. In order to capture the quantitative dynamics of the system under investigation, they then transform these BMs into their continuous homologue functions in order to describe the system as a set of ordinary differential equations (ODE), a process call "odefication", using the Odefy toolbox.

This process is chosen, since the description of a system through ODEs captures the dynamics of a system very well, and helps the authors to identify a (unique) GRN characterizing the influence of the transcription and growth factors under consideration.

1.2.2 Structure

As noted above, [2] investigates GRNs relevant to osteogenesis in embryonic stem (ES) cells. Based on existing research on osteogenetic differentiation

of mouse ES cells, they derive a set of three genes and two growth factors (GFs) to be of particular interest for this GRN, and aim to identify the relationship between them. These are the genes *Runx2*, *Dlx5*, *Msx2* and the GFs *TGF β 1* and *BMP2*.

Consequently the authors test different GRNs involving these genes and GFs, narrowing down the possible candidates.

Starting from modeling the GRNs as Boolean Models, they transform these into differential equations ("odefication") fitted to measured data in order to test predictions about the GRNs behaviour under over- and under-expression of the GFs.

Their experimental data comprises a set of measurements of the TFs of interest (*Dlx5*, *Msx2*, *Runx2*) at 5 fixed time points, exposed to the GFs *BMP2*, *TGF β 1* and a combination of these. The expression levels of the TFs were used to reduce the number of candidate GRNs through comparing the expression profiles to the stable states of the GRNs.

The remaining GRNs (i.e. the ones matching the experimental expression profiles), were then transformed into continuous ODEs, and their over- and underexpression behaviour was compared to experimental results, ruling out all but one network.

The paper closes with a discussion and the description of the experimental setup.

2 Theory

2.1 Boolean Models

As mentioned in the Introduction, [2] starts out with boolean models of the species in the GRN. Boolean models (BMs) express qualitative biological knowledge, meaning that a species is either active or inactive (binary state). Represented as a graph (see figure 1), the species are seen as nodes, whereas the interactions between the species are modeled as edges (which can be either activating, inhibitory or no influence). Despite their coarseness, BMs can reproduce the qualitative behaviour of biological systems. Due to their discrete nature however, they are not suited for quantitative models. Thus they can neither explain nor predict quantitative experiments, which is increasingly important in systems biology. For these applications, the transformation to systems of ordinary differential equations (ODEs) has shown to be fruitful.

2.1.1 Formal description

Formally boolean models for N species X_1, X_2, \dots, X_N , represented by variables $x_i \in \{0, 1\}$, are defined by a set of species $R_i := \{X_{i_1}, X_{i_2}, \dots, X_{i_{N_i}}\} \subset \{X_1, \dots, X_N\}$ influencing X_i , as well as an update function $B_i : \{0, 1\}^{N_i} \rightarrow \{0, 1\}$ for every combination of $(x_{i_1}, \dots, x_{i_2}, x_{i_{N_i}}) \in \{0, 1\}^{N_i}$.

Mapped on the (hyper-) unit cube, the nodes $(\xi_{i_1}, \xi_{i_2}, \dots, \xi_{i_{N_i}}) \in \{0, 1\}^{N_i}$ corresponding to the domain of B_i , can logically be interpreted as $(\bigwedge_{ij|\xi_{ij}=1} x_{ij}) \wedge (\bigwedge_{ij|\xi_{ij}=0} \neg x_{ij})$, meaning that B_i will be true for the nodes of the cube where the coordinate equals the boolean value of influence of the activation function. With the sum-of-product representation, B_i can now be viewed as: $B_i(x_{i_1}, x_{i_2}, \dots, x_{i_{N_i}}) = \bigvee_{(\xi_{i_1}, \xi_{i_2}, \dots, \xi_{i_{N_i}}) | B_i(\xi_{i_1}, \xi_{i_2}, \dots, \xi_{i_{N_i}})=1} [(\bigwedge_{ij|\xi_{ij}=1} x_{ij}) \wedge (\bigwedge_{ij|\xi_{ij}=0} \neg x_{ij})]$.

