

Week 1: Introductions

Intro to Bioinfo & Compbio

- Lay of the bioinfo-compbio land
- Reading papers | Framing the problem
- Choosing a good problem
- Resources @ MSU
- Getting help

Bioinformatics & Computational Biology

Computational biology

- The study of biology using computational techniques.
- Goal: learn new biology, knowledge about living systems. *It is about science.*



Margaret Dayhoff – One of the first bioinformaticians

Applying math & computational techniques to the sequencing of proteins and nucleic acids.

- 1965: First collection of protein seqs. Single-letter code for amino acids.
- 1966: 'Evolutionary trees'.
- 1978: First AA similarity-scoring matrix.
- 1980: Launched the Protein Information Resource, the first online database system that could be accessed by telephone line.

Bioinformatics

- The creation of tools (algorithms, databases) that solve problems.
- Goal: build useful tools that work on biological data. *It is about engineering.*

Bioinformatics & Computational Biology – Today

“Computational thinking and techniques are so central to the quest of understanding life that today **all biology is computational biology**.

- Brings order into our understanding,
- Makes biological concepts rigorous and testable, and
- Provides a reference map that holds together individual insights.

The next modern synthesis in biology will be driven by mathematical, statistical, and computational methods being absorbed into mainstream biological training, turning biology into a quantitative science.”

Shifting roles of computational biologists

	Past	Current
Role in research	Supportive	Driver of research
A feeling for the biology	Computer science-centered	Biology- and computer science-centered
Environment	Isolated	Integrated
Data generation	Constrained	Resourceful
Data exploration	Largely limited to hypothesis testing	Both exploratory and hypothesis testing

Markowitz (2017) PLoS Comp. Biol.

Yanai & Chmielnicki (2017) Genome Biol.

Some broad research areas & related analytical methods

- Sequence alignment and search
- Genome assembly and annotation
- Molecular evolution and comparative genomics
- Quantitative genetics
- Regulatory genomics
- Functional genomics and data integration
- RNA/Protein structure prediction
- Molecular docking and dynamics simulations
- Artificial life and digital evolution
- Biological networks
- Modeling signaling, regulatory pathways
- Metabolic reconstructions and models
- Dynamic programming
- de Bruijn graphs, Hidden Markov Models
- Tree construction, Suffix trees
- Statistical inference, Multiple testing
- Expectation maximization, Gibbs sampling
- Dimensionality reduction, Machine learning
- Maximum entropy modeling
- Atomic, physical simulation
- Agent-based modeling
- Graph theory, Label propagation
- Dynamical simulation, State space, Bifurcations
- Linear programming

