LS50: Integrated Science

Course overview

LS50 is a two-semester, double course that introduces the natural sciences as an integrated whole. Its goal is to teach students how to solve scientific problems by drawing methods and concepts from biology, chemistry, physics, and mathematics. The course uses examples from biology as an integrating theme, principles from physics and mathematics to reduce complex problems to simpler forms, and computer simulation to allow students to develop their intuition about the behavior of the dynamical systems that control the physical and biological universe. Each semester will include a project lab, in which students will work in small teams to do original research on unsolved biological problems.

Natural science is a single intellectual enquiry into the universe of objects that surround us. Its components are linked by a common method: inducing hypotheses from a mixture of data and intuition, deducing predictions, and testing them by experiment and observation. The sciences depend on mathematics, from the simple act of counting to sophisticated methods required for computational chemistry and theoretical physics. The Integrated Science curriculum will introduce motivated freshmen to the concepts and methods needed to attack the life sciences in the 21st century. For both semesters, students will take the equivalent of two courses, meeting for formal instruction every day, performing hands-on, original research, and using modern computer methods to simulate scenarios and analyze data.

Darwin and Wallace's theory of evolution revealed that living things have a purpose: their structure, function, and behavior are integrated to leave as many progeny as possible. For much of the 20th century, this difference, and the astonishing diversity of form and function, tended to separate biology from the other natural sciences: biology's complexity made it unappealing to many mathematicians, physicists, and chemists, and the "assume a spherical cow" flavor of theorists' simplifying assumptions made biologists skeptical about how useful theory was for understanding biology. Two advances have pushed biologists towards theorists and computer scientists: the need to test our understanding of biological processes by making explicit, mathematical models and the need to convert large datasets into information and, ultimately, knowledge.

We teach students that the answer to "How will you solve this problem?" is "By any means necessary!" Our goal is to teach them how to find interesting problems, the means to solve them, and above all, the knowledge and courage to invent the new methods that make previously insoluble problems soluble. Coupling concepts and methods to problems that excite students and making them use these tools in their own research will embed the concepts in their working memory.

We teach through iterated cycles of experiment and analysis, making use of experimental computation to simulate a system of interacting entities and explore the effect of parameter variation on the system's properties. Our goal is to complement the formal derivation of theorems, show the productive interplay between theory, simulation, and experiment, and show that computer systems and programs, like biological objects, have purposes. Mastering a restricted syntax to write algorithms will help students think about how biological systems use the restricted syntax of chemistry and genetics to accomplish tasks. Concepts like modularity, exploration with selection, error detection and correction, and recycling previous inventions are important in the function and evolution of both organisms and code. Five faculty will teach the course, working in pairs of one life scientist and one physical scientist.

Students will use their knowledge to conduct original scientific research. The project labs will be based on the research of and run by faculty and JHDSF Fellows, independent scientists who spend five years at Harvard after their PhDs and run small research groups. We assume don't assume that students have any prior experience in scientific research.

Class format

Lectures M, Tu, W, Th, F from 10:30-11:45 am, in the Northwest building room B108 (NWB108).

Mid-terms: Three 75 minute midterms each semester. Each midterm covers the preceding month's worth of lecture material.

Spring term only: Wednesday activities, W, 10:30-11:45 am. These will be of various types, including lectures on faculty research and discussion of scientific papers and careers.

Weekly discussion sections with teaching fellows, at locations and times to be determined based on student availability.

Weekly problem sets, to be submitted as indicated on each assignment.

Research-based laboratory: students work on original research projects in small groups, with open access to a sophisticated modern research laboratory. Students are expected to spend 3-6 hr/week working on their projects. Labs will meet at times to be determined, probably Tuesdays and Wednesdays from 1-4:00 pm (students choose which day they'll attend).

Faculty

Ben de Bivort	OEB
Michael Desai	OEB
Cassandra Extavour	OEB
Aravi Samuels	Physics
Andrew Murray	MCB, Course Head

The year-long course will be divided into thirds, each team-taught by a pair of faculty, one from the physical and one from the life sciences.

Laboratory Heads

Sergey Ovchinnikov	John Harvard Distinguished Science Fellow
Emily Nagy	College Fellow

Teaching Fellows

Mariela Petkova	Biophysics	Early fall
Emma Nagy	MCB	Early fall
Daniel Gainey	Biophysics	Late fall
Matthew Melissa	Physics	Late fall
Mike Jin	Systems Biology	Early spring
Milo Johnson	OEB	Early spring
Ceejay Lee	MCB	Late spring
Anurag Limdi	MCB	Late spring

Syllabus

Below we list the topics covered, week-by-week for the fall and spring semesters. Since the course will meet every day for two semesters, it will have roughly 100 lectures. As a result the description below is brief and lists weekly topics rather than those of individual lectures. A full schedule is on the course website.