Thus each product can be represented as a hyperedge between the start nodes $S \subset \{X_{i_1}, X_{i_2}, \dots, X_{i_{N_i}}\}$ and the end node X_i , with each pair (s, X_i) , $s \in S$ carrying a sign on whether it is noted as a factor x_{ij} or $\neg x_{ij}$ in the BM.

2.2 General approach to make discrete models continuous

In order to derive continuous models from the BMs, continuous homologues of B_i have to be found. For this, the discrete variables x_i are replaced by continuous variables $\bar{x}_i \in [0, 1]$, yielding the *continuous homologue* functions $\bar{B}_i : [0, 1]^{N_i} \rightarrow [0, 1]$ on the unit interval.

Using the continuous homologues of the BMs, the behavior of the species can be described through the differential equation

$$\dot{\bar{x}}_i = \frac{1}{\tau_i} (\bar{B}_i(\bar{x}_{i_1}, \bar{x}_{i_2}, \dots, \bar{x}_{i_{N_i}}) - \bar{x}_i),$$

where the production of \bar{x}_i is given by \bar{B}_i , and $\tau_i := \frac{1}{\gamma_i}$ stands for the life-time of species X_i , with γ_i as the decay rate of X_i .

For the (in respect to the BM) homologue continuous functions \bar{B}_i , three properties need to hold (as noted in [1]):

- **Accuracy:** \bar{B}_i and B_i need to agree on the vertices of the unit cube $(\{0, 1\}^{N_i})$ ¹
- **Good analytical properties** such as smoothness, so a mathematical analysis as a system of ODEs can be performed
- **Minimality and uniqueness:** \bar{B}_i should be the unique minimal solutions in their interpolation class

¹ \bar{B}_i are then called *perfect* continuous homologues, exhibiting similar steady-state behavior

2.2.1 HillCube

In order to achieve the desired properties sketched above, normalized *HillCubes* are constructed. HillCubes employ *Hill functions* on the edges of the unit cube described above for the functional behavior of the interpolations. As a first step to arrive at (normalized) HillCubes, *BooleCubes* are constructed. These are functions that interpolate B_i linearly through the use of multivariate polynomial interpolation. In order to get a unique solution, the function with the minimal degree (of the polynomial) is chosen. This function satisfies the properties set out in 2.2.

The functions $\overline{B}_i^I(\overline{x}_{i_1}, \overline{x}_{i_2}, \dots, \overline{x}_{i_{N_i}})$ define a system of ODEs that describes the temporal development of \overline{x}_i through the equations

$$\dot{\overline{x}}_i = \frac{1}{\tau_i} (\overline{B}_i^I(\overline{x}_{i_1}, \overline{x}_{i_2}, \dots, \overline{x}_{i_{N_i}}) - \overline{x}_i).$$

These *BooleCube* functions are affine multilinear². However, molecular interactions exhibit switch-like behavior rather than affine multilinear behavior. In order to model this, (sigmoid shape) *Hill functions* ($f(\overline{x}) = \frac{\overline{x}^n}{\overline{x}^n + k^n}$) are used.

These functions are parameterized by the *Hill coefficient* (n), that determines the slope of the curve and measures the cooperativity of the interaction, and the threshold where a species is considered as 'on' in the BM (k), thus where the value of activation is half maximal. Mathematically, the HillCube functions \overline{B}_i^H are described through

$$\overline{B}_i^H(\overline{x}_{i_1}, \overline{x}_{i_2}, \dots, \overline{x}_{i_{N_i}}) := \overline{B}_i^I(f_{i_1}(\overline{x}_{i_1}), f_{i_2}(\overline{x}_{i_2}), \dots, f_{i_{N_i}}(\overline{x}_{i_{N_i}})).$$

Since the Hill functions approach 1 only asymptotically, HillCubes are not perfect homologues of B_i , and not all desired properties sketched in 2.2 are fulfilled. Thus, in the odefy-approach, the Hill functions are subsequently normalized on the unit interval. HillCubes $\overline{B}_i^{H_n}$ are defined by $\overline{B}_i^{H_n}(\overline{x}_{i_1}, \overline{x}_{i_2}, \dots, \overline{x}_{i_{N_i}}) := \overline{B}_i^I(\frac{f_{i_1}(\overline{x}_{i_1})}{f_{i_1}(1)}, \frac{f_{i_2}(\overline{x}_{i_2})}{f_{i_2}(1)}, \dots, \frac{f_{i_{N_i}}(\overline{x}_{i_{N_i}})}{f_{i_{N_i}}(1)})$.

