10-14-2015

Research-driven facilitation of systems thinking with computational models in life sciences education

Heather E. Bergan-Roller
University of Nebraska-Lincoln, heather.bergan@unl.edu

Nicholas Galt
University of Nebraska-Lincoln, ngalt2@unl.edu

Tomáš Helikar
University of Nebraska-Lincoln, thelikar2@unl.edu

Joseph T. Dauer
University of Nebraska-Lincoln, joseph.dauer@unl.edu

Abstract for DBER Group Discussion on 2015-10-15

Authors and Affiliations:
Heather E. Bergan-Roller[1], Nicholas Galt[2], Tomáš Helikar[2], Joseph T. Dauer[1]
[1]School of Natural Resources, University of Nebraska-Lincoln
[2]Department of Biochemistry, University of Nebraska-Lincoln

Title
Research-driven facilitation of systems thinking with computational models in life sciences education

Abstract
Systems thinking, computational modeling, and simulating systems are examples of important skills stressed in life sciences education by Vision and Change. In response to these calls, we have designed a computational modeling and simulation-driven intervention to supplement current instruction in the life sciences curriculum. As part of our pre-intervention assessment we evaluated students on their systems thinking in the context of cellular respiration. For this assessment, we had students create conceptual models. We found that students with lecture instruction are able to recall more components associated with the cellular respiration process but are not better able to integrate these components into the system compared to students without lecture instruction. As a result, we have designed computational interventions to facilitate learning about complex biological processes. In these activities, we have students make and test predictions and apply simulation results to cellular mechanisms. We then assess student thinking to examine if the computational intervention improves systems thinking and modeling skills. Our preliminary data suggest that this intervention increases students' mechanistic reasoning abilities. Currently, we are deploying computational activities and assessing students thinking on the topics of cellular respiration and gene regulation in all LIFE 120 laboratories. Finally, we are in the process of developing new computational activities to be used as learning tools for additional topics on complex biological systems.

Research-driven facilitation of systems thinking with computational models in life sciences education

Heather E. Bergan-Roller[1], Nicholas J. Galt[2],

[1]School of Natural Resources, University of Nebraska-Lincoln
[2]Department of Biochemistry, University of Nebraska-Lincoln

Slide 1

1 delete slide
Heather Bergan,

3 whatever terminology we use, we need to be consistent and are jargon-less as possible so the audience can easily follow
Heather Bergan,

4 Do you want me to talk here?
Heather Bergan,

5 are we cutting this slide? if so, can you move it to the end section?
Heather Bergan,

6 what does this mean? I don't think we need to spend a lot of time talking about the new versus old user interface
Heather Bergan,

7 I vote we skip this and jump right into our CMI. I think I have set it up well for this.
Heather Bergan,

8 maybe move this to the beginning of your section (around slide 13) because my "baseline" results are in the context of cellular respiration, and then we switch to Gene regulation abruptly.
Heather Bergan,

Thank you!

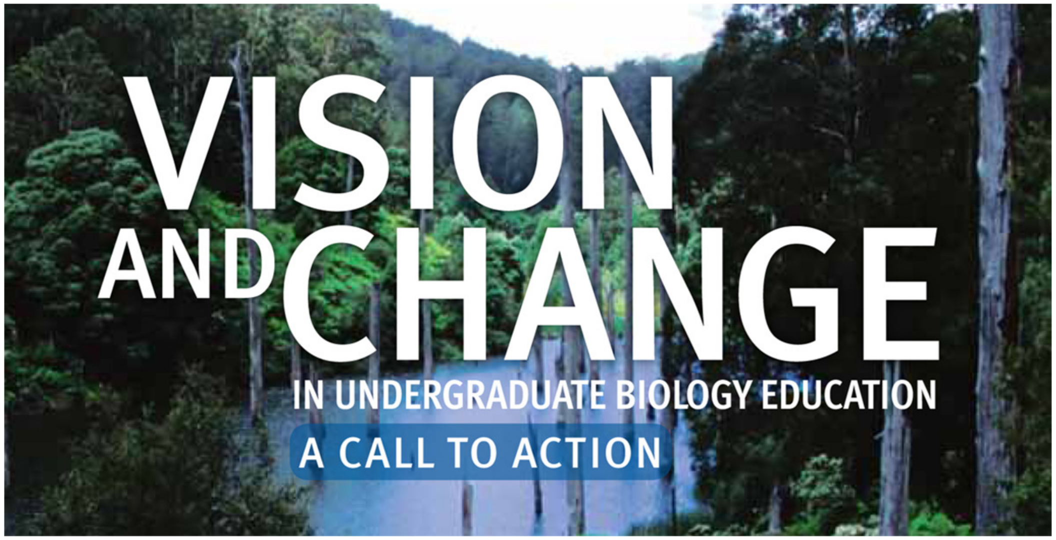
STEM education seminar organizers
LIFE instructors, lab coordinators, TAs
Steve Harris

Helikar Group
Tomáš Helikar
David Tichy
Bryan Kowal
Bhargav Gorthi
Audrey Crowther

Dauer Group
Joe Dauer
Sinan Akkoseoglu
Jai Kumar Mediratta
McKenzie Kjose
Jacob Winters

National Science Foundation IUSE #1432001

Our world is complex



Core Concepts

- **Systems**
- Structure and function
- Information flow, exchange, storage
- Evolution
- Pathways and transformation of energy and matter

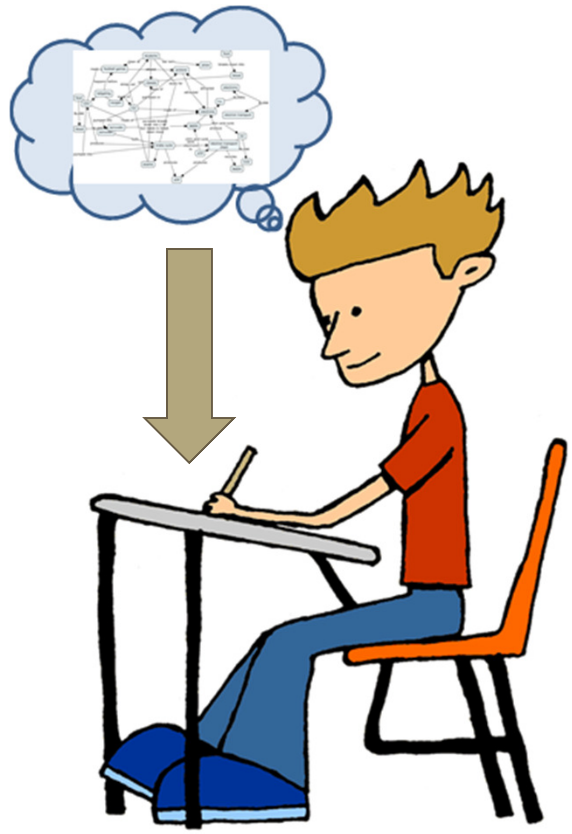
Core Competencies

- **Modeling, simulations, computational, and systems-level approaches to discovery and analysis**
- Process of science
- Quantitative reasoning
- Interdisciplinary communication and collaboration
- Science and Society

AAAS, 2011

Modeling and Learning

1. Externalize mental models
2. Decrease cognitive load
3. Address explicit interactions/mechanisms
4. Facilitate metacognition
5. Facilitate instructor feedback
6. Facilitate assessment of thinking



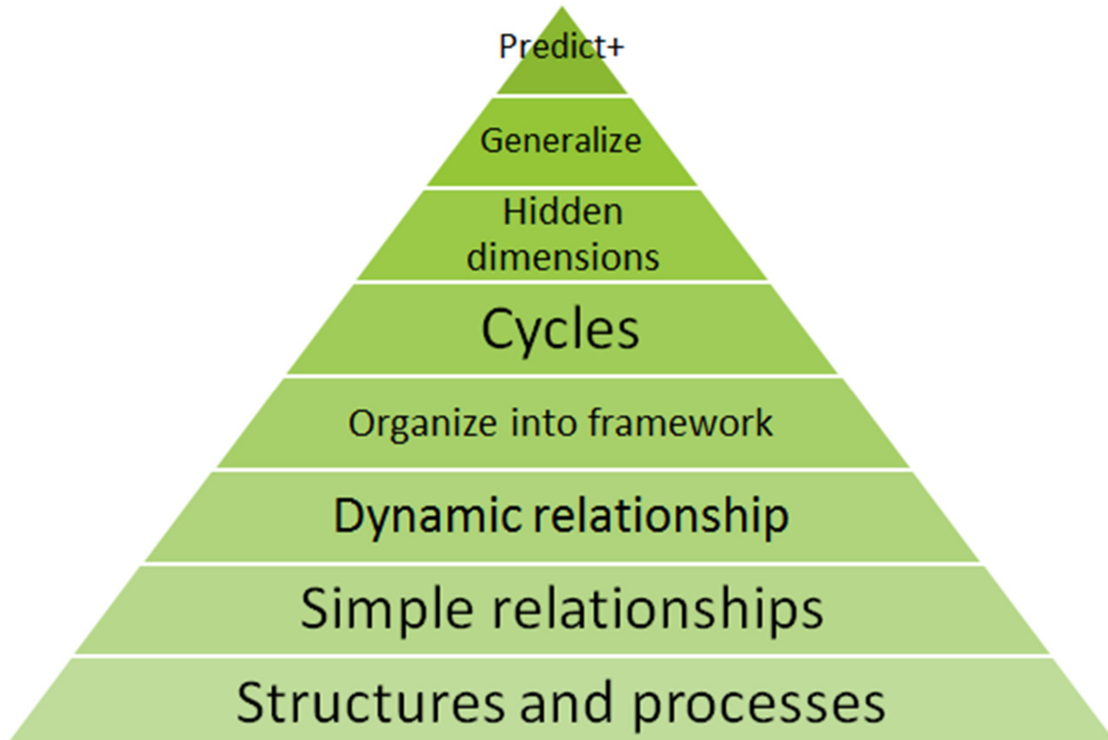
Computational model learning activities for improved systems thinking

Outline

1. **Systems thinking in our students (baseline)**
2. **Computational model learning activities**
3. **Feedback from you!**

What is the state of systems thinking in our introductory biology students?

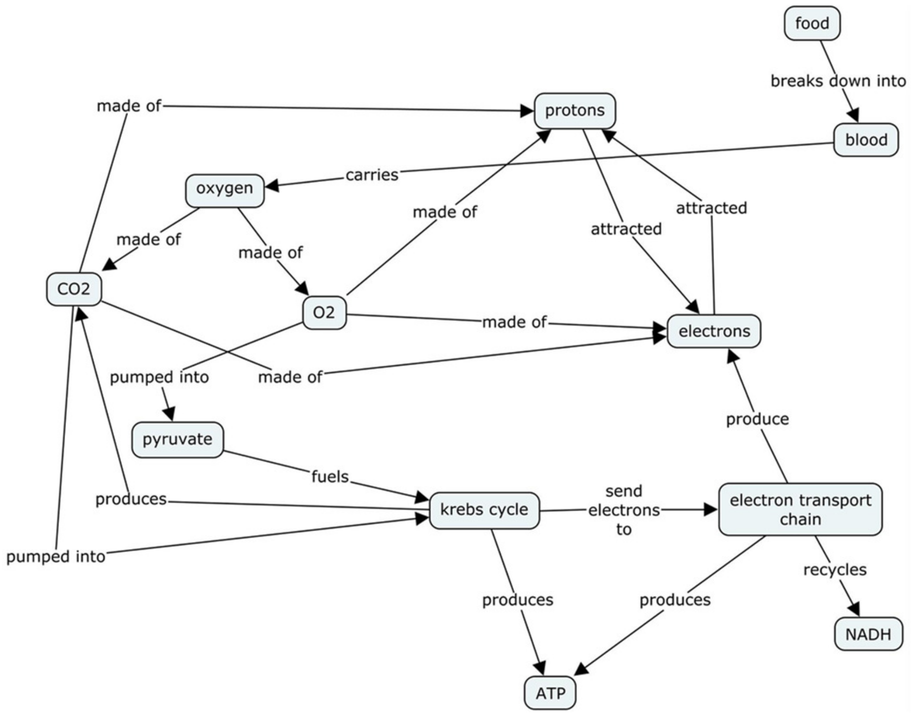
Systems thinking hierarchy framework



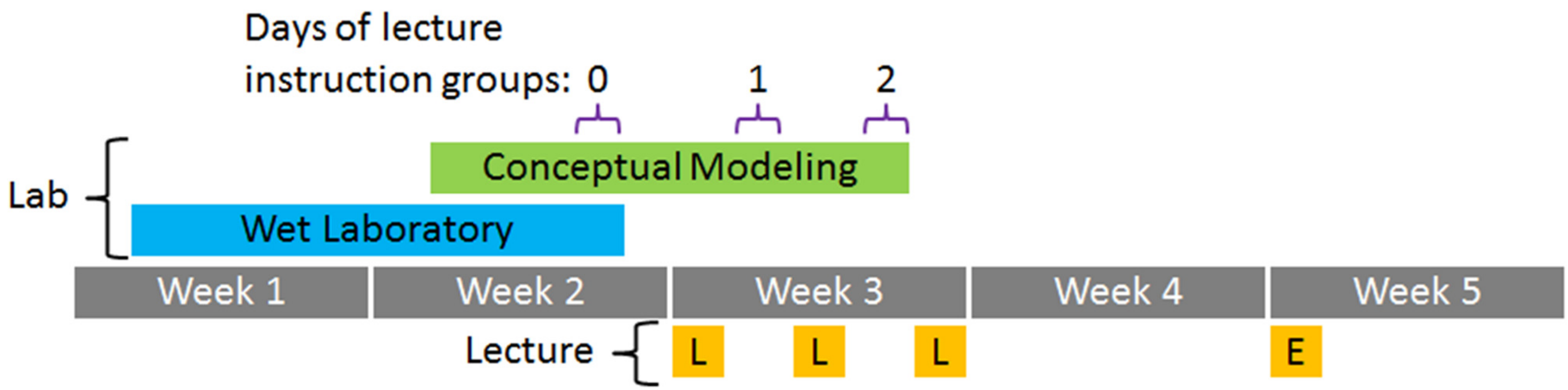
Assaraf and Orion, 2010

Conceptual models to assess thinking

(Jordan et al., 2013; Dauer et al., 2013; Vattam et al., 2011; Ifenthaler, 2010; Hmelo, Holton, Kolodner, 2000)