Fall Week	Topics
1	Intro and course overview. Biology as computation. Central Dogma. Intro to probability. Equilibria, energy, and steady state.
2	Probability and Bayesian analysis. The Lac operon as a paradigm for gene regulation. Orbitals, elements, periodic table.
3	Molecular orbitals, organic molecules, molecular forces, proteins. Math primer, introduction to differential equations, linear algebra.
4	Macromolecules. Linear algebra. Molecular genetics.
5	Molecular genetics. Principal component analysis,
6	**EXAM 1**. Cell division cycle.
7	Computation and biological sequence analysis. Cell division cycle. Classical genetics.
8	Computation and biological sequence analysis. Statistical mechanics.
9	Computation and biological sequence analysis. Statistical mechanics.
10	**EXAM 2.** Statistical mechanics. Cellular energy budget. Cooperativity.
11	Cooperativity. Diffusion.
12	Membranes. Photosynthesis. THANKSGIVING!
13	ATP synthesis. Diffusion limited reactions. Reaction-diffusion systems. Gene regulation.
14	**EXAM 3**. Gene regulation.

Spring week	
1	Graphs & networks. Dynamical systems. Chemotaxis.
2	Dynamical systems. Chemotaxis.
3	Dynamical systems. Intro to neuroscience. Hodgkin-Huxley axon model. Non-linearity in neuroscience.
4	Stochastic phenomena in biology. Symmetry breaking. Animal navigation & behavior.
5	**EXAM 1**. Markov chains. Behavior as a Markov chain. Simulating stochastic processes.
6	Markov chain Monte Carlo. Simulating behavior. Structure determination. Fourier Analysis.
7	Fourier analysis. Nucleic acid & Protein structure. Protein folding.
8	Enzymes. Microscopy. Cell structure.
9	**EXAM 2.** Cytoskeleton. Introduction to evolution and animal development.
10	Population genetics. Embryonic development.
11	Population genetics. Evolution of development. Symmetry breaking in development.
12	Molecular Evolution. Evolution of development. Origin of life.
13	**EXAM 3**. History of life.

Matlab/Python

We assume no prior experience in writing computer programs, but a major goal of the course is to teach scientific computation, using the language Matlab or Python. Unless they have previous experience with Matlab or Python, students will participate in a boot camp run by Adam Cohen and colleagues (Matlab) or Sergey Ovchinnikov (Python). Example of information can be found here: https://wiki.harvard.edu/confluence/display/fasmatlab (you need a Harvard key to access).

Problem sets will have questions that depend on your ability to use Matlab/Python, but we will increase their difficulty gradually so as not penalize those encountering a computer language for the first time.

Assessment

Assessment will be based on five criteria:

In-class tests
We will administer three in-class tests, each 75 min long and each covering the material presented in the preceding month. The question format and content will be similar to the weekly problem sets. Students can bring a single-sided 8.5 x 11 inch piece of paper, with any information they wish to remember to each exam and we will provide a sheet with key formulae and equations.

Problem sets
Weekly problem sets will review the topics covered in lectures and discussion sections. We strongly encourage students to collaborate to solve the problems, which may include a comparison of scratch work, methods used, and numerical results. All submitted work should be written independently and reflect the students' own understanding. Students should not submit work that they would be unable to explain or reproduce on their own.

This policy reflects our intent in assigning the problem sets: to help you master the material under low-stress circumstances so you will not be blindsided by higher-stakes tests. While you should be wary of over-reliance, classmates are an excellent resource for help since they keep similar hours, live nearby, and remember their own learning process well enough to explain concepts appropriately. On the flip side, mentoring colleagues whenever possible is not merely kind and forward-thinking, but also helps crystallize your understanding and practice the mentorship and presentation skills needed for work in the scientific community.

Laboratory research
Students will be graded on the research projects they undertake, focusing on their ability to design and interpret experiments and their commitment to and engagement with their research more than on the quality of the results they produce (subject as these are to the whims of the Gods). Approximately half of the grade from laboratory research will reflect weekly participation in lab, with the remainder attributed to a final lab report.

Participation
Students will be graded on their participation in class, which we expect to be highly interactive, weekly conferences, and the discussions associated with laboratory research.

These criteria will be accorded the following weights:

In-class tests	40%
Problem sets	30%
Laboratory research	20%
Participation	10%