Data types and repositories – some examples

Genomes & proteomes

all encompassing

Ensemble

comparative genomics

COGs | InParanoid | OrthoMCL

ref. gene/transcript sequences
& annotations

RefSeq | Entrez | GENCODE

sequences variation

1000 Genomes | dbSNP

everything protein

UniProt | InterPro | SCOP | CATH |
PDB

Functional annotations & relationships

biol. processes, mol. functions,
cellular components

Gene Ontology

pathways

Reactome, KEGG, WikiPathways

networks

BioGRID, TRANSFAC, STRING

Phenotype-, Disease-association

OMIM | GWAS Catalog | ClinVar |
COSMIC

Genome-Phenome

dbGaP | UK Biobank

Functional/regulatory genomics

data sets

NCBI GEO | EBI ArrayExpress

raw reads

NCBI SRA | EBI ENA

consortia

ENCODE | Roadmap | GTEx | TCGA

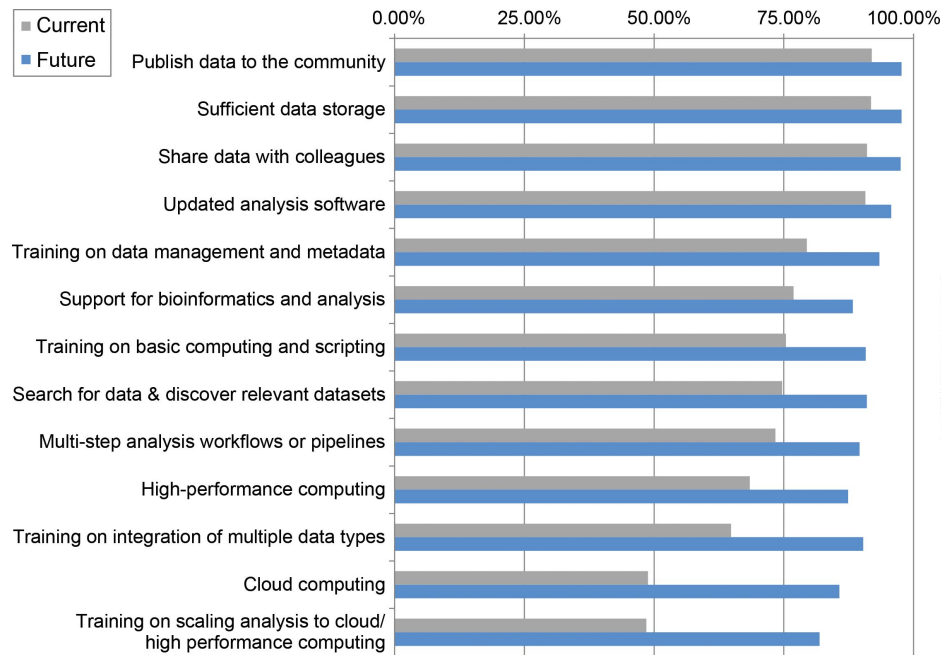
curated public data

Dryad | Repositive | Expression Atlas

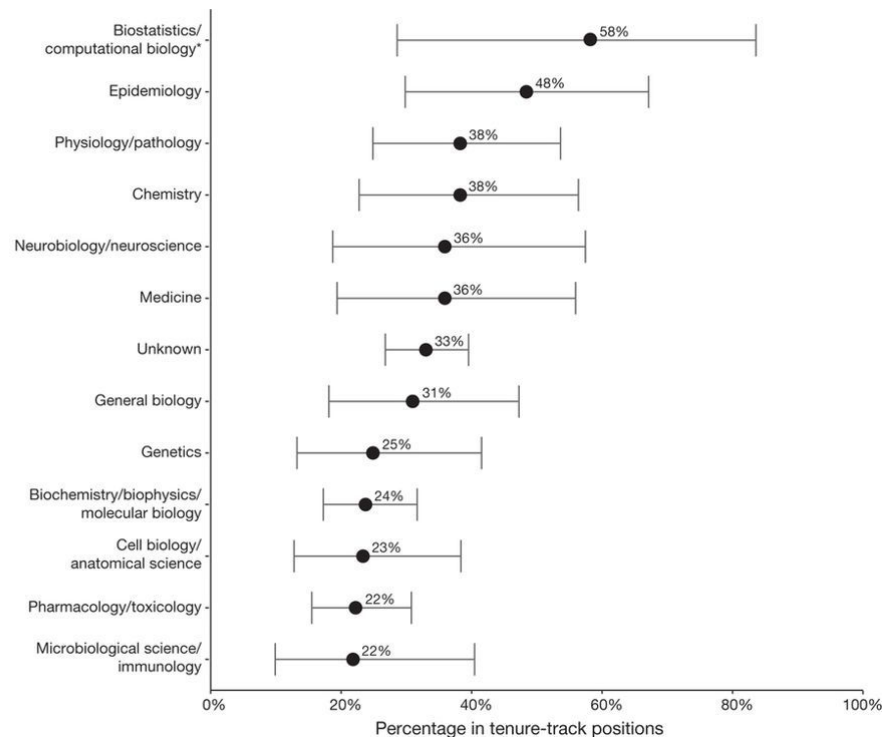
Model organism databases

MGI | RGD | TAIR | FlyBase | WormBase
| ZFin | SGD

Opportunities & Unmet needs



Doctoral degree field



Community – Meetings / Conferences; MSU Seminars

Some National/International Conferences