The value of the Hill function f_{i_j} on the value 1 thus normalizes the HillCubes on the unit interval. Normalized HillCubes yield a perfect continuous homologue of the Boolean functions B_i . Thus, a steady-state of the BM will also be a steady-state of the continuous system³, as shown in [1]. However, the authors note that this method does not accurately transform the Boolean update rule into a continuous activation function.

This is due to a systematic difference between the Boolean logic and the

² $1 \leq j \leq N_i, \overline{x}_{ik}, k \neq j$ fixed: $\exists a, b \in \mathbb{R} : \overline{B}_i^I(\overline{x}_{i_1}, \overline{x}_{i_2}, \dots, \overline{x}_{i_{N_i}}) = a + b\overline{x}_{ij}$

³ This exemplifies the importance of normalization, since steady-states of the BM are not (necessarily) steady-states of the non-normalized HillCubes.

analytic form of the activation function (see [1]). For the HillCube models, the imperfect agreement is caused by the asymptotic behavior of the Hill functions, and thus can be made arbitrarily small (through the appropriate choice of parameters), justifying the described approach.

2.3 Odefy

Odefy, a MATLAB and GNU Octave compatible toolbox for the "odefication" (transformation into continuous ODEs) of boolean models, has been developed by the Institute for Computational Biology of the Helmholtz-Zentrum München.

ODEs allow for a more detailed and quantitative characterization of GRNs. Qualitative (biological) knowledge is represented as BMs, which are transformed into continuous model, that are fitted with quantitative data (see figure 2).

Odefy uses the HillCube modeling technique. With \bar{B}_i as the Hill-Cube model of \bar{x}_i , the following differential equation is reached: $\dot{\bar{x}}_i = \frac{1}{\tau_i}(\bar{B}_i(\bar{x}_{i_1}, \bar{x}_{i_2}, \dots, \bar{x}_{i_{N_i}}) - \bar{x}_i)$, where τ_i stands for the lifespan of species X_i (as described in 2.2).

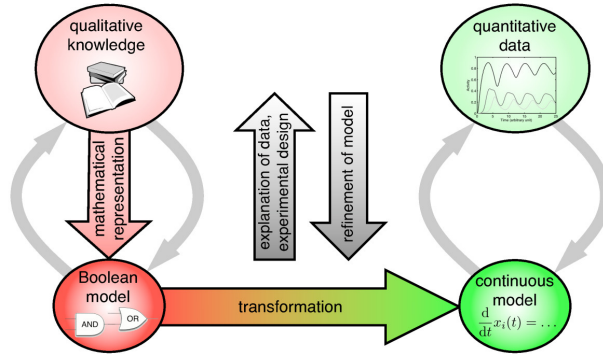


Figure 1: Sketch of the odefy workflow.

3 Results

The simulations were realized with the Odefy 1.19 toolbox in GNU Octave 4.0.0. To follow the workflow of the study, first the candidate networks depicted in figure 1 were created. Subsequently the continuous models were simulated.