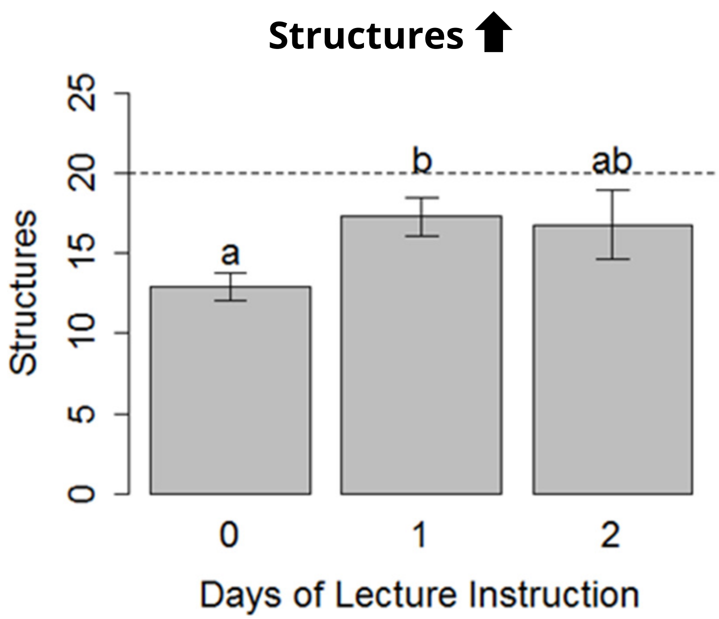


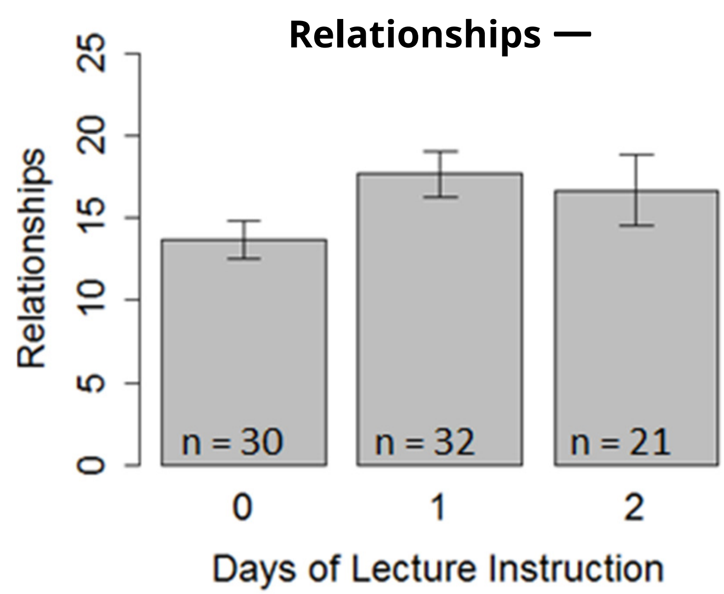
Baseline Timeline



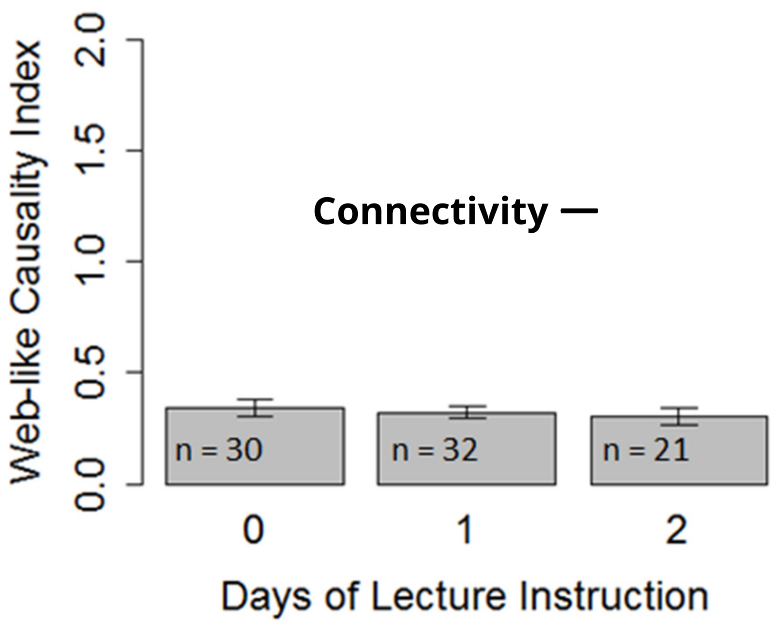
	n = 83
0	n = 30
1	n = 32
2	n = 21

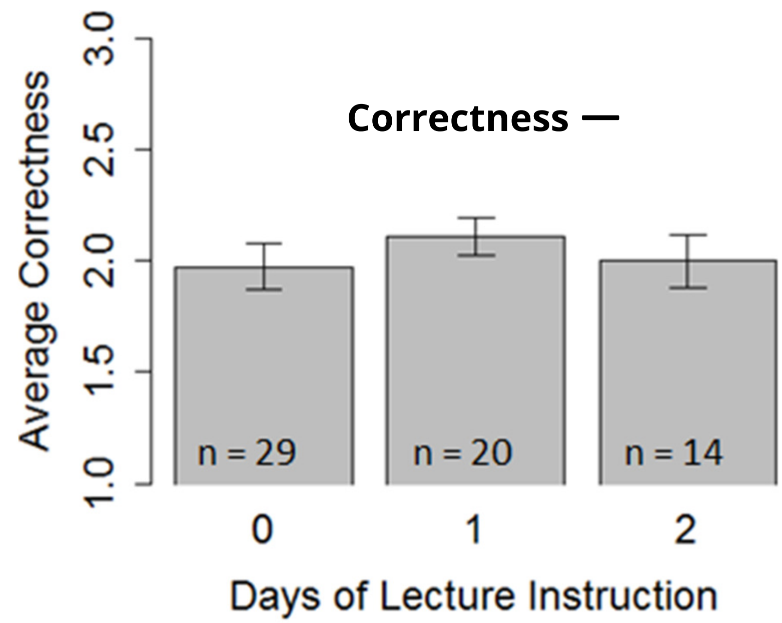
Baseline Results





Baseline Results

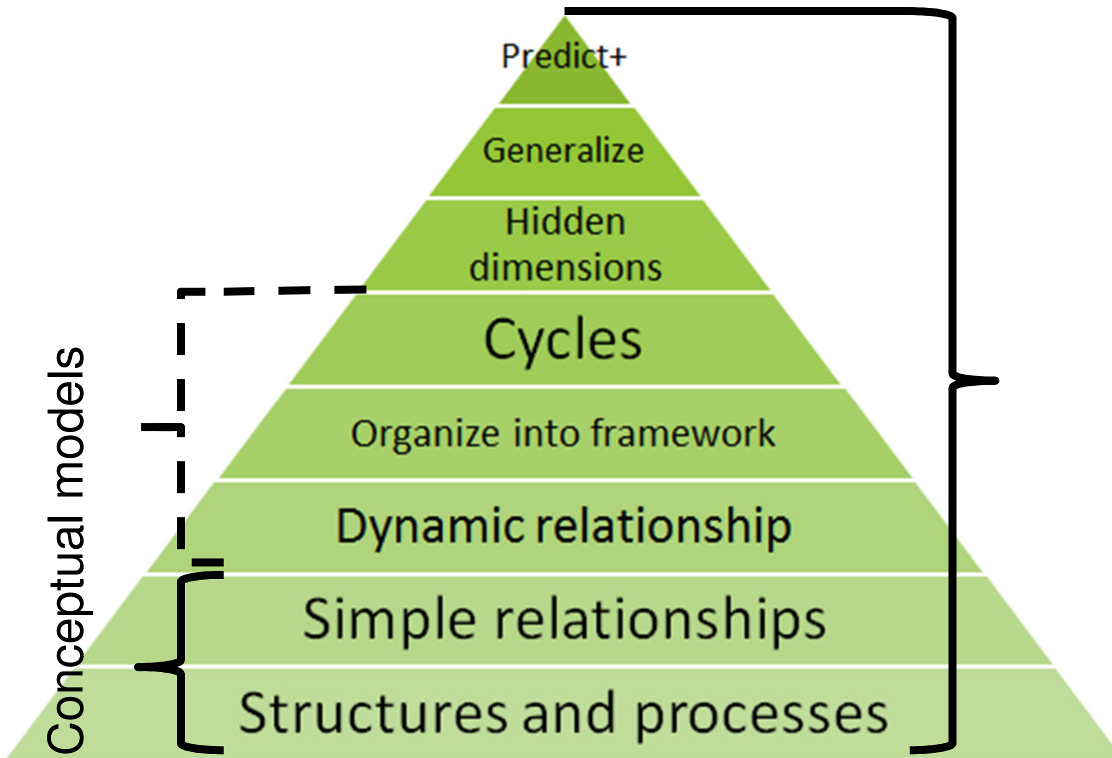




Baseline Conclusions

Instruction increases the number of structures

Instruction does not affect relationships, connectivity, or correctness



Assaraf and Orion, 2010

Computational Model Learning Activities

Goals

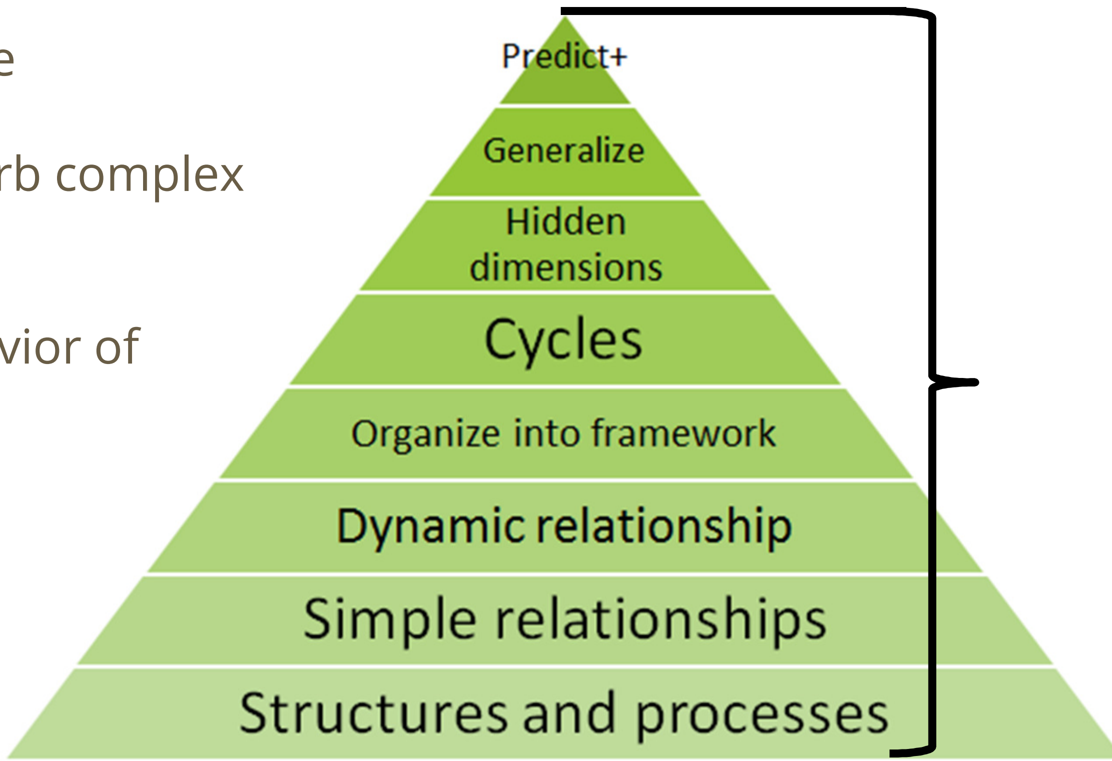
- Assess spectrum of systems thinking skills
- Provide systems thinking practice
- Improve systems thinking
- Address student misconceptions

Why computational modeling?

- Manage content knowledge
- Create, simulate and perturb complex biological systems
- Observe the dynamic behavior of systems

Why computational modeling?

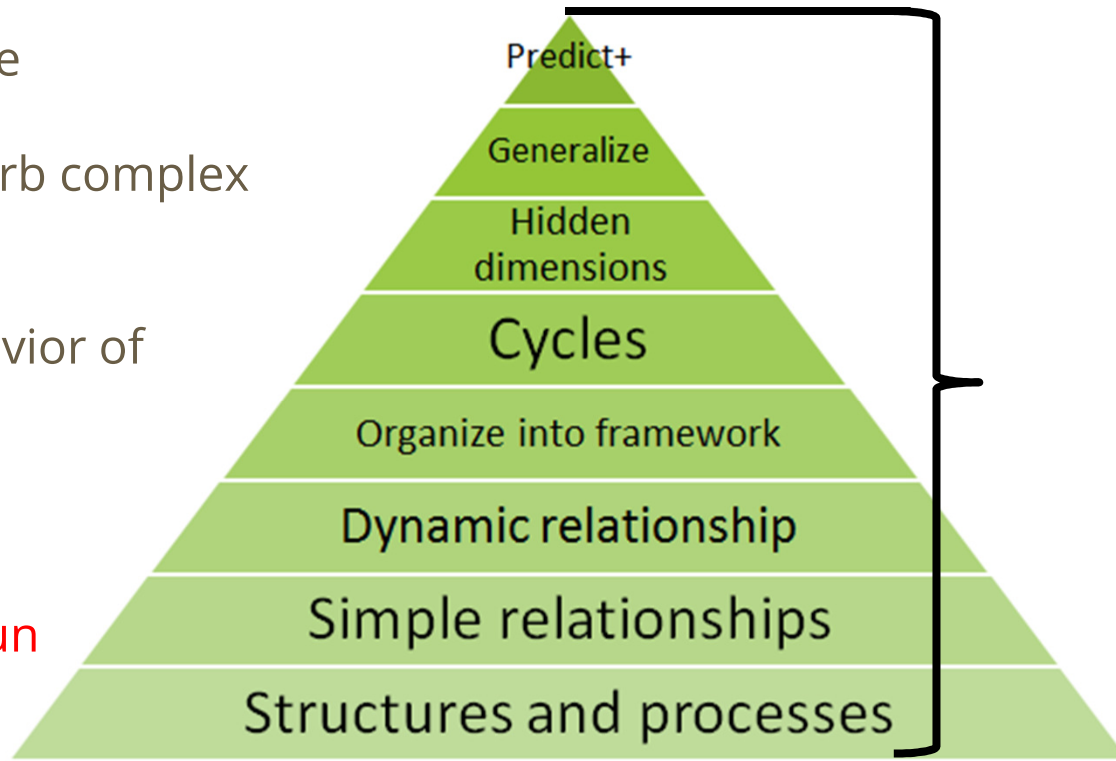
- Manage content knowledge
- Create, simulate and perturb complex biological systems
- Observe the dynamic behavior of systems
- Promote systems thinking



Assaraf and Orion, 2010

Why computational modeling?

