

Reverse engineering of biological models

Dennis Keil

Hamburg University of Technology

dennis.keil@tuhh.de

May 13, 2015

Contents

- 1 Biological systems and models
- 2 Computer algebra
- 3 Workflow with example
- 4 Demo
- 5 Future Work & Conclusions

Introduction

Biological systems and models

- Measurement of inner cell processes
- Collection of experimental data
- Function / Interaction can be described in biological models

Biological systems and models

- Measurement of inner cell processes
- Collection of experimental data
- Function / Interaction can be described in biological models

Biological systems and models

- Measurement of inner cell processes
- Collection of experimental data
- Function / Interaction can be described in biological models

Computer Algebra

- Symbols instead of floating points
- Polynomials and e.g. solve polynomial equations

Computer Algebra

- Symbols instead of floating points
- Polynomials and e.g. solve polynomial equations

Idea

Reverse engineer biological models using computer algebra

Workflow

- Experimental data
- Discretize data
- Interpolate particular solution
- Determine vanishing ideal
- Reduce particular solution to general solution
- General solution

Idea

Reverse engineer biological models using computer algebra

Workflow

- Experimental data
- Discretize data
- Interpolate particular solution
- Determine vanishing ideal
- Reduce particular solution to general solution
- General solution

Idea

Reverse engineer biological models using computer algebra

Workflow

- Experimental data
- Discretize data
- Interpolate particular solution
- Determine vanishing ideal
- Reduce particular solution to general solution
- General solution

Idea

Reverse engineer biological models using computer algebra

Workflow

- Experimental data
- Discretize data
- Interpolate particular solution
- Determine vanishing ideal
- Reduce particular solution to general solution
- General solution

Idea

Reverse engineer biological models using computer algebra

Workflow

- Experimental data
- Discretize data
- Interpolate particular solution
- Determine vanishing ideal
- Reduce particular solution to general solution
- General solution

Idea

Reverse engineer biological models using computer algebra

Workflow

- Experimental data
- Discretize data
- Interpolate particular solution
- Determine vanishing ideal
- Reduce particular solution to general solution
- General solution

Modeling in Systems Biology

Experimental data

| Time | g_1 | g_2 | \cdots | g_n | |
|----------|-------|-----------|-----------|----------|-----------------------|
| 1 | (| $s_{1,1}$ | $s_{1,2}$ | \cdots | $s_{1,n}$) = s_1 |
| 2 | (| $s_{2,1}$ | $s_{2,2}$ | \cdots | $s_{2,n}$) = s_2 |
| 3 | (| $s_{3,1}$ | $s_{3,2}$ | \cdots | $s_{3,n}$) = s_3 |
| \vdots | (| \vdots | \vdots | \vdots | \vdots) = \vdots |
| m | (| $s_{m,1}$ | $s_{m,2}$ | \cdots | $s_{m,n}$) = s_m |

Experimental data (example)

| Time | g_1 | g_2 | g_3 |
|------|--------|--------|--------|
| 1 | 1.6104 | 1.2042 | 1.0072 |
| 2 | 1.7073 | 1.3252 | 1.0185 |
| 3 | 1.7254 | 1.4118 | 1.0336 |
| 4 | 1.7011 | 1.4616 | 1.0508 |
| 5 | 1.6601 | 1.4814 | 1.0685 |

Discretization

- Map real numbers into a finite number p of possible states
- Often used: Boolean networks with two states (gene on/off)

Discretization

- Map real numbers into a finite number p of possible states
- Often used: Boolean networks with two states (gene on/off)

Discretization (example)

3 states $(-1, 0, 1)$

| Value | g_1 | g_2 | g_3 |
|-------|-------|-------|-------|
| -1 | - | - | - |
| 0 | 1.650 | 1.250 | 1.02 |
| 1 | 1.702 | 1.420 | 1.05 |

| Time | g_1 | g_2 | g_3 |
|------|--------|--------|--------|
| 1 | 1.6104 | 1.2042 | 1.0072 |
| 2 | 1.7073 | 1.3252 | 1.0185 |
| 3 | 1.7254 | 1.4118 | 1.0336 |
| 4 | 1.7011 | 1.4616 | 1.0508 |
| 5 | 1.6601 | 1.4814 | 1.0685 |

Discretization (example)

Data in $\mathbb{Z}/3$

| Time | g_1 | g_2 | g_3 |
|------|-------|-------|-------|
| 1 | -1 | -1 | -1 |
| 2 | 1 | 0 | -1 |
| 3 | 1 | 0 | 0 |
| 4 | 0 | 1 | 1 |
| 5 | 0 | 1 | 1 |

Interpolation

Transition function for each gene i that fits all time steps s_j

$$f_i(s_j) = s_{j+1,i}$$

Lagrange interpolation, Chinese Remainder Theorem

Interpolation (example)

| Time | g_1 | g_2 | g_3 |
|------|-----------|-----------|-----------|
| 1 | -1 | -1 | -1 |
| 2 | 1 | 0 | -1 |
| 3 | 1 | 0 | 0 |
| 4 | 0 | 1 | 1 |
| 5 | 0 | 1 | 1 |

