The Emergence of Systems Biology

Vijay Saraswat
IBM TJ Watson Research Center and
The Pennsylvania State University

Systems Biology

Goal: To help the biologist model, simulate, analyze, design and diagnose biological systems.

- Develop system-level understanding of biological systems
  - Genomic DNA, Messenger RNA, proteins, information pathways, signaling networks
  - Intra-cellular systems, Inter-cell regulation…
  - Cells, Organs, Organisms
    - ~12 orders of magnitude in space and time!
- Key question: Function from Structure
  - How do various components of a biological system interact in order to produce complex biological functions?
  - How do you design systems with specific properties (e.g. organs from cells)?
- Share Formal Theories, Code, Models …

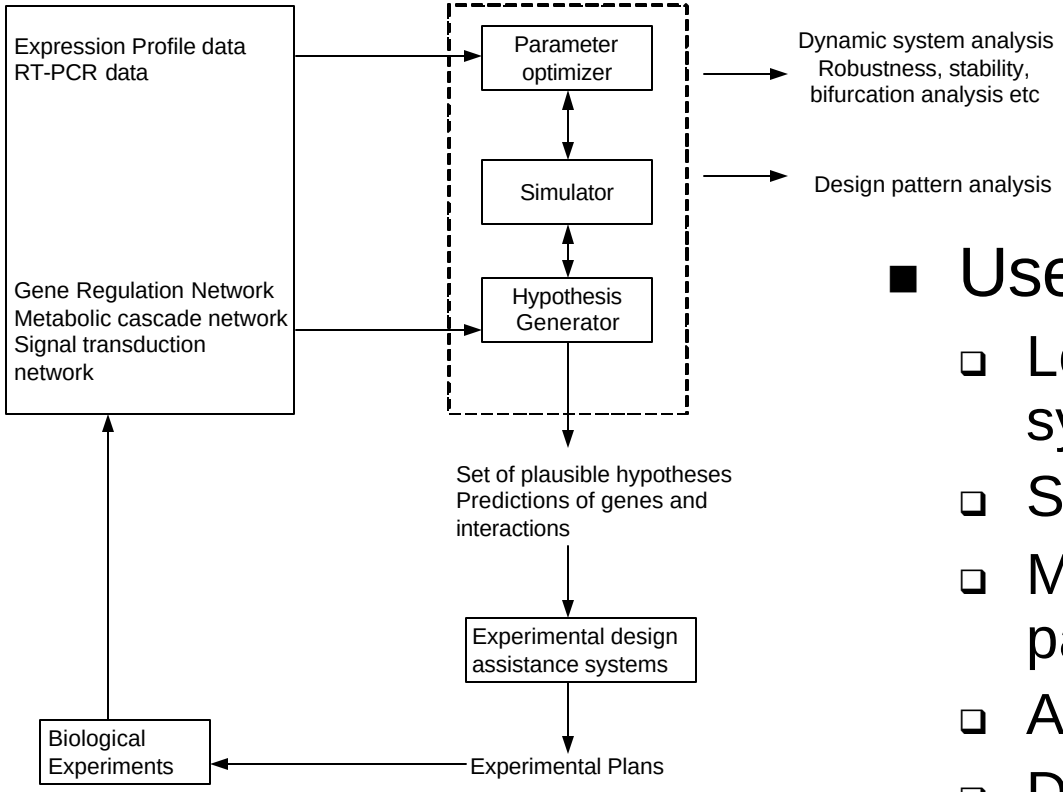
Promises profound advances in Biology and Computer Science

Systems Biology

- Work subsumes past work on mathematical modeling in biology:
  - Hodgkin-Huxley model for neural firing
  - Michaelis-Menten equation for Enzyme Kinetics
  - Gillespie algorithm for Monte-Carlo simulation of stochastic systems.
  - Bifurcation analysis for Xenopus cell cycle
  - Flux balance analysis, metabolic control analysis…
- Why Now?
  - Exploiting genomic data
  - Scale
    - Across the internet, across space and time.
  - Integration of computational tools
  - Integration of new analysis techniques
  - Collaboration using markup-based interlingua
  - Moore's Law!