- Manage content knowledge
- Create, simulate and perturb complex biological systems
- Observe the dynamic behavior of systems
- Promote systems thinking
- Less memorizing = more fun

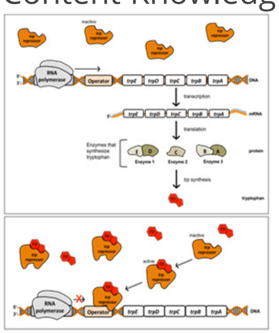


Assaraf and Orion, 2010

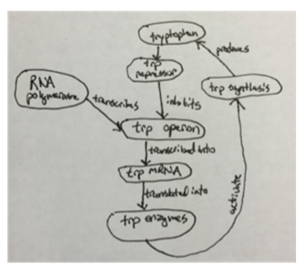
Computational Modeling Platform: Cell Collective

- Web-based (thecellcollective.org)
- Accessible and easy to use
- No entering/modifying mathematical expressions or computer code

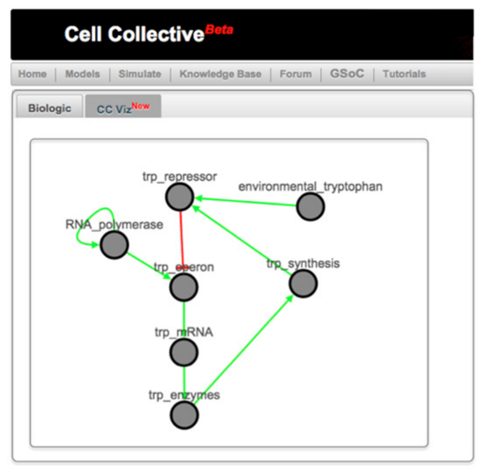
Content Knowledge



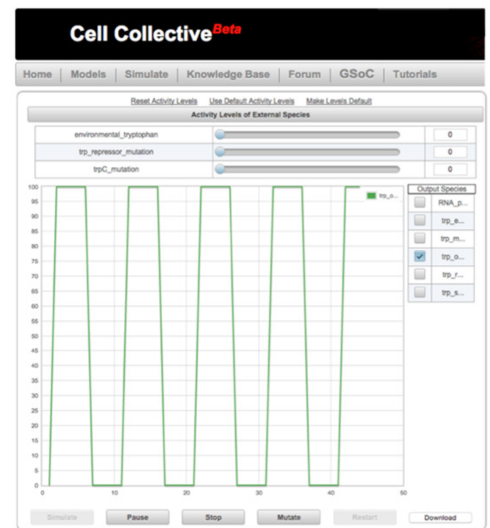
Conceptual Model



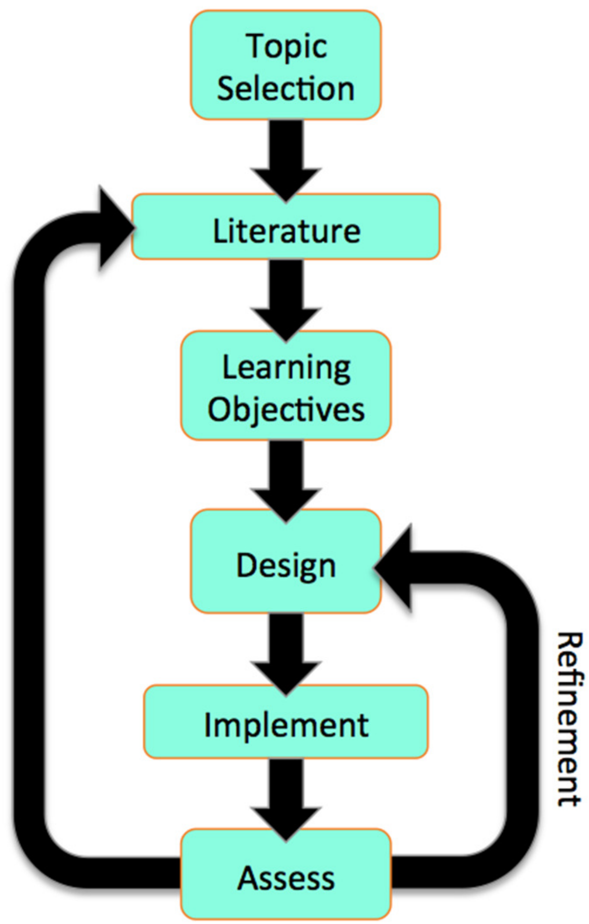
Computational Model



Dynamic Simulation



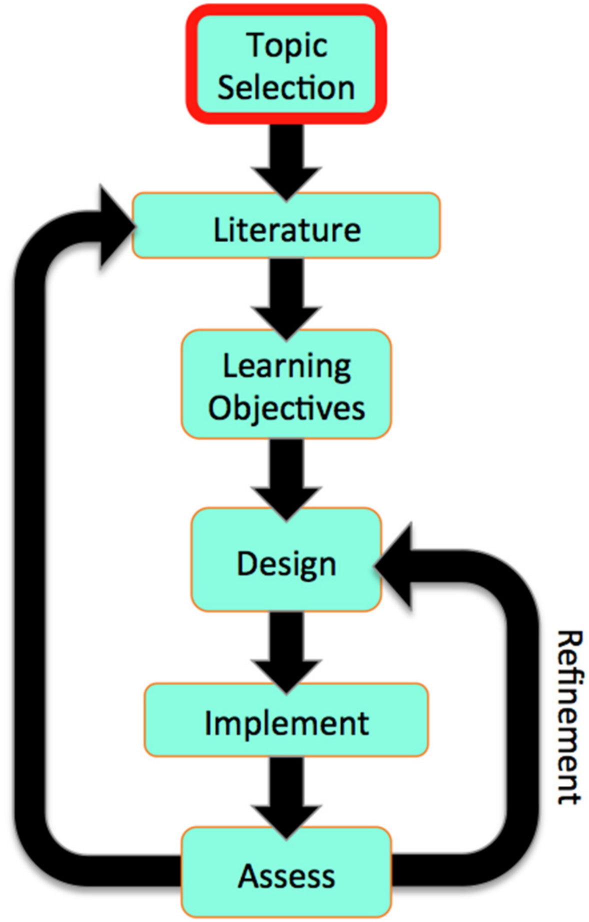
Computational Model Learning Activities



Computational Model Learning Activities

Topic Selection

- What is needed?



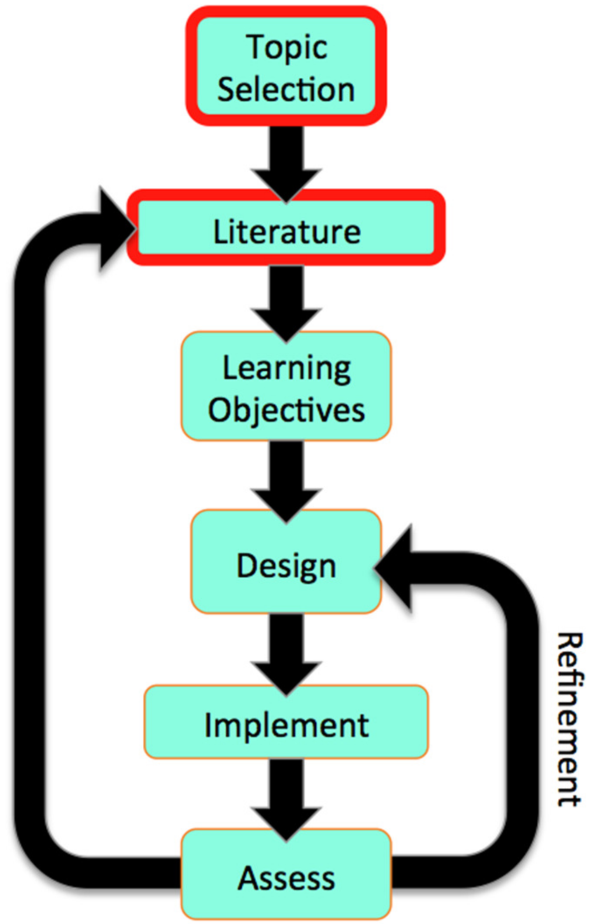
Computational Model Learning Activities

Topic Selection

- What is needed?

Literature

- What does the literature recommend?
- What do experts value?



Computational Model Learning Activities

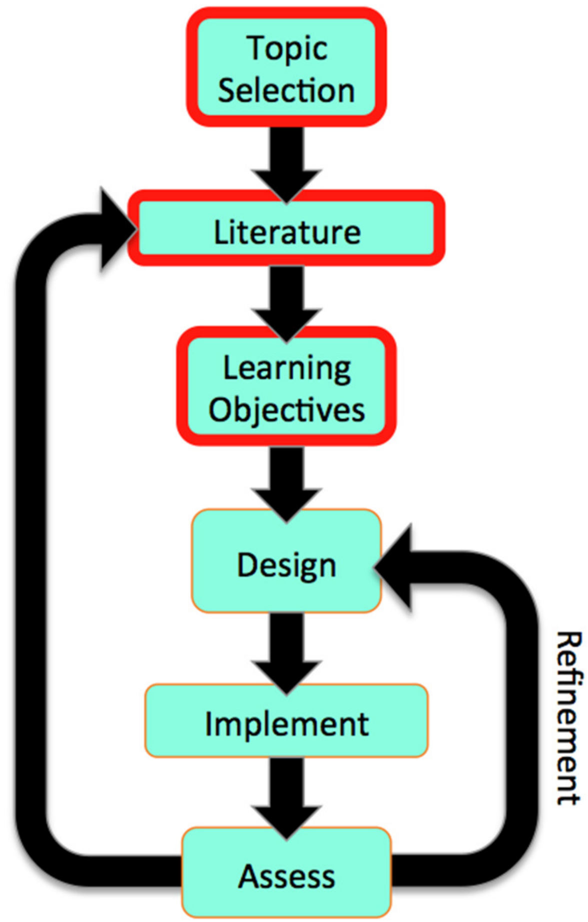
Topic Selection

- What is needed?

Literature

- What does the literature recommend?
- What do experts value?

Learning Objectives



Computational Model Learning Activities

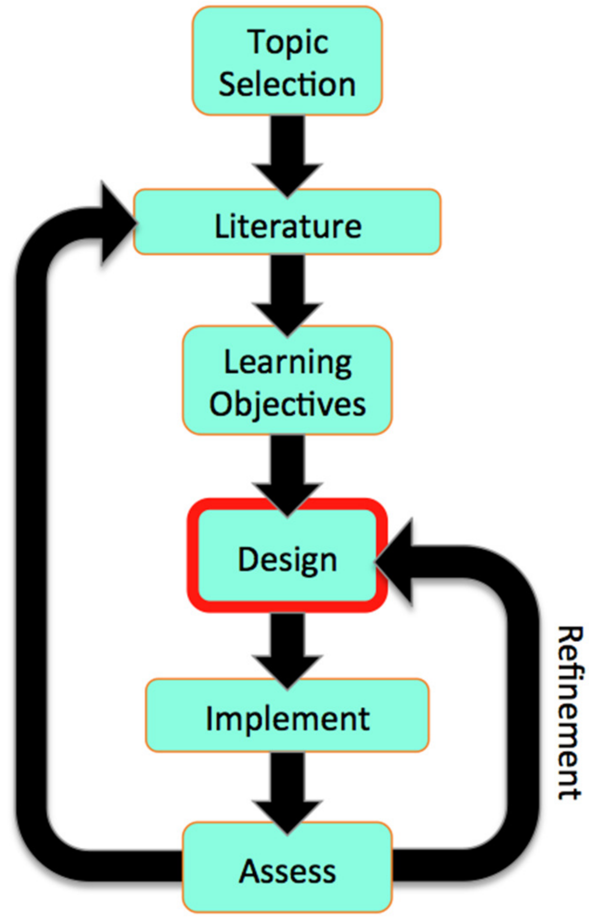
Design

a. Background Information

b. Introduction to Computational Model

c. Simulation Setup

d. Investigations



Computational Model Learning Activities

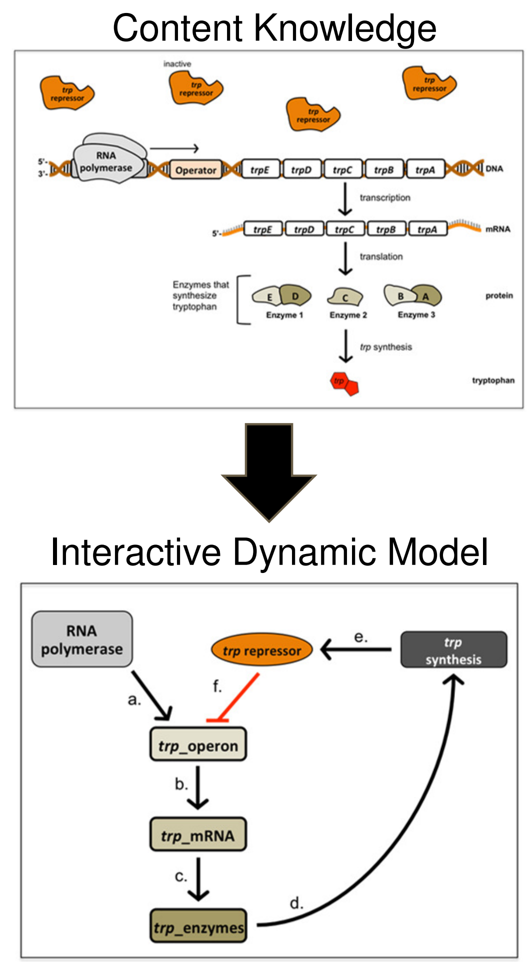
Design

a. Background Information

b. Introduction to Computational Model

c. Simulation Setup

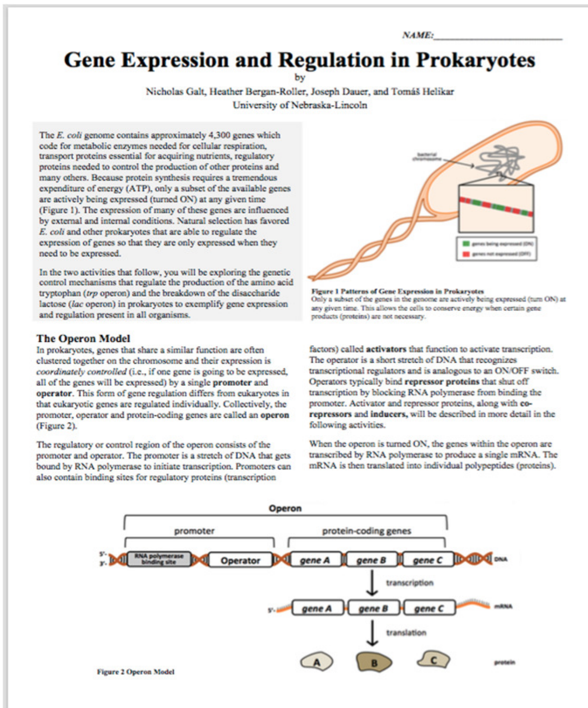
d. Investigations



Example using the tryptophan Operon

Computational Model Learning Activities Design

Part 1: Background Information



*NAME:*______________________

Gene Expression and Regulation in Prokaryotes

by

Nicholas Galt, Heather Bergan-Roller, Joseph Dauer, and Tomáš Helikar
University of Nebraska-Lincoln

The *E. coli* genome contains approximately 4,300 genes which code for metabolic enzymes needed for cellular respiration, transport proteins essential for acquiring nutrients, regulatory proteins needed to control the production of other proteins and many others. Because protein synthesis requires a tremendous expenditure of energy (ATP), only a subset of the available genes are actively being expressed (turned ON) at any given time (Figure 1). The expression of many of these genes are influenced by external and internal conditions. Natural selection has favored *E. coli* and other prokaryotes that are able to regulate the expression of genes so that they are only expressed when they need to be expressed.