Interpolation (example)

| Time | g_1 | g_2 | g_3 |
|------|----------|----------|-----------|
| 1 | -1 | -1 | -1 |
| 2 | 1 | 0 | -1 |
| 3 | 1 | 0 | 0 |
| 4 | 0 | 1 | 1 |
| 5 | 0 | 1 | 1 |

Interpolation (example)

| Time | g_1 | g_2 | g_3 |
|------|----------|----------|----------|
| 1 | -1 | -1 | -1 |
| 2 | 1 | 0 | -1 |
| 3 | 1 | 0 | 0 |
| 4 | 0 | 1 | 1 |
| 5 | 0 | 1 | 1 |

Interpolation (example)

| Time | g_1 | g_2 | g_3 |
|------|----------|----------|----------|
| 1 | -1 | -1 | -1 |
| 2 | 1 | 0 | -1 |
| 3 | 1 | 0 | 0 |
| 4 | 0 | 1 | 1 |
| 5 | 0 | 1 | 1 |

Interpolation (example)

Polynomials in $\mathbb{Z}/3[x_1, x_2, x_3]$

$$f_1^0 = x_1^2 x_3 - x_1^2 + x_1 x_3 + x_1$$

$$f_2^0 = -x_1^2 x_3 + x_1^2 - x_1 x_3 - x_1 + 1$$

$$f_3^0 = -x_1^2 x_3 - x_1^2 - x_1 x_3 + x_1 + 1$$

Vanishing ideal

Find all polynomials that vanish (equal zero) on the input data

$$g(s_j) = 0 \quad \text{for all} \quad s_j$$

Vanishing ideal (example)

$$I_j = \langle x_1 - s_{j,1}, \dots, x_n - s_{j,n} \rangle$$

| Time | g_1 | g_2 | g_3 |
|----------|----------|----------|-----------|
| 1 | -1 | -1 | -1 |
| 2 | 1 | 0 | -1 |
| 3 | 1 | 0 | 0 |
| 4 | 0 | 1 | 1 |
| 5 | 0 | 1 | 1 |

$$I_2 = \langle x_1 - 1, x_2, x_3 + 1 \rangle$$

Vanishing ideal

Find all polynomials that vanish (equal zero) on the input data

$$g(s_j) = 0 \quad \text{for all} \quad s_j$$

$$I_j = \langle x_1 - s_{j,1}, \dots, x_n - s_{j,n} \rangle$$

$$I = \bigcap_{j=1}^m I_j$$

Vanishing ideal (example)

$$I_1 = \langle x_1 + 1, x_2 + 1, x_3 + 1 \rangle$$

$$I_2 = \langle x_1 - 1, x_2, x_3 + 1 \rangle$$

$$I_3 = \langle x_1 - 1, x_2, x_3 \rangle$$

$$I_4 = \langle x_1, x_2 - 1, x_3 - 1 \rangle$$

$$I_5 = \langle x_1, x_2 - 1, x_3 - 1 \rangle$$

$$I = \langle x_1 + x_2 - 1, x_2 x_3 - x_3^2 + x_2 - x_3, x_2^2 - x_3^2 + x_2 - x_3 \rangle$$

Reduction

Suppose there are two Polynomials f_i, h_i that interpolate for gene i

$$f_i(s_j) = s_{j+1,i} = h_i(s_j)$$

$$f_i(s_j) - h_i(s_j) = 0 \quad \text{for all} \quad s_j$$

All possible solutions

$$f_i^0 + I = \{f_i^0 + g : g \in I\}$$

Reduction

Divide our solution f_i^0 by $I = \langle b_1, \dots, b_n \rangle$

$$f_i^0 = h_1 b_1 + \dots + h_n b_n + f_i$$

$$f_i$$

Reduction (example)

Particular solution

$$f_1^0 = x_1^2 x_3 - x_1^2 + x_1 x_3 + x_1$$

$$f_2^0 = -x_1^2 x_3 + x_1^2 - x_1 x_3 - x_1 + 1$$

$$f_3^0 = -x_1^2 x_3 - x_1^2 - x_1 x_3 + x_1 + 1$$

Reduced by vanishing ideal I

$$f_1 = -x_3^2 + x_3$$

$$f_2 = x_3^2 - x_3 + 1$$

$$f_3 = -x_3^2 + x_2 + 1$$

Demo

<http://sedk1661.github.io/reverse-engineering-of-biological-models>

Future Work & Conclusions

Future Work

- 1 Use different monomial orders for the algorithm (see gröbner fan)
- 2 Enhance implementation and parallelism of algorithms

Future Work

- ① Use different monomial orders for the algorithm (see gröbner fan)
- ② Enhance implementation and parallelism of algorithms

Conclusions

- ① Software for reverse engineering discrete biological models
- ② Study biological systems more easily

Conclusions

- ① Software for reverse engineering discrete biological models
- ② Study biological systems more easily

Thank you for listening!