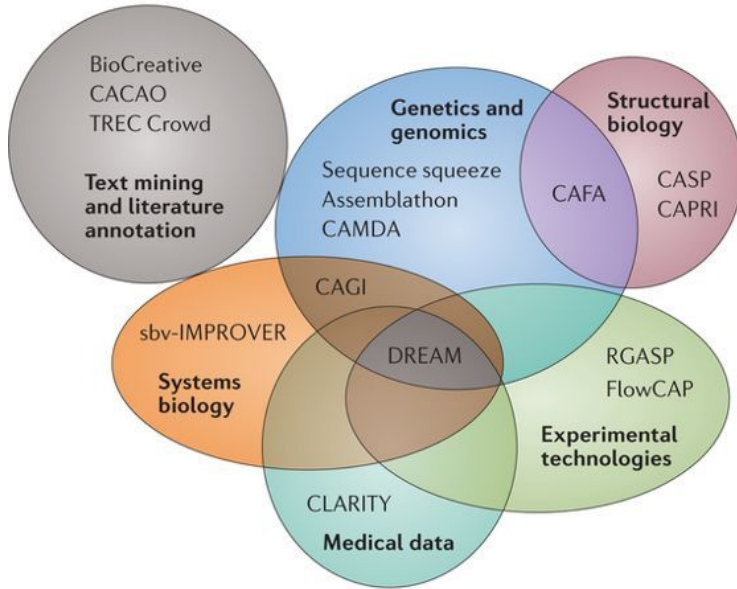
- Intelligent Systems for Molecular Biology
- Research in Computational Molecular Biology
- Pacific Symposium on Biocomputing
- ACM Conference on Bioinformatics, Computational Biology, & Health Informatics
- Rocky Mountain Bioinformatics Conference
- Great Lakes Bioinformatics Conference
- Cold Spring Harbor Laboratories Meetings (Network Biology, Genome Informatics)

Relevant MSU Seminar Series

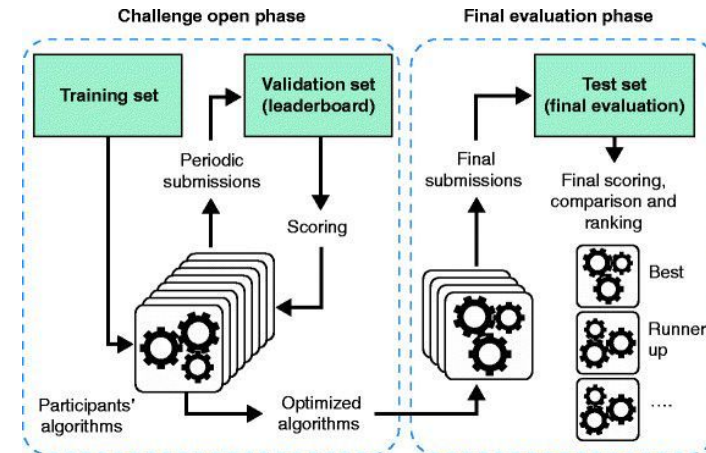
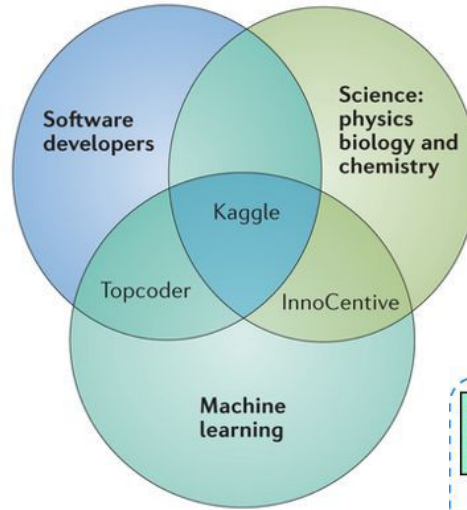
- Science at the Edge
- Machine Learning
- Computational Math, Sci, & Egr
- Institute for Quant. Health Sci. & Engg

Community – Open Challenges

Researcher-driven domain-specific Challenges



Intermediary commercial Challenge platforms



Open science

Code: The field has dramatically shifted in thinking on how to publish code.

- Code used in research should be made available for research use free of charge.
- This is not just code for downloading & using. Original code must be made publicly available for others to use, review, and edit.
- Most common way to share code: GitHub.