In the two activities that follow, you will be exploring the genetic control mechanisms that regulate the production of the amino acid tryptophan (*trp* operon) and the breakdown of the disaccharide lactose (*lac* operon) in prokaryotes to exemplify gene expression and regulation present in all organisms.

Figure 1 Patterns of Gene Expression in Prokaryotes
Only a subset of the genes in the genome are actively being expressed (turn ON) at any given time. This allows the cells to conserve energy when certain gene products (proteins) are not necessary.

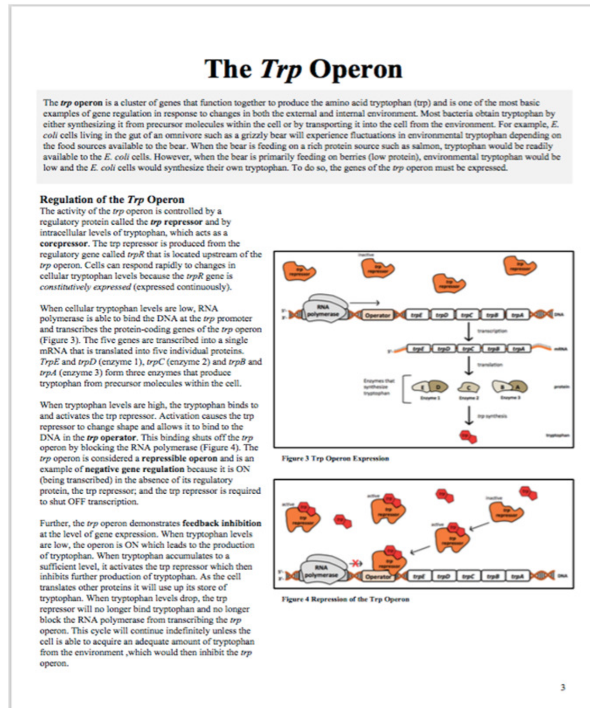
The Operon Model

In prokaryotes, genes that share a similar function are often clustered together on the chromosome and their expression is *coordinately controlled* (i.e., if one gene is going to be expressed, all of the genes will be expressed) by a single **promoter** and **operator**. This form of gene regulation differs from eukaryotes in that eukaryotic genes are regulated individually. Collectively, the promoter, operator and protein-coding genes are called an **operon** (Figure 2).

The regulatory or control region of the operon consists of the promoter and operator. The promoter is a stretch of DNA that gets bound by RNA polymerase to initiate transcription. Promoters can also contain binding sites for regulatory proteins (transcription factors) called **activators** that function to activate transcription. The operator is a short stretch of DNA that recognizes transcriptional regulators and is analogous to an ON/OFF switch. Operators typically bind **repressor proteins** that shut off transcription by blocking RNA polymerase from binding the promoter. Activator and repressor proteins, along with **co-repressors** and **inducers,** will be described in more detail in the following activities.

When the operon is turned ON, the genes within the operon are transcribed by RNA polymerase to produce a single mRNA. The mRNA is then translated into individual polypeptides (proteins).

Figure 2 Operon Model



The *Trp* Operon

The ***trp* operon** is a cluster of genes that function together to produce the amino acid tryptophan (trp) and is one of the most basic examples of gene regulation in response to changes in both the external and internal environment. Most bacteria obtain tryptophan by either synthesizing it from precursor molecules within the cell or by transporting it into the cell from the environment. For example, *E. coli* cells living in the gut of an omnivore such as a grizzly bear will experience fluctuations in environmental tryptophan depending on the food sources available to the bear. When the bear is feeding on a rich protein source such as salmon, tryptophan would be readily available to the *E. coli* cells. However, when the bear is primarily feeding on berries (low protein), environmental tryptophan would be low and the *E. coli* cells would synthesize their own tryptophan. To do so, the genes of the *trp* operon must be expressed.

Regulation of the *Trp* Operon

The activity of the *trp* operon is controlled by a regulatory protein called the ***trp* repressor** and by intracellular levels of tryptophan, which acts as a **corepressor**. The trp repressor is produced from the regulatory gene called *trpR* that is located upstream of the *trp* operon. Cells can respond rapidly to changes in cellular tryptophan levels because the *trpR* gene is *constitutively expressed* (expressed continuously).

When cellular tryptophan levels are low, RNA polymerase is able to bind the DNA at the *trp* promoter and transcribes the protein-coding genes of the *trp* operon (Figure 3). The five genes are transcribed into a single mRNA that is translated into five individual proteins. *TrpE* and *trpD* (enzyme 1), *trpC* (enzyme 2) and *trpB* and *trpA* (enzyme 3) form three enzymes that produce tryptophan from precursor molecules within the cell.

When tryptophan levels are high, the tryptophan binds to and activates the trp repressor. Activation causes the trp repressor to change shape and allows it to bind to the DNA in the ***trp* operator**. This binding shuts off the *trp* operon by blocking the RNA polymerase (Figure 4). The *trp* operon is considered a **repressible operon** and is an example of **negative gene regulation** because it is ON (being transcribed) in the absence of its regulatory protein, the trp repressor; and the trp repressor is required to shut OFF transcription.

Further, the *trp* operon demonstrates **feedback inhibition** at the level of gene expression. When tryptophan levels are low, the operon is ON which leads to the production of tryptophan. When tryptophan accumulates to a sufficient level, it activates the trp repressor which then inhibits further production of tryptophan. As the cell translates other proteins it will use up its store of tryptophan. When tryptophan levels drop, the trp repressor will no longer bind tryptophan and no longer block the RNA polymerase from transcribing the *trp* operon. This cycle will continue indefinitely unless the cell is able to acquire an adequate amount of tryptophan from the environment ,which would then inhibit the *trp* operon.

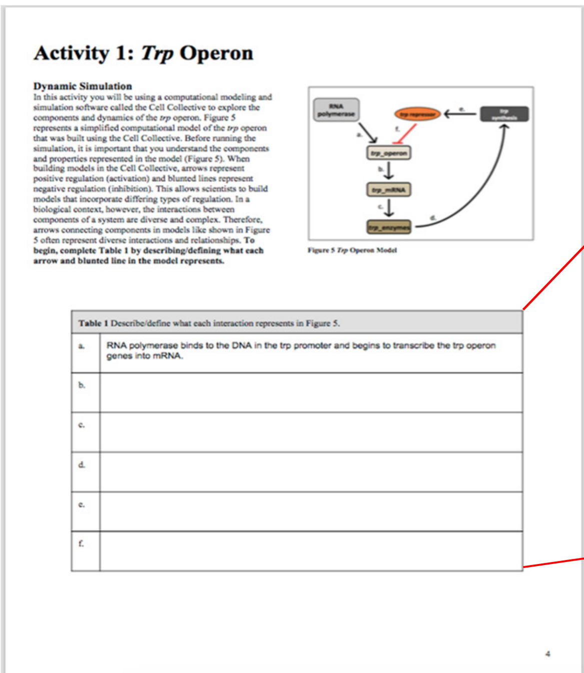
Figure 3 Trp Operon Expression

Figure 4 Repression of the Trp Operon

3

Computational Model Learning Activities Design

Part 2: Introduction to Computational Model



Activity 1: *Trp* Operon

Dynamic Simulation

In this activity you will be using a computational modeling and simulation software called the Cell Collective to explore the components and dynamics of the *trp* operon. Figure 5 represents a simplified computational model of the *trp* operon that was built using the Cell Collective. Before running the simulation, it is important that you understand the components and properties represented in the model (Figure 5). When building models in the Cell Collective, arrows represent positive regulation (activation) and blunted lines represent negative regulation (inhibition). This allows scientists to build models that incorporate differing types of regulation. In a biological context, however, the interactions between components of a system are diverse and complex. Therefore, arrows connecting components in models like shown in Figure 5 often represent diverse interactions and relationships. **To begin, complete Table 1 by describing/defining what each arrow and blunted line in the model represents.**

Figure 5 *Trp* Operon Model

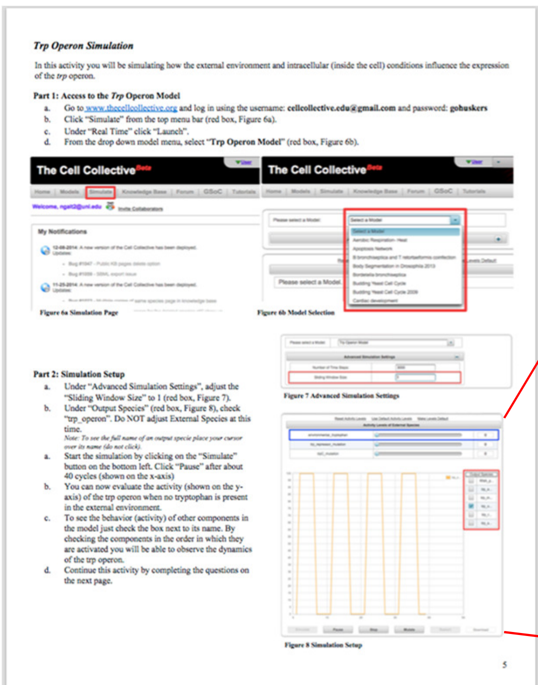
Table 1 Describe/define what each interaction represents in Figure 5.	
a.	RNA polymerase binds to the DNA in the trp promoter and begins to transcribe the trp operon genes into mRNA.
b.	
c.	
d.	
e.	
f.	

4

Table 1 Describe/define what each interaction represents in Figure 5.	
a.	RNA polymerase binds to the DNA in the trp promoter and begins to transcribe the trp operon genes into mRNA
b.	Trp operon transcribes into a single mRNA.
c.	mRNA translates into 5 individual proteins which form 3 enzymes TrpE & trpD (enzyme 1) trpC (enzyme 2) & trpB & trpA (enzyme 3)
d.	The three enzymes produce tryptophan from precursor molecules within the cell.
e.	When levels are high, the tryptophan binds and activates the trp repressor
f.	The trp repressor can change shape & bind to the trp operator when activated. The binding shuts off the trp operon when RNA polymerase is blocked

Computational Model Learning Activities Design

Part 3: Simulation Setup



Trp Operon Simulation

In this activity you will be simulating how the external environment and intracellular (inside the cell) conditions influence the expression of the *trp* operon.

Part 1: Access to the *Trp* Operon Model

a. Go to www.thecellcollective.org and log in using the username: **cellcollective.edu@gmail.com** and password: **gohuskers**
b. Click "Simulate" from the top menu bar (red box, Figure 6a).
c. Under "Real Time" click "Launch".
d. From the drop down model menu, select "**Trp Operon Model**" (red box, Figure 6b).

Figure 6a Simulation Page

Figure 6b Model Selection

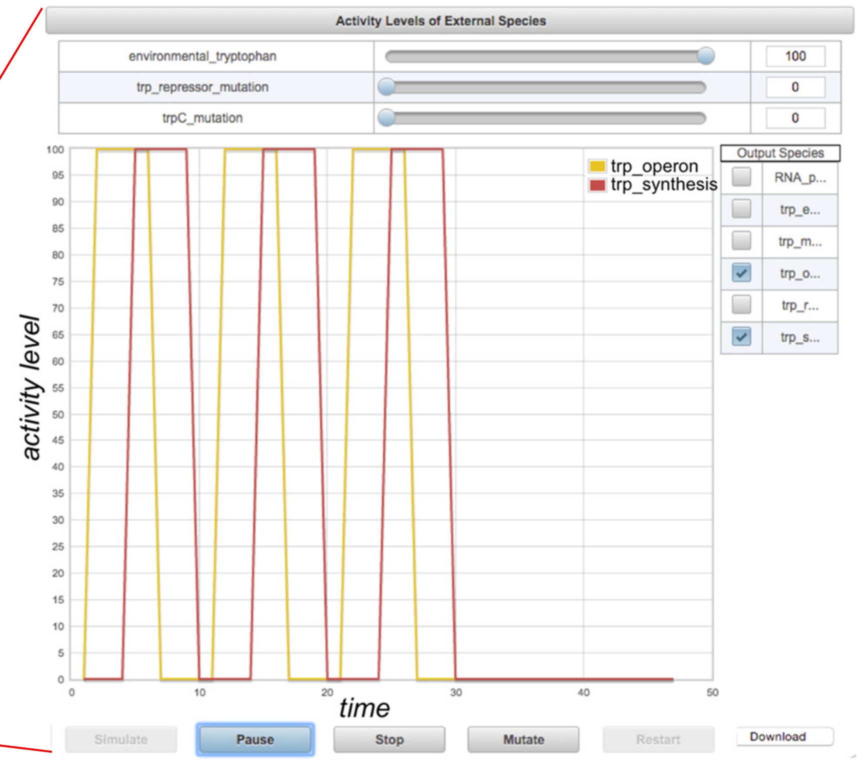
Figure 7 Advanced Simulation Settings

Part 2: Simulation Setup

a. Under "Advanced Simulation Settings", adjust the "Sliding Window Size" to 1 (red box, Figure 7).
b. Under "Output Species" (red box, Figure 8), check "trp_operon". Do NOT adjust External Species at this time.
Note: To see the full name of an output specie place your cursor over its name (do not click).
a. Start the simulation by clicking on the "Simulate" button on the bottom left. Click "Pause" after about 40 cycles (shown on the x-axis)
b. You can now evaluate the activity (shown on the y-axis) of the trp operon when no tryptophan is present in the external environment.
c. To see the behavior (activity) of other components in the model just check the box next to its name. By checking the components in the order in which they are activated you will be able to observe the dynamics of the trp operon.
d. Continue this activity by completing the questions on the next page.

Figure 8 Simulation Setup

5



Computational Model Learning Activities Design

Part 4: Investigations

The purpose of this section is to learn *how* the *trp* operon functions. You will be conducting simulations to verify your own predictions about the dynamic interactions between the components of the *trp* operon. Complete each statement thoroughly.