3.1 Creation of the BM

The candidate BM, as derived from the literature (see also figure 1), use the GFs BMP2 and TGF β 1, as well as the genes Dlx5, Msx2 and Runx2. Since the authors assumed that the influence of TGF β 1 to Msx2, BMP2 to Msx2 and Dlx5, as well as the influence of Dlx5 on Runx2 was activatory, the following substrings are fixed in the model description (odefy-syntax): 'Msx2 = TGF β 1 || MBP2', 'Dlx5 = BMP2', 'Runx2 = Dlx5'⁴. To model the unclear relations under investigation, 5 variables are used, where the value -1 represents an inhibitory relation, 0 no influence of the first factor onto the second and 1 as activatory influence. The variables used are as follows:

variable	biological meaning	value
Msx2Dlx5	Influence of Msx2 on Dlx5	(index / 3 ⁰) % 3 - 1
Dlx5Msx2	Influence of Dlx5 on Msx2	(index / 3 ¹) % 3 - 1
Msx2Runx2	Influence of Msx2 on Runx2	(index / 3 ²) % 3 - 1
TGFb1Runx2	Influence of TFG β 1 on Runx2	(index / 3 ³) % 3 - 1
Runx2Dlx5	Influence of Runx2 on Dlx5	(index / 3 ⁴) % 3 - 1

3.1.1 Model creation

The BMs specified above are 'odified' through the *ExpressionsToOdefy* command of the odefy toolbox. This command requires an expression describing the boolean network. For every candidate configuration of the GRN (see figure 1), the interactions of the species are modeled. For all three genes, the relevant interactions are either added as activatory (x), left out or added as inhibitory (~x).

For this project, default parameters were chosen⁵, since no parameters were reported in [2], and it is to be expected that default parameters were used in the publication. The following source code exemplifies how the expressions are constructed:

```
models = [0:242]
for index = [0 : 242]

# behavior of Msx2
Msx2String = "Msx2 = TGFb1 || BMP2";
if (rem(floor(index / 3), 3) - 1 == -1)
```

⁴Meaning that for LHS = RHS the LHS is considered 'on' iff the RHS is (and 'off' otherwise).

⁵n=3, k=0.5, $\tau=1$, according to [3]

```

        Msx2String = [Msx2String, " || ~Dlx5"];
elseif(rem(floor(index / 3), 3) - 1 == 1)
        Msx2String = [Msx2String, " || Dlx5"];
endif

# behavior of Dlx5
Dlx5String = "Dlx5 = BMP2"
# influence of Msx2
if(rem(floor(index), 3) - 1 == -1)
        Dlx5String = [Dlx5String, " || ~Msx2"]
elseif(rem(floor(index), 3) - 1 == 1)
        Dlx5String = [Dlx5String, " || Msx2"]
endif
# influence of Runx2
if(rem(floor(index / 81), 3) - 1 == -1)
        Dlx5String = [Dlx5String, " || ~Runx2"]
elseif(rem(floor(index / 81), 3) - 1 == 1)
        Dlx5String = [Dlx5String, " || Runx2"]
endif

# behavior of Runx2
Runx2String = "Runx2 = Dlx5"
# influence of Msx2
if(rem(floor(index / 9), 3) - 1 == -1)
        Runx2String = [Runx2String " || ~Msx2"]
elseif(rem(floor(index / 9), 3) - 1 == 1)
        Runx2String = [Runx2String " || Msx2"]
endif
# influence of TGFB1
if(rem(floor(index / 27), 3) - 1 == -1)
        Runx2String = [Runx2String " || ~TGFB1"]
elseif(rem(floor(index / 27), 3) - 1 == 1)
        Runx2String = [Runx2String " || TGFB1"]
endif

# construct model string
expressionString = ["BMP2 = <>,", " TGFB1 = <>, ", Msx2String, ", "]
# construct BM for odefy
expression = {"BMP2 = <>,", " TGFB1 = <> ", Msx2String, Dlx5String, Runx2String}
model = ExpressionsToOdefy(expression);

```



```
models(index) = model
endfor
```

As mentioned in [3], the boolean update functions are accessible as multidimensional arrays (hypercubes of edge length two). These are accessible through

```
model.tables(x).inspecies
```

for a list of species that influence the species, and

```
model.tables(x).truth
```

for the truth table for the expression the model was based upon. In the above code snippets, *x* stands for the index of the table struct to be accessed. Using *ExpressionToOdefy*, the full BM is stored as an Odefy-internal representation with Octave.