Scientific publishing: Preprints

- Rapid publication of new science + free access (e.g. bioRxiv).
- Major source of cutting-edge research.
- Can have multiple (progressively better) versions of each manuscript.
- Preprints have NOT been peer-reviewed for quality and soundness of science.
So, read/use with caution.

Reading primary research papers – Learning to frame the problem

Great way to learn how to frame a problem, choose the methods/tools, set up an analysis workflow, establish groundwork, & generate a series of supportive results towards answering the central question.

Types of computational research studies

- New analytical/computational method
- Improvement of an existing method
- Evaluation of existing methods
- Development of (re-)usable software, web-service, or database
- New insights w/ new/existing methods

Journals to follow

Bioinformatics	Cell
bioRxiv Bioinformatics	eLife
bioRxiv Genomics	Nature
BMC Bioinformatics	Nature Biotechnology
BMC Genomics	PNAS
Briefings in Bioinformatics	Science
Cell Systems	Science Translational Med.
Genome Biology	
Genome Research	
Molecular Systems Biology	
Nature Genetics	PubMed Alerts
Nature Methods	Google Scholar Alerts
Nucleic Acids Research	
PLoS Computational Biology	
Cell Systems	

Reading primary research papers

Title & Abstract

1. Use **Title, Abstract, & Figures** to select a paper. Read them again last!

Introduction

2. Read the **Introduction**:

Data & Methods

- a. Identify *the* question. What is the big challenge the authors are trying to solve?
- b. What are the then current approaches for solving that problem? What are their limitations that, according to the authors, need to be addressed?
- c. What are the *specific* questions this paper is setting out to answer?

Results

Discussion

References

Reading primary research papers

Title & Abstract

3. Read **Data & Methods**: [Be critical!]

Introduction

- a. For each specific Q, note data (type & source) & method (algorithms/techniques, software, & approach).
 - i. ALWAYS read the **Supplemental Materials**. These days much of the good stuff is in here!
- b. Are the data & methods describes sufficient to answer the Qs raised in the Intro?
- c. Make detailed notes on: 1) what's unclear, 2) what you might do differently.

Data & Methods

Results

Discussion

References

Reading primary research papers

Title & Abstract

Introduction

Data & Methods

Results

Discussion

References

4. Read the **Results**: [Be critical!]

- a. Go figure-by-figure, panel-by-panel. Based on your reading of Data & Methods, is there enough information to know/reproduce that analysis?
- b. Try to interpret each figure/panel, then read the figure legend and the part of the results that explains it. [**Supplemental figures/tables** abound!]
 - i. Do your interpretations match that of the authors'?
 - ii. Are the results answering the specific Qs?
- c. Make detailed notes on: 1) what's unclear, 2) what you might do differently.

Reading primary research papers

Title & Abstract

5. Read the **Discussion/Conclusions, Title, & Abstract**:

- a. Step back to think about contributions, limitations, open Qs, & next steps.

Introduction

Data & Methods

6. Read what other researchers (**papers that cite this paper**) say about this work.

Results

Reading a paper can be overwhelming.

Discussion

- Read the paper in phases & more than once.
- It is perfectly fine if things are not clear on first pass. Happens to everyone.
- Understanding will *always* improve & the big picture will emerge with re-reading.

References

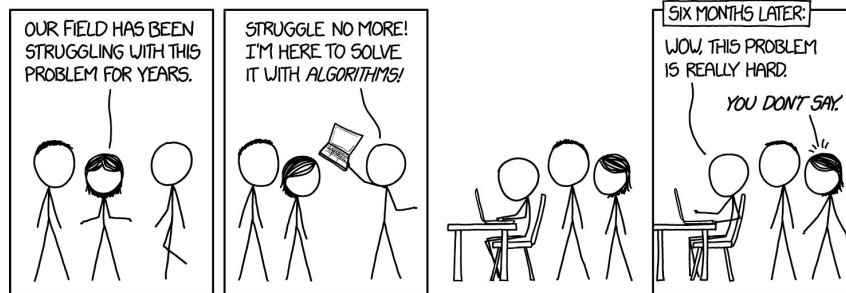
Reading primary research papers

Reading, Retention, and Reuse

- Make reading papers a habit.
- Critically analyze what you read/hear. Don't be swayed by high-profile papers, media hype, or current dogma. Don't be swayed by the narrative.
- Use a reference manager (e.g. Zotero), put *everything* you read into it. Add add notes about specific take-homes. Use tags to group papers by subfield/method/data.
- Create and maintain a single source of all the technical terms and vocabulary for your project.
- Create and maintain a single source (R/Jupyter Notebook) with notes/text-excerpts/figures from all papers & reading materials.
- Contextualize what you read in relation to everything else you know / have read. Specifically consider limitations. Analyze information in terms of you and your project.