Consider the following scenarios:

Investigation 1:

a. Predict how the presence of tryptophan in the environment influences the expression of the genes in the *trp* operon. <u>Support your prediction by describing the components involved and how they interact (mechanism).</u>

b. Test your prediction by adjusting the "environmental_tryptophan" slider to 100 (See blue box, Figure 8) (Hint: compare the activity of the trp_repressor, *trp*_operon, etc.). Record and describe the simulation results in the space below.

c. Do your simulation results match your prediction? (circle one) Yes No

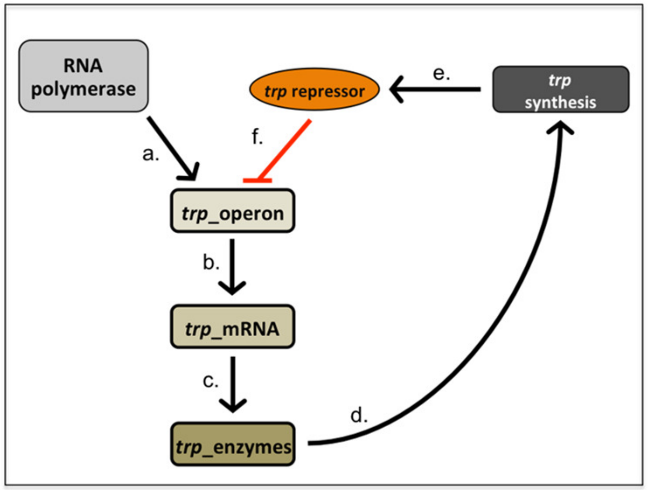
d. Describe what your results indicate is occurring in the cell (use your results to support a mechanistic explanation).

Computational Model Learning Activities Design

Part 4: Investigations

a. Prediction
- Support prediction by describing the components involved and how they interact (mechanism)
- Encouraged to use diagram of the computational model

Computational Model of
Trp Operon

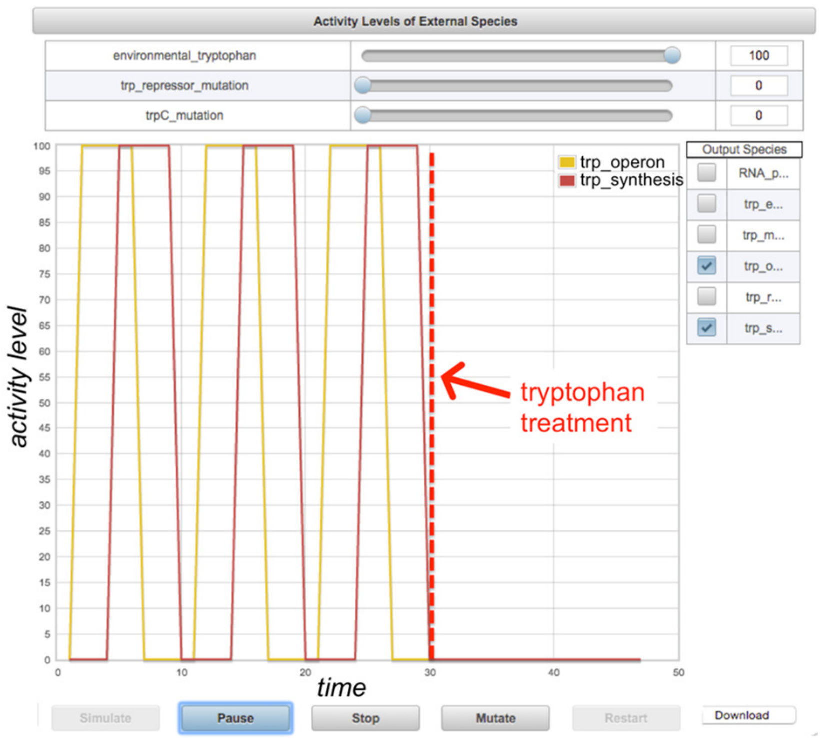


Computational Model Learning Activities Design

Part 4: Investigations

a. Prediction

a. Test Prediction and Record Results



Computational Model Learning Activities Design

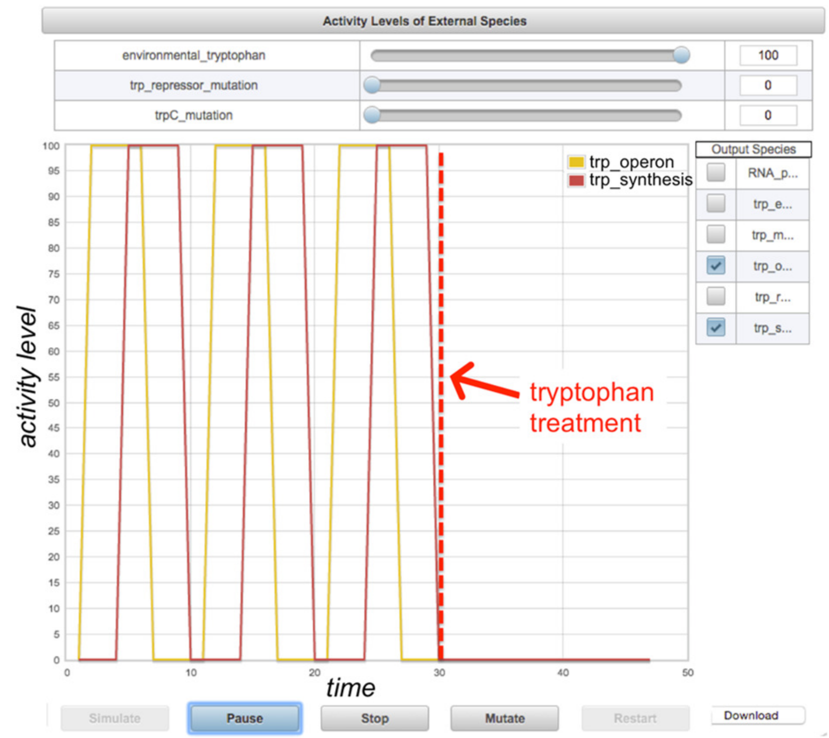
Part 4: Investigations

a. Prediction

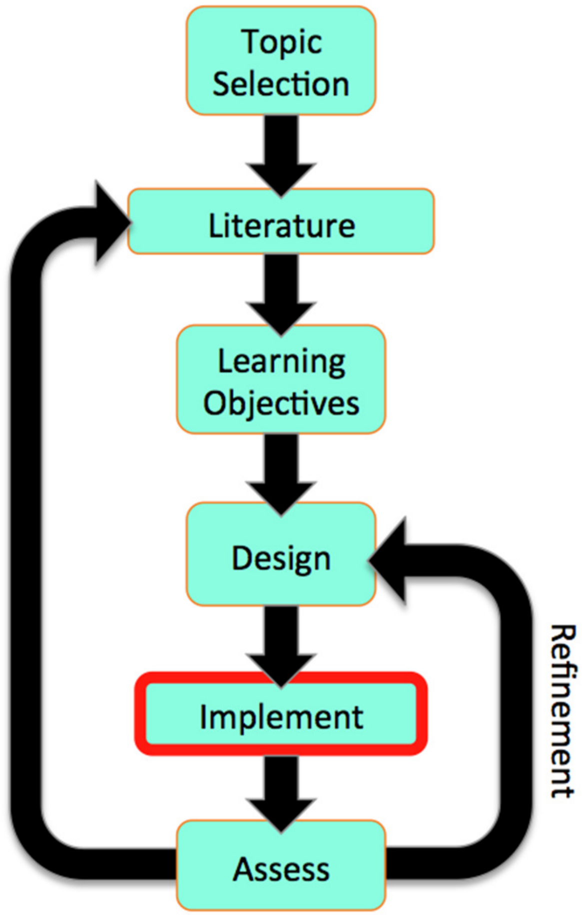
a. Test Prediction and Record Results

a. Describe what the results indicate is occurring in the cell.

- Integrate results into a mechanistic description of the biological process

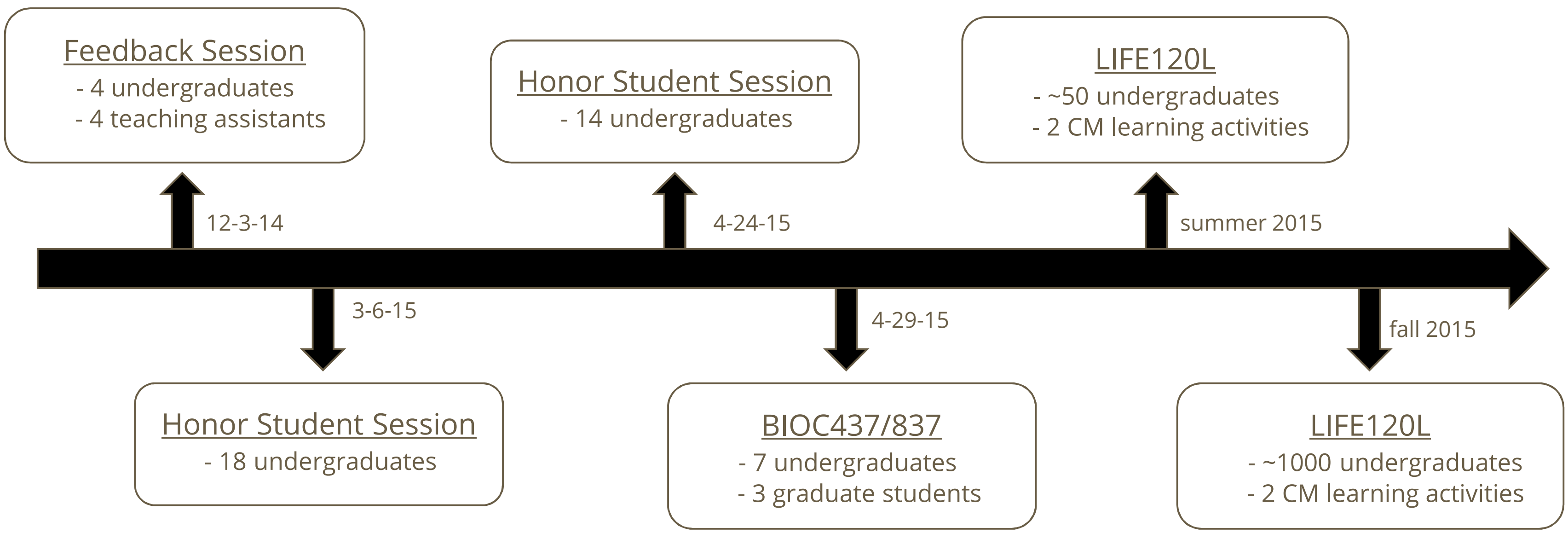


Computational Model Learning Activities



Computational Model (CM) Learning Activities

Implementation Timeline



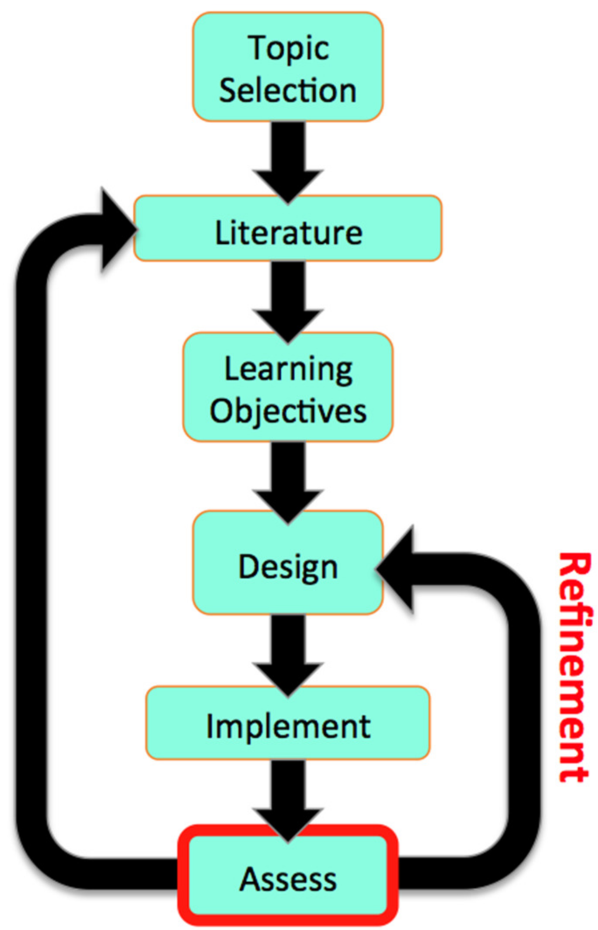
Computational Model Learning Activities

Assessment and Refinement

Systems Thinking
- Conceptual models
- Interviews

Mechanistic Reasoning
- Student responses to module questions

Quality Control
- Student interviews
- TA interviews
- Usability testing
- Classroom observations



Slide 34

2 the formatting is really inconsistent among fonts, colors, bullets, etc.
Heather Bergan,

Evidence-based Refinement of Learning Activities

Refinement Timeline with Preliminary Data

12-3-14

Feedback Session
- clarify directions
- less recording of results

3-6-15

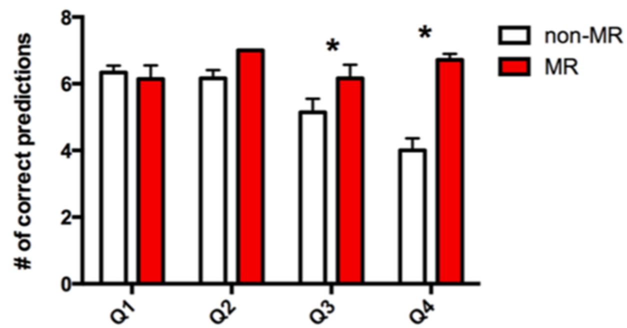
Honor Student Session
- include "hints"
- ask "How" not "Why" questions

Table 1. Prevalence of Mechanistic Reasoning (MR)

	MR	non-MR
Implicit Opportunities (94 Total)	1	93
Explicit Opportunities* (34 Total)	12	22

Evidence-based Refinement of Learning Activities

Refinement Timeline with Preliminary Data



12-3-14

Feedback Session

- clarify directions
- less recording of results

6-9-15

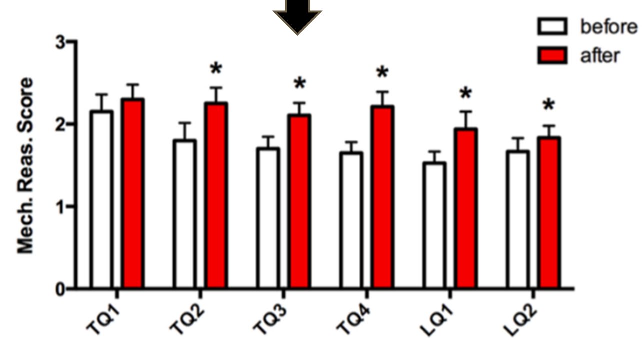
3-6-15

Honor Student Session

- include "hints"
- ask "How" not "Why" questions
- question clarity

Table 1. Prevalence of Mechanistic Reasoning (MR)		
	MR	non-MR
Implicit Opportunities (94 Total)	1	93
Explicit Opportunities* (34 Total)	12	22