3.2 Simulation of continuous models

After the construction of the BM sketched above, simulations of the networks are performed using the *OdefySimulation* ommand. This command takes a simstruct as parameter which has to be constructed from the models using the *CreateSimstruct* command. The code thus is as follows:

```
# construct struct for simulation
simstruct = CreateSimstruct(model);
simstruct.type = 'hillcubenorm';
# set presence of BMP2
simstruct = SetInitialValue(simstruct, 'BMP2', 1);
# set presence of TGFβ1
simstruct = SetInitialValue(simstruct, 'TGFβ1', 0);
# simulate
simres = OdefySimulation(simstruct);
plot(simres);
legend(model.species);
xlabel('time');
ylabel('expression')
```

The *simstruct* is a data structure that holds the information required for the simulation, like initial states, simulation type and parameters.

In the example above, the initial values of the GFs are set arbitrarily to 'on' (BMP2) and 'off' (TGFβ1) to exemplify the simulation run in the *BMP2 only* environment. This is set purely for illustrative purposes, and all three

setups of the environment presented in [2] are employed in the simulation. The *type* attribute of the *simstruct* specifies that the normalized HillCube variant is employed.

The simulation is invoked by the *OdefySimulation* function that takes the *simstruct* as parameter. Consequently, the expression of the species used is plotted.

4 Discussion

5 Representation of gene regulatory networks

In [2], 243 possible GRN candidates are derived from the literature, namely the GRNs that differ in the mutual influence between *Msx2* and *Dlx5*, the influence of *TGF β 1* and *Msx2* on *Runx2*, and the influence of *Runx2* on *Dlx5* (see figure 1 for a graphical representation of the initial GRNs). The GRNs are derived from literature, where two or more studies confirming the influence of one species upon another is understood as certain influence (i.e. no GRNs are generated that indicate no or contrary influence), one study reporting an influence is seen as unclear (dotted lines in fig. 1, thus all three possible modalities are tested), and no reported finding is interpreted as no influence of one species onto another.

References

- [1] D.M. Wittmann et. al. Transforming boolean models to continuous models: methodology and application to t-cell receptor signaling. *BMC Systems Biology*, 3(98), 2009.
- [2] G.R. Kirkham et. al. Early gene regulation of osteogenesis in embryonic stem cells. *Integrative biology*, 4(12):1470–7, 2012.
- [3] J. Krumsiek et. al. From discrete to continuous gene regulation models - a tutorial using the odefy toolbox. *Applications of MATLAB in Science and Engineering*, 2011.

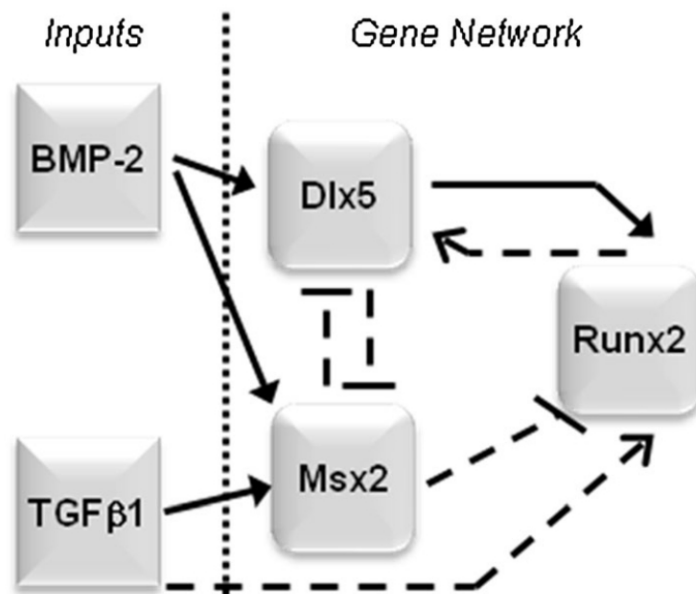


Figure 2: initial GRN as in [2]. Dotted lines indicate an unclear influence of one species onto another.