This is not the first time…

Integrating Computation into experimentation



An integrated approach to biological experimentation
(From Kitano [Sys-Bio])

- Use *all* of Comp Sci
  - Logic and Hybrid systems
  - Symbolic Analysis Tools
  - Machine learning and pattern recognition
  - Algorithms
  - Databases
  - Modeling languages

Area is Exploding in interest…

- Conferences…
  - BioConcur ‘03
  - Pacific Sym BioComputing ‘04
  - International Workshop on Systems Biology
  - Comp Methods in Sys Bio, 2004
  - Systeomatics 2004
- Websites
  - www.sbml.org
  - www.cellml.org
  - www.systemsbiology.org
- Projects
  - BioSpice (DARPA)
  - CellML (U Auckland)
  - SBML
    - CalTech, U Hertfordshire, Argonne, Virginia, U Conn…
  - Post-genomic institutes
    - Harvard/MIT, Princeton
- Systems
  - BioSpice, Charon, Cellerator, COPASI, DBSolve, E-Cell, Gepasi, Jarnac, JDesigner, JigCell, NetBuilder, StochSim, Virtual Cell…

Hybrid Systems

- Traditional Computer Science
  - Discrete state, discrete change (assignment)
  - E.g. Turing Machine
  - Brittleness:
    - Small error → major impact
    - Devastating with large code!
- Traditional Mathematics
  - Continuous variables (Reals)
  - Smooth state change
    - Mean-value theorem
    - E.g. computing rocket trajectories
  - Robustness in the face of change
  - Stochastic systems (e.g. Brownian motion)

- Hybrid Systems combine both
  - Discrete control
  - Continuous state evolution
  - Intuition: Run program at every real value.
    - Approximate by:
      - Discrete change at an instant
      - Continuous change in an interval
- Primary application areas
  - Engineering and Control systems
    - Paper transport
    - Autonomous vehicles…
  - And now.. *Biological Computation*.

Emerged in early 90s in the work of Nerode, Kohn, Alur, Dill, Henzinger…

Hcc: Hybrid Concurrent Constraint Progg.

Very flexible programming and modeling language

Based on a general theory of concurrency and constraints

- Has a built-in notion of continuous time
- Supports smooth *and* discontinuous system evolution
- Supports stochastic modeling
- Provides powerful, extensible constraint solver
- Can handle variable-structure systems
- Supports qualitative and quantitative modeling.
- Built on a formal operational and denotational semantics
- Supports meta-programming (dynamic generation of programs)
- Completely integrated with Java

Saraswat, Jagadeesan, Gupta "jcc: Integrating TDCC into Java"

Hcc: A language for hybrid modeling

- Hcc is based on a very few primitives
  - **`c`**
    - Establish constraint c now
  - **`if(c){S}`**
    - Run S when c holds (at this instant)
  - **`unless(c){S}`**
    - Run S unless c holds (at this instant)
  - **`S,S`**
    - Run the two in parallel
  - **`hence S`**
    - Run S at every real after now
- Language can be used to express any pattern of evolution across time:
  - **`always{S}`**
    - run S at every time point
  - **`every(c){S}`**
    - run S at every time point at which c holds.
  - **`watching(c){S}`**
    - run S, aborting it as soon as c holds.

Gupta, Jagadeesan, Saraswat "Computing with continuous change", SCP 1998

Hcc for Systems Biology

Systems Biology	*jcc*
Reaching Threshold	Discrete change
Time, species conc	Continuous variables
Kinetics	Differential equations
Gene interaction	Concurrency, defaults
Stochastic behavior	Stochastic variables

Bockmayr, Courtois: "Using hccp to model dynamic biological systems", ICLP 02

Basic example

- Expression of gene x inhibits expression of gene y; above a certain threshold, gene y inhibits expression of gene x:

2geneinteraction.ps

if (y < 0.8) {x'= -0.02*x + 0.01},

If (y >= 0.8) {x'=-0.02*x, y'=0.01*x}

Bioluminescence in E Fischeri

- Bioluminescence in *V. fischeri*
  - When density passes a certain threshold, (marine) bacteria suddenly become luminescent
- Model:
  - Variables x7,x9 represents internal (ext) concentration of Ai.
  - Variables x1,..x6,x8 represent other species
- Use generic balance eqn:
  - x'=vs - vd +/- vr +/- vt
    - vs: synthesis rate
    - vd: degradation rate
    - vr: reaction rate
    - vt: transportation rate
  - E.g.

```
always{
if (x7 <Ai_min) x1'=mu1*((0.5*x)-x1),
If (x7 >= Ai_plus) x1'=-mu1*x1,…
}
```

The conditional ODEs governing 9 system variables can be directly transcribed into jcc.