7-7-15



Evidence-based Refinement of Learning Activities

Refinement Timeline with Preliminary Data

12-3-14

Feedback Session
- clarify directions
- less recording of results

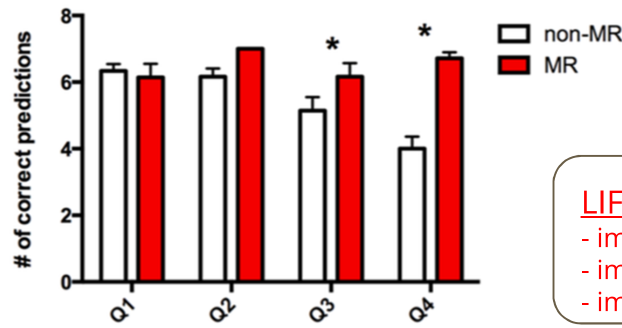
3-6-15

Honor Student Session
- include “hints”
- ask “How” not “Why” questions
- question clarity

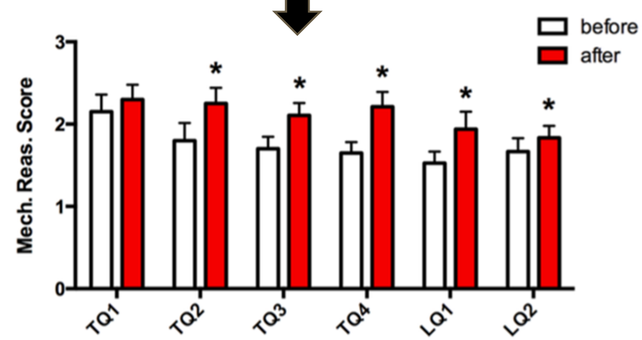
Table 1. Prevalence of Mechanistic Reasoning (MR)

	MR	non-MR
Implicit Opportunities (94 Total)	1	93
Explicit Opportunities* (34 Total)	12	22

6-9-15



7-7-15

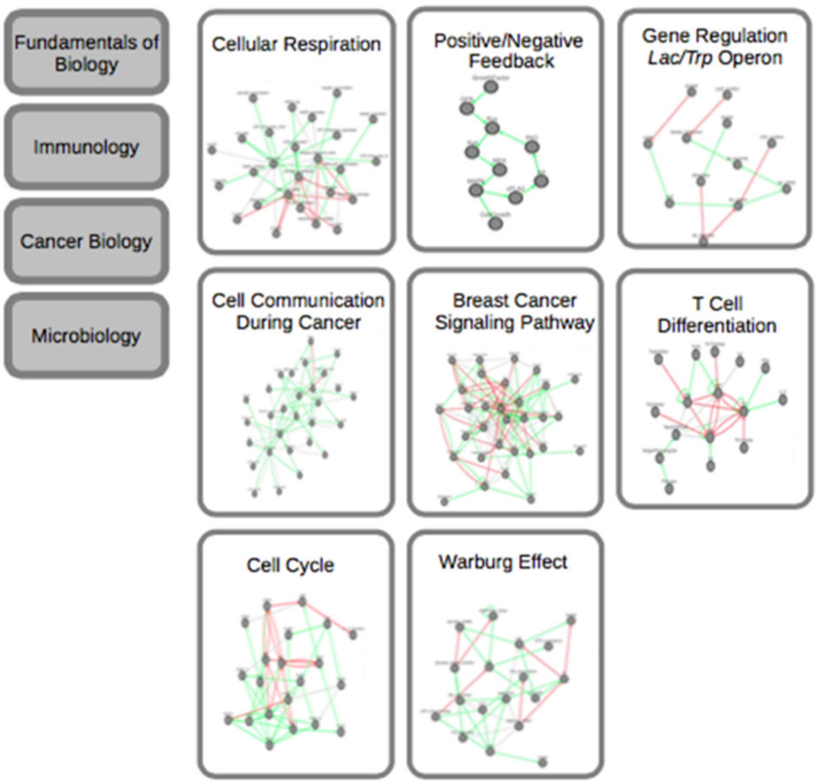


Fall 2015

LIFE120L
- improved systems thinking?
- improved mechanistic reasoning?
- improved content knowledge?

Computational Models and Activities

Currently Available



In Development

Operon Construction

Food Web

Population Dynamics

Endocrine Systems

Audience Feedback

1. Questions?

1. What topics and/or concepts would you be interested in see as a computational learning module?

1. What other elements would you want to see in the activities?

1. How would you want these learning activities to be implemented in your class (e.g., in-class, homework, online courses, labs)

Bonus slides

Data Trp Operon

Results

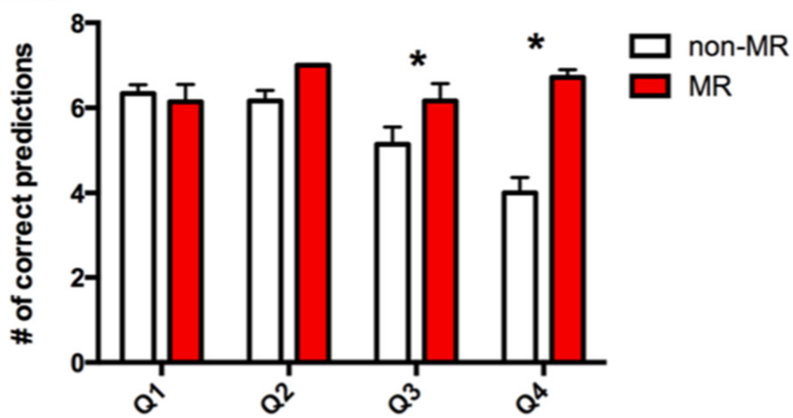


Figure 6: Prediction Correctness
Comparing the frequency of correct predictions when MR was identified in student responses. (Q=question; n=13 per question; mean±SEM, *p<0.05)

Results

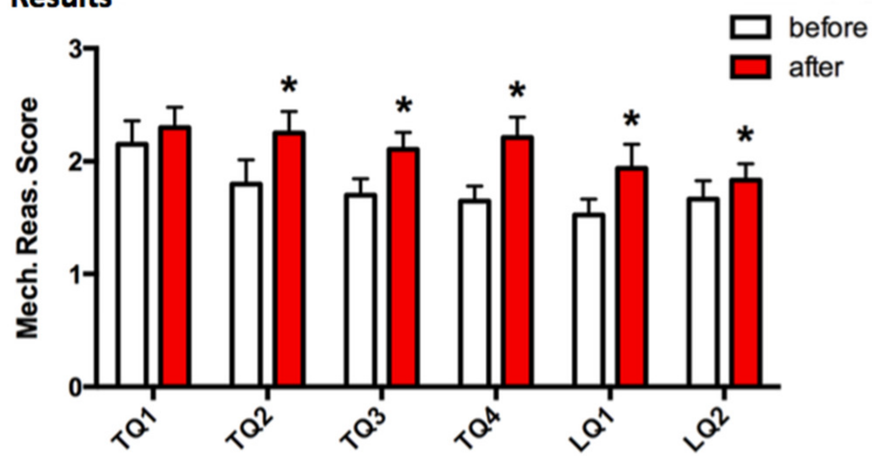


Figure 9: Mechanistic Reasoning *Before* and *After* Simulation
Identifying the effect of the dynamic simulation on MR score. (TQ=*trp* operon question, LQ= *lac* operon question; n=20; mean±SEM, *p<0.05)

Objectives

I tried to preface this on slide 12, fyi

Enable learning about complex biological systems through computational modeling and simulations.

- E.g., by building, simulating, breaking, and re-simulating computer models of biological systems.

Increase systems and dynamical level thinking when learning about biological systems.

Our golas:

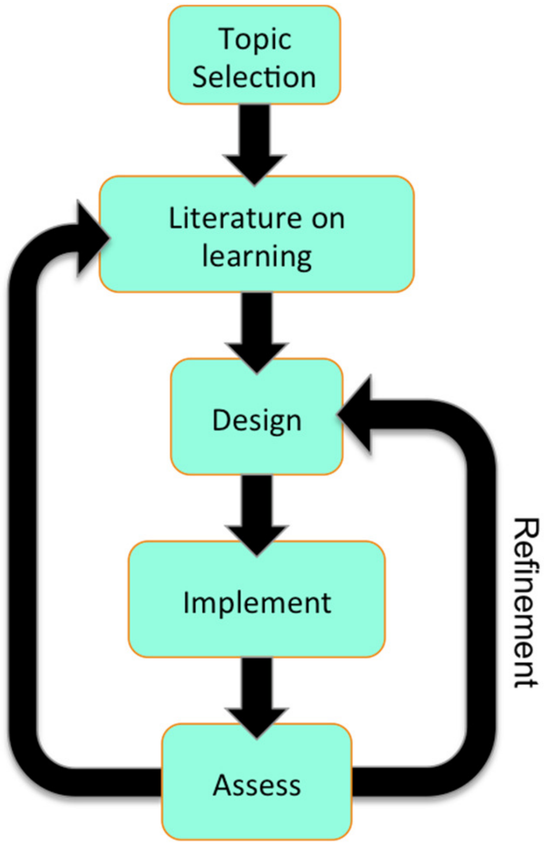
- address misconceptions (evidence based)
- improve systems thinking
- **stand alone! Easy to use**
- in-class, lab, take-home and demonstrations

Computational Learning Modules

1. Topic Selection

1. Identify Learning Objectives
    a. What does the literature recommend?
    b. What do instructors value?

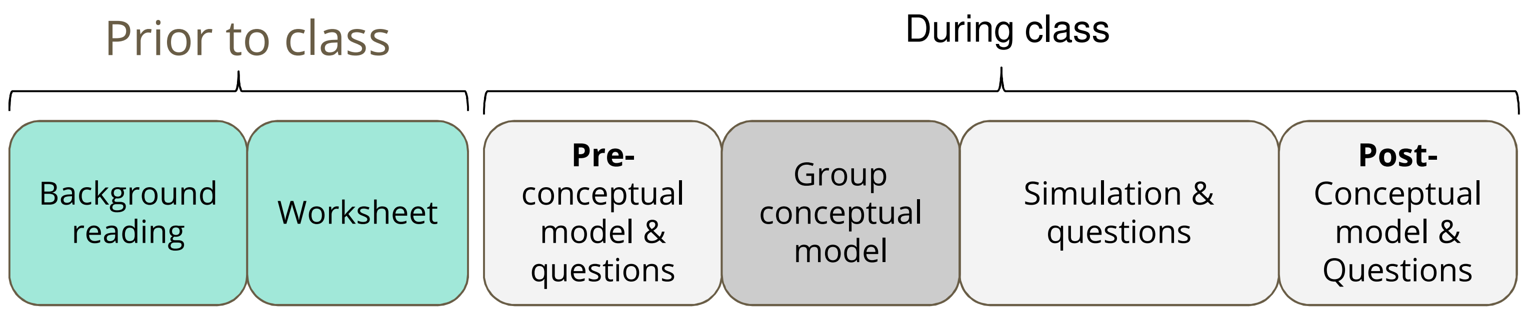
1. Module Design
    a. Background Information
    b. Introduction to Computational Model
    c. Simulation Setup
    d. Inquiry-based Questions
2. Implementation
3. Assessment
4. Refinement