Choosing a good computational biology problem

xkcd.com/1831



Take an existing study

New

Area / system

Problem / question

Algorithm / technique

I Want

Insights / improvement / clarity / efficiency / usability

Interesting example



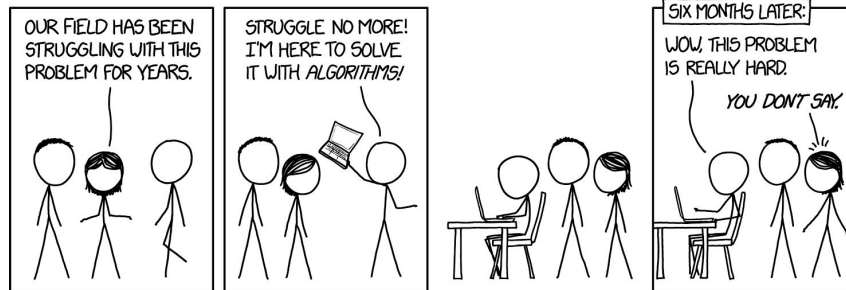
Generalized problem



Large-scale solution/insight

Choosing a good computational biology problem

xkcd.com/1831



Explore and prototype early to fail fast and learn

- Exploration + prototyping: critical for determining if the problem is well-defined & tractable.
- Perform preliminary analysis with simple baselines, sample datasets, and toy examples.
- Don't speculate or make assumptions. Instead, implement something and check them.
- The value lies not in the code/plots you produce, but in the lessons you learn.

Resources @ MSU

Institute for Cyber-Enabled Research

- High-Performance Computing Cluster: wiki.hpcc.msu.edu
- Training resources: www.icer.msu.edu/education-events/training-resources
- Seminars and workshops: www.icer.msu.edu/upcoming-workshops →
- Regular open office hours.

R-Ladies East Lansing

- >500 members from the larger MSU community
- <https://rladies-eastlansing.github.io/>

JAN
26

NVidia Webinar: First Deep Neural Network

This workshop is an introduction to Deep Learning using Tensorflow and Keras.

FEB
02

ICER Webinar: Introduction to Linux

Learn to navigate the UNIX file system and write a basic shell script as a prerequisite for submitting computational jobs on the HPCC.

FEB
04

ICER Webinar: Introduction to HPCC

This is a hands-on introductory workshop on using MSU's High Performance Computing Center (HPCC).

FEB
18

XSEDE Webinar: Performance Tuning and Single Processor Optimization

This presentation is targeted at attendees who both do their own code development and need their calculations to finish as quickly as possible.

FEB
23

NVidia Webinar: Convolution Neural Network Models

This workshop is an introduction to deep learning using convolution neural networks . It includes a Jupyter notebook using Tensorflow and Keras.

MAR
30

NVidia Webinar: Hyperparameter Optimization


This workshop illustrates how to search the set of training and model parameters to find the best possible model for a given training set. Simple grid search is used with seven different examples to illustrate how you can integrate Keras and Scikit-learn.

Getting help

- **Linux** | rik.smith-unna.com/command_line_bootcamp, commandline.guide, & swcarpentry.github.io/shell-novice
- **Python** | Introduction: learnpythonthehardway.org/book & developers.google.com/edu/python | Data analysis: jakevdp.github.io/WhirlwindTourOfPython | Visualization: www.r-graph-gallery.com
- **R** | Introduction: swcarpentry.github.io/r-novice-inflammation & swirlstats.com ('R Programming' & 'Data Analysis') | Data analysis: r4ds.had.co.nz | Visualization: python-graph-gallery.com
- **Git & GitHub** | swcarpentry.