Delta-Notch signaling in X. Laevis

- Consider cell differentiation in a population of epidermic cells.
- Cells arranged in a hexagonal lattice.
- Each cell interacts concurrently with its neighbors.
- The concentration of Delta and Notch proteins in each cell varies continuously.
- Cell can be in one of four states: Delta and Notch inhibited or expressed.

- Experimental Observations:
  - Delta (Notch) concentrations show typical spike at a threshold level.
  - At equilibrium, cells are in only two states (D or N expressed; other inhibited).

Ghosh, Tomlin: “Lateral inhibition through Delta-Notch signaling: A piece-wise affine hybrid model”, HSCC 2001

Delta-Notch Signaling

- Model:
  - VD, VN: concentration of Delta and Notch protein in the cell.
  - UD, UN: Delta (Notch) production capacity of cell.
  - UN=sum_i VD_i (neighbors)
  - UD = -VN
  - Parameters:
    - Threshold values: HD,HN
    - Degradation rates: MD, MN
    - Production rates: RD, RN

- Model:
  - Cell in 1 of 4 states: {D,N} x {Expressed (above), Inhibited (below)}

if (UN(i,j) < HN) {VN'= -MN*VN},

if (UN(I,j)>=HN){VN'=RN-MN*VN},

if (UD(I,j)<HD){VD'=-MD*VD},

if (UD(I,j)>=HD){VD'=RD-MD*VD},

- Stochastic variables used to set random initial state.
- Model can be expressed directly in hcc.

Results: Simulation confirms observations. Tiwari/Lincoln prove that States 2 and 3 are stable.

Alternative splicing regulation

- Alternative splicing occurs in post transcriptional regulation of RNA
- Through selective elimination of introns, the same premessenger RNA can be used to generate many kinds of mature RNA
- The SR protein appears to control this process through activation and inhibition.
- Because of complexity, experimentation can focus on only one site at a time.
- Bockmayr et al use Hybrid CCP to model SR regulation at a single site.
  - Michaelis-Menten model using 7 kinetic reactions
- This is used to create an n-site model by abstracting the action at one site via a splice efficiency function.

Results described in [Alt], uses default reasoning properties of HCC.

Programming Languages Issues

- Languages for large-scale modeling
  - Hi-perf num computation
  - Arrays
  - Stochastic methods
  - Large-scale parallelism (e.g SPMD)
- Efficient compilation issues
  - Identify patterns, integrate libraries of high-performance code
- Integration of reasoning techniques
  - Eg finite state analysis of hybrid systems
- Syntax/Semantics
- Integration of Spatial dimension
  - Moving to PDEs
- Developing models across the Internet
  - Semantic web…

Exciting time for the development of new languages!

Acknowledgements

- [Sys-Bio]: Kitano “Systems Biology: Towards system-level understanding of Biological Systems”, in *Foundations of Systems Biology*, MIT Press, 2001
- [Delta-Notch]: Tiwari, Lincoln “Automatic Techniques for stability analysis of Delta-Notch lateral inhibition mechanism”, *CSB* 2002.
- [HCC-Bio]: Bockmayr, Courtois “Using hybrid concurrent constraint programming to model dynamic biological systems”, *ICLP* 2002
- [Alt]: Eveillard, Ropers, de Jong, Branlant, Bockmayr “A multi-site constraint programming model of alternate splicing regulation”, *INRIA Tech Rep*, May 2003

HCC references

- Gupta, Jagadeesan, Saraswat “Computing with Continuous Change”, Science of Computer Programming, Jan 1998, 30 (1—2), pp 3--49
- Saraswat, Jagadeesan, Gupta “Timed Default Concurrent Constraint Programming”, Journal of Symbolic Computation, Nov-Dec1996, 22 (5—6), pp 475-520.
- Gupta, Jagadeesan, Saraswat “Programming in Hybrid Constraint Languages”, Nov 1995, Hybrid Systems II, LNCS 999.
- Alenius, Gupta “Modeling an AERCam: A case study in modeling with concurrent constraint languages”, CP’98 Workshop on Modeling and Constraints, Oct 1998.

CFP: Wkshp Comp Methods in Sys Bio

Deadline : March 1, 2004

Call for Papers - International Workshop on Computational Methods in Systems Biology 2004 (CMSB'04)

Organized by Genoscope, Evry – Génopole, Evry – CNRS – University of Paris VII – BioPathways Consortium

Hotel Meridien Montparnasse, Paris, France 26-28 May, 2004

Deadline : March 1st, 2004

http://www.genoscope.cns.fr/biopathways/CMSB04/