Goals

- designed to stand alone- start here, self-contained; no work for instructors

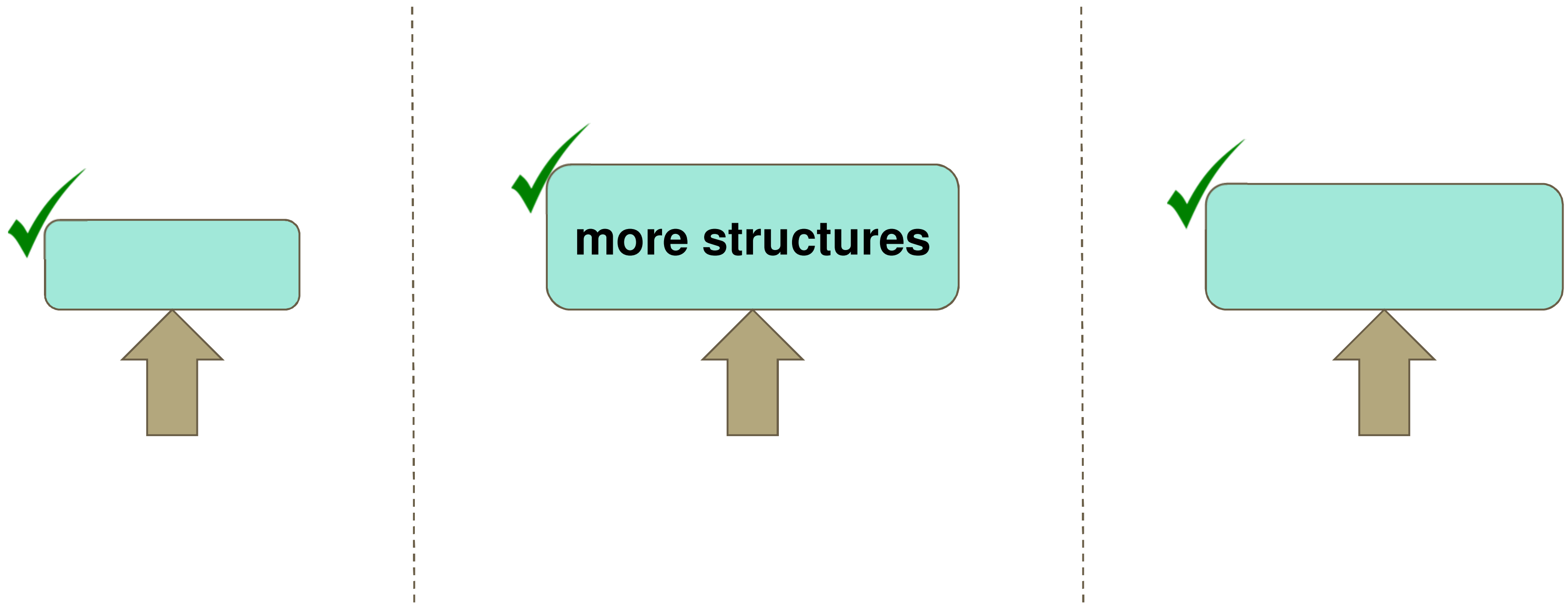
Fall 2015 Experimental Design LIFE 120



n = 543 students

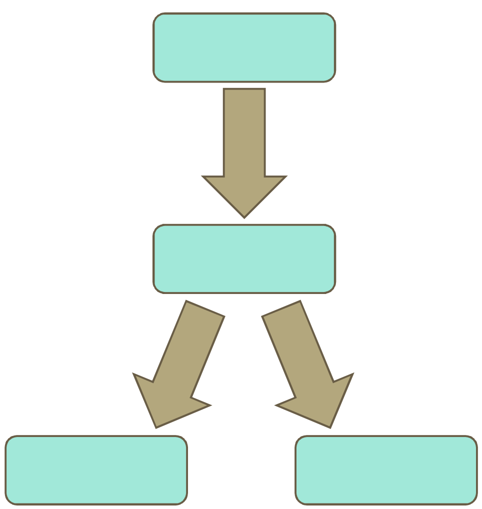
Baseline Findings

Without any lecture instruction | With 1 lecture | With 2 lectures

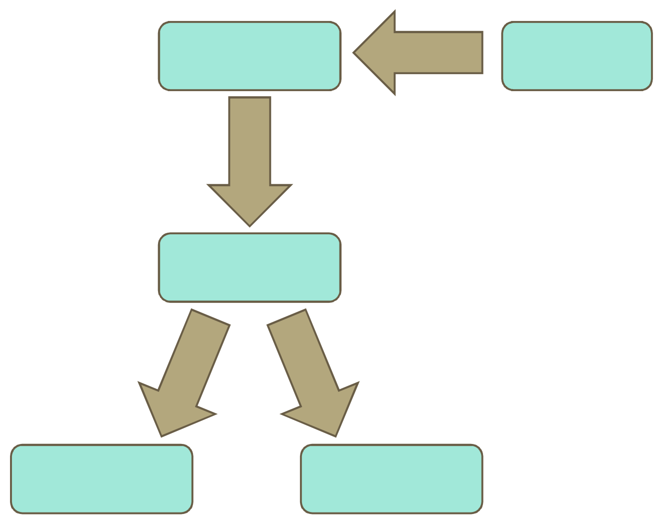


Baseline Findings

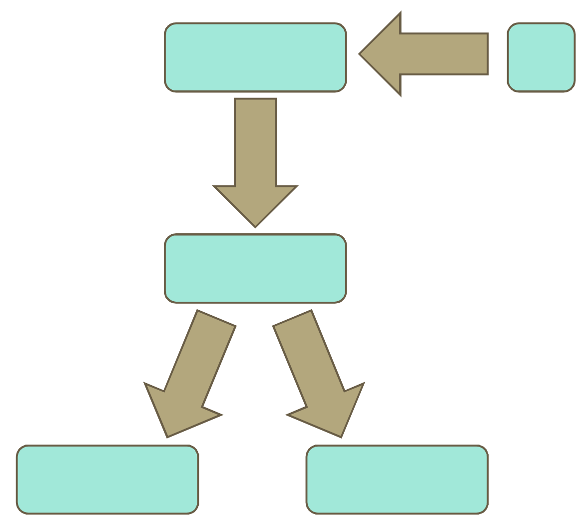
With<u>out</u> any lecture instruction



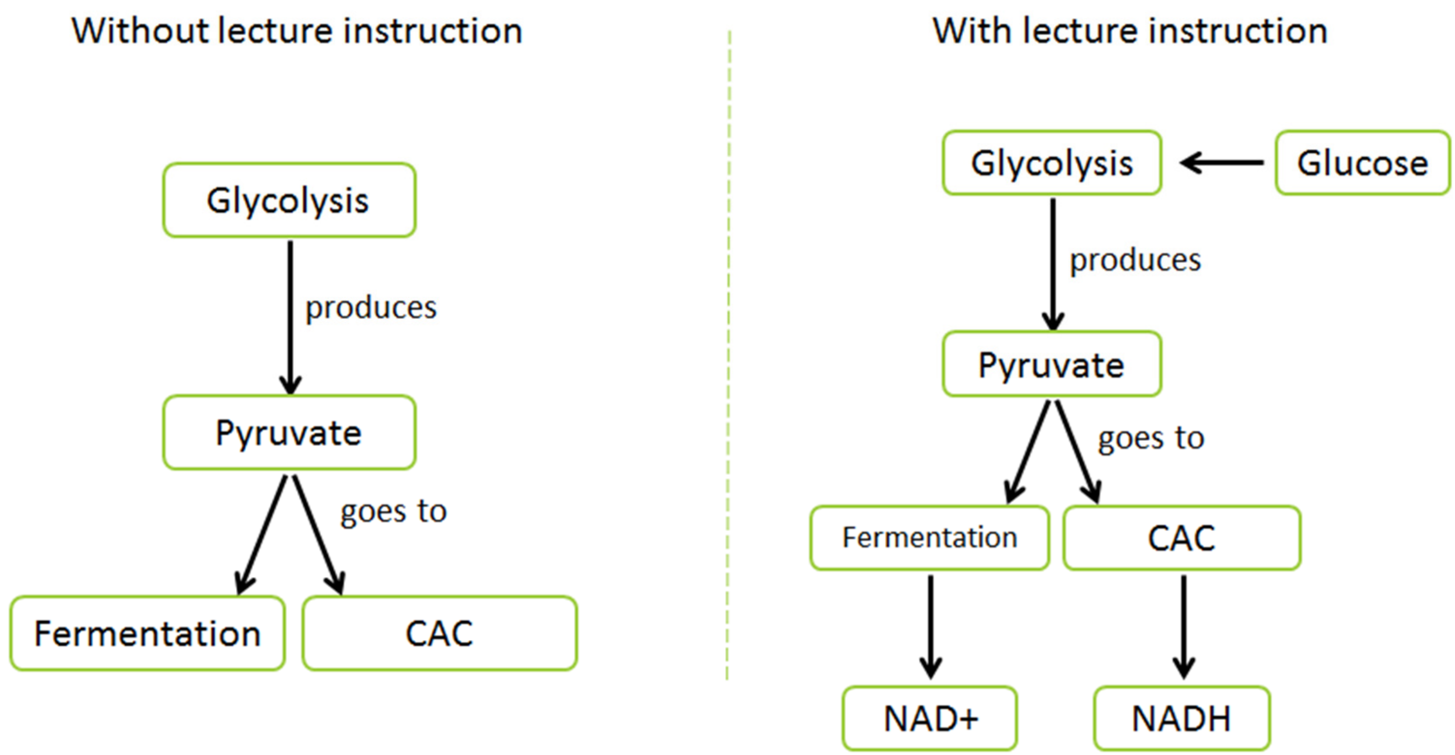
With 1 lecture



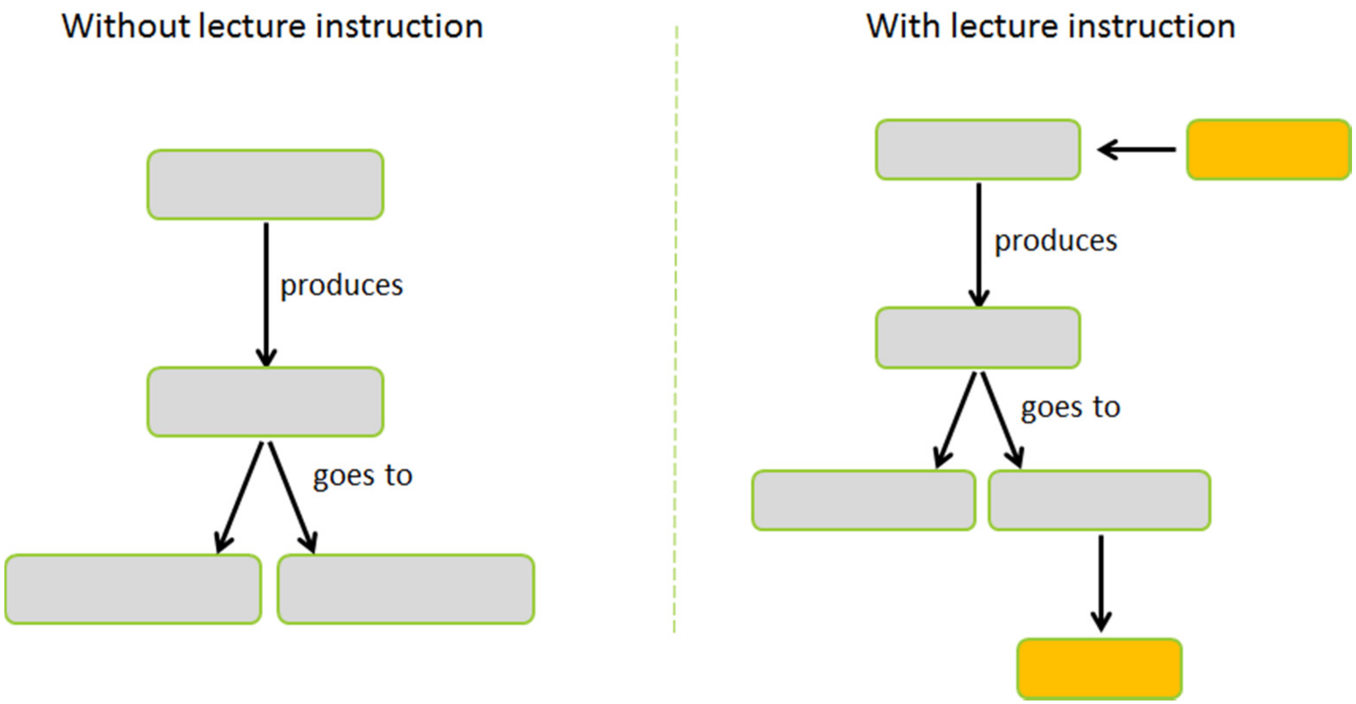
With 2 lectures



Baseline Findings: replace with infograph



Baseline Findings



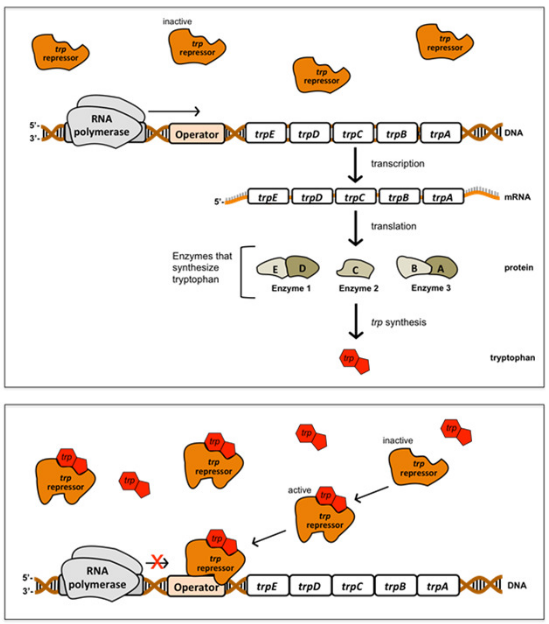
Baseline Findings

Without any lecture instruction	With 1 lecture	With 2 lectures
12.9 $\pm$ 0.9	**17.3 $\pm$ 1.2**	16.8 $\pm$ 2.1

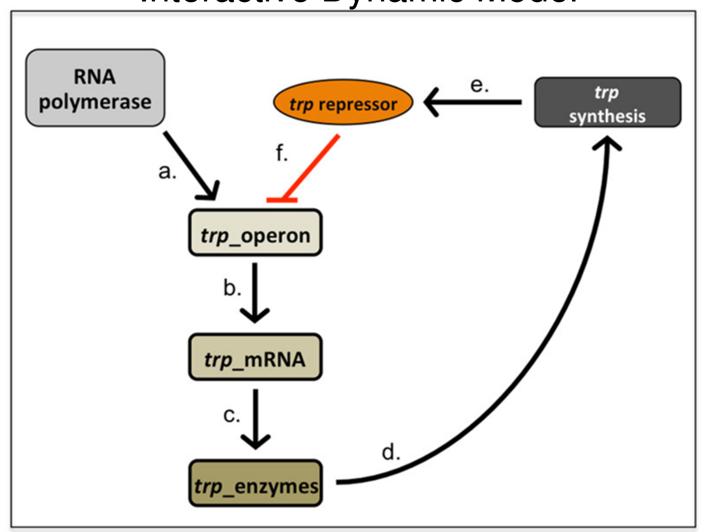
Learning Activities Design

Tryptophan Operon

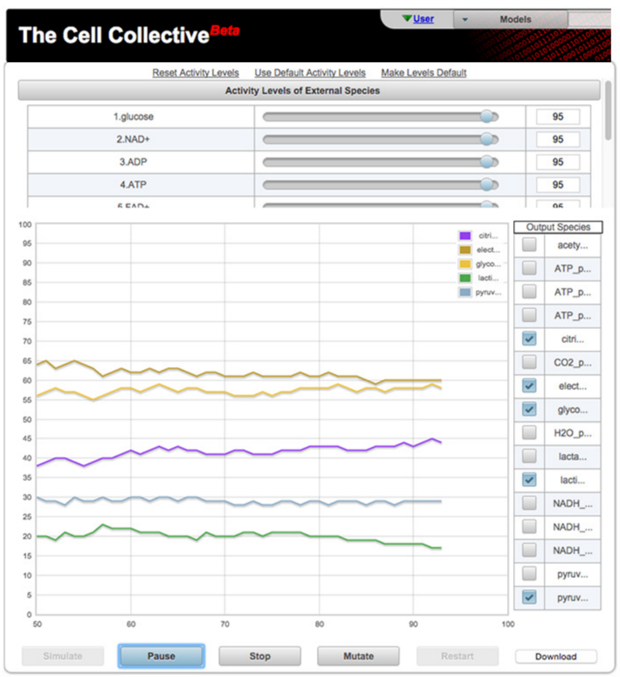
Static Diagrams



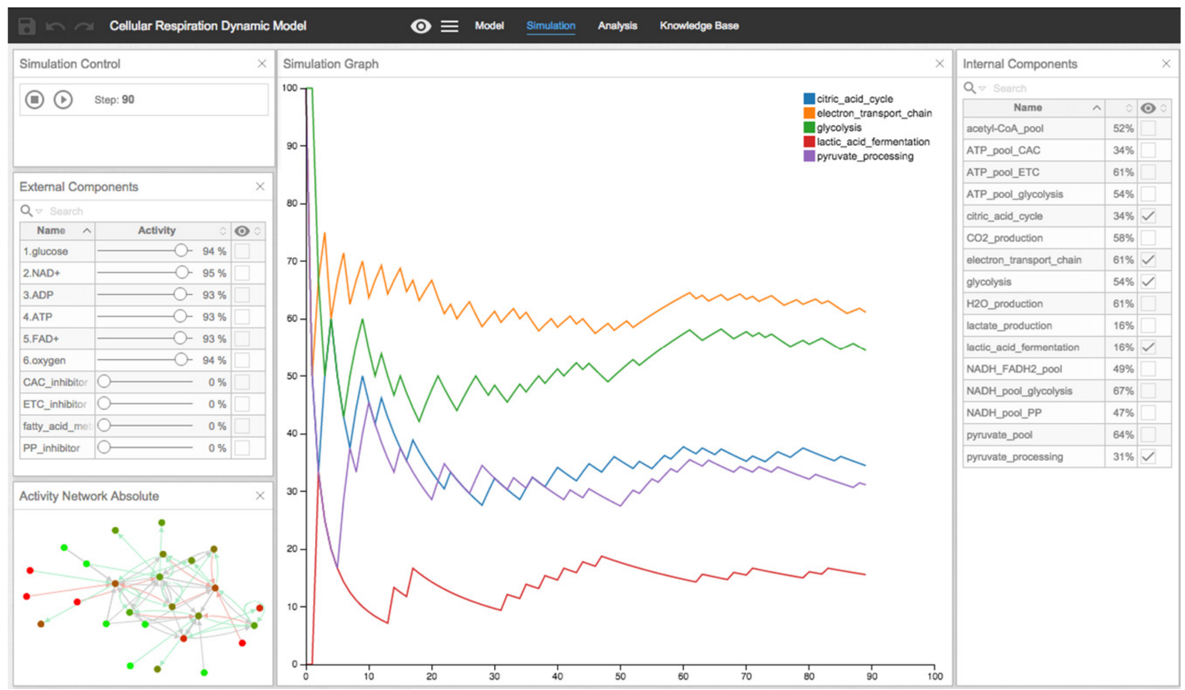
Interactive Dynamic Model



Refinement of Modeling Platform



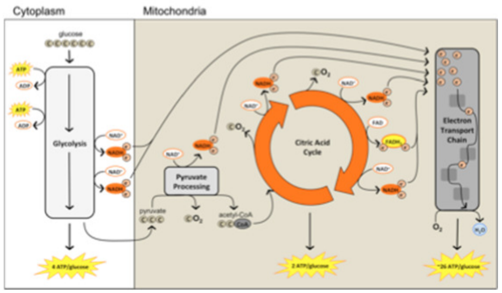
Coming soon...

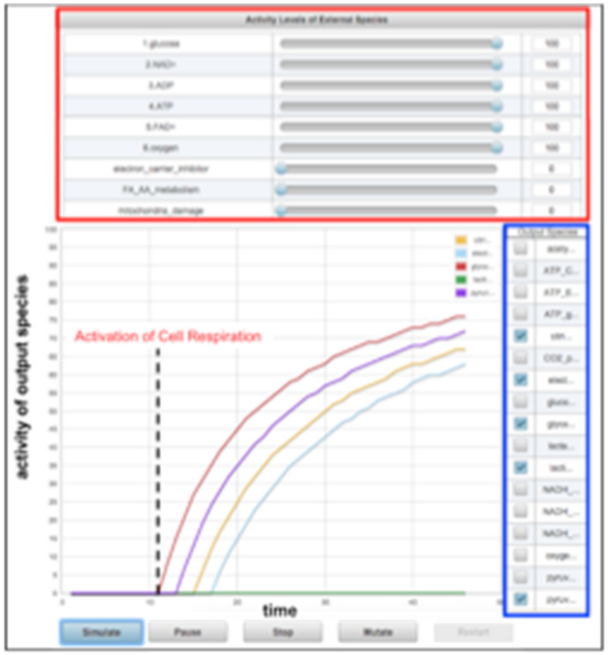


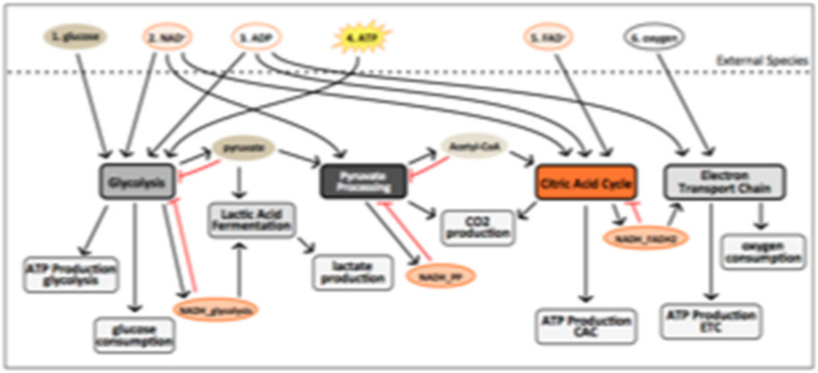
Background on Cell Collective

CC network screenshots

ease of use



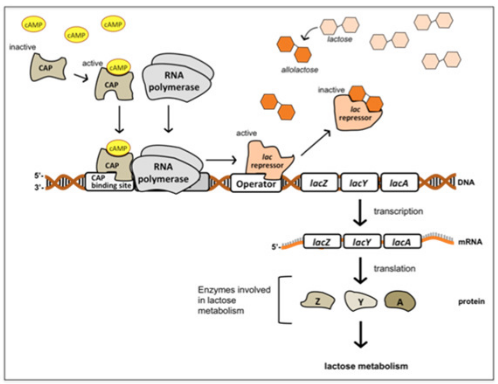


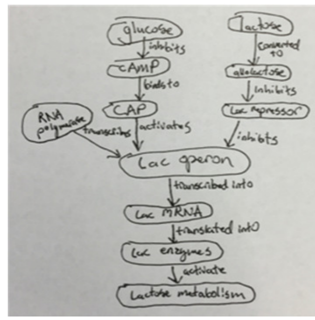


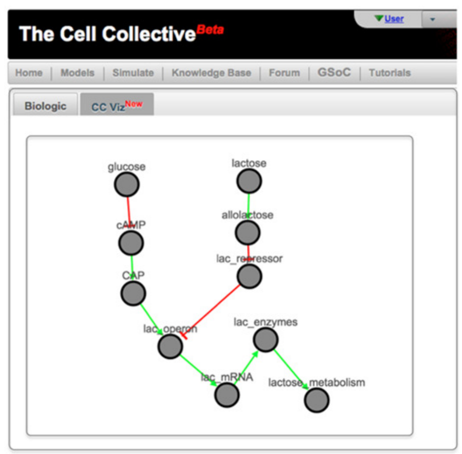
Background on Cell Collective

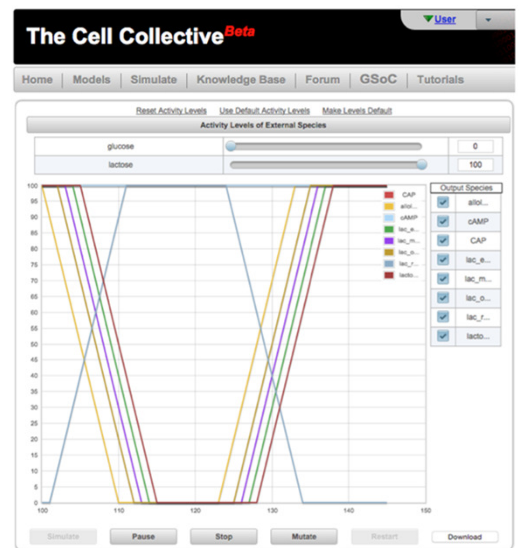
CC network screenshots

ease of use





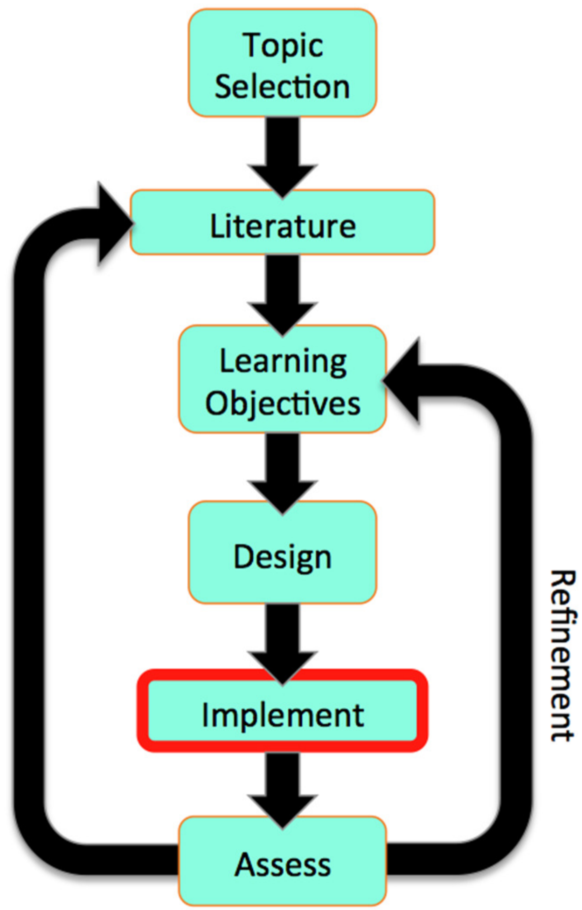




Computational Modeling Learning Activities

Implementation

- Feedback Sessions
  - ~10 undergraduates
  - ~5 Teaching Assistants
- Honor Student Sessions
  - ~15 undergraduates
- LIFE120 Lab Summer Pilot
  - 2 lab sections
  - ~ 20 undergraduates per section
- LIFE120 Lab Fall 2015 Full Implementation
- Upper level



Evidence-based Refinement of Learning Activities

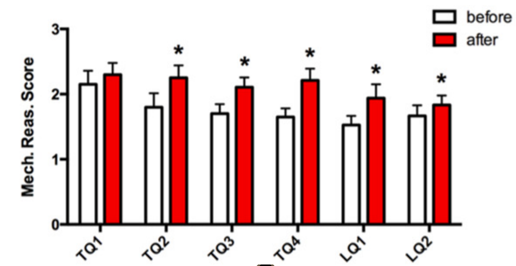
Refinement Timeline with Preliminary Data

Pilot 1
Major findings

date

Table 1. Prevalence of Mechanistic Reasoning (MR)		
	MR	non-MR
Implicit Opportunities (94 Total)	1	93
Explicit Opportunities* (34 Total)	12	22

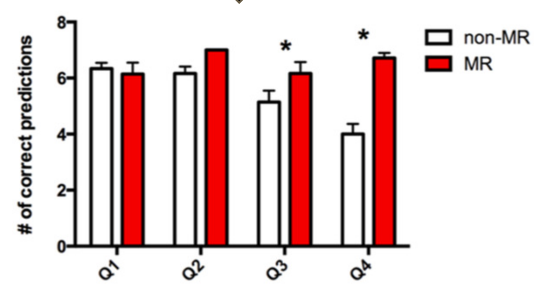
date



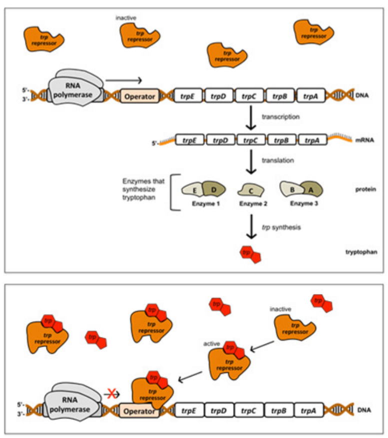
date

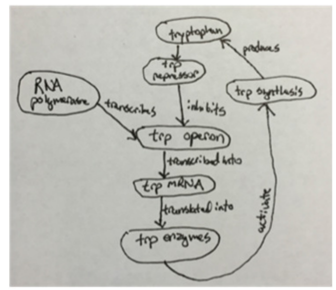
Honors Student Session 1

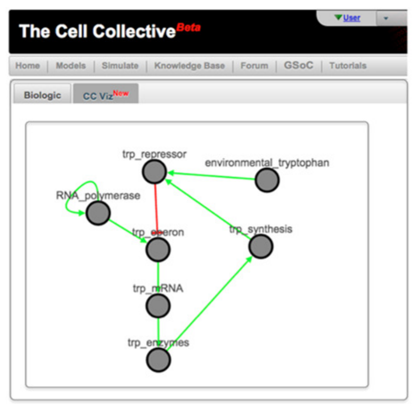
Major findings



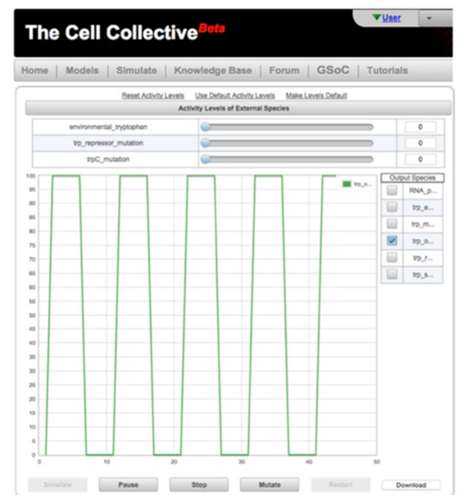
Background on Cell Collective







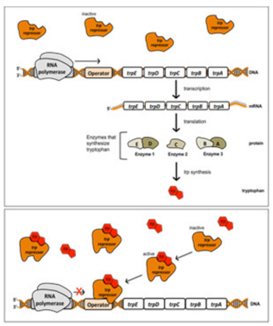
observe dynamics



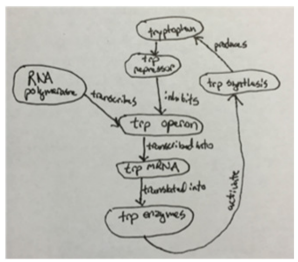
Computational Modeling Platform

Cell Collective

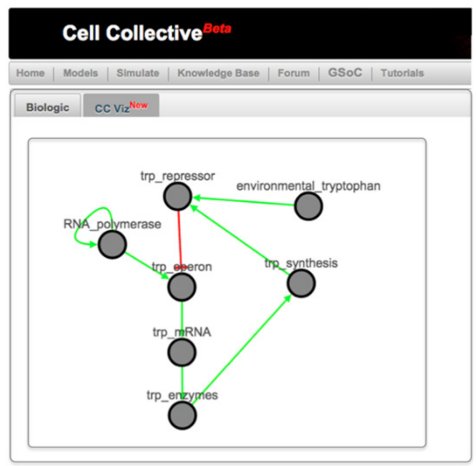
Content Knowledge



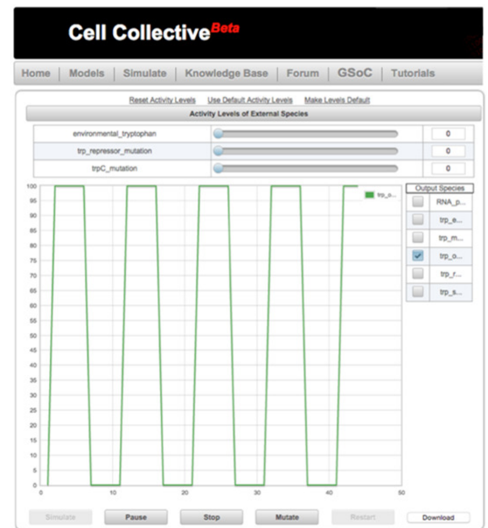
Conceptual Model



Computational Model



Dynamic Simulation



Features:

- Web-based (thecellcollective.org)
- Easy to use
    - No mathematical expression
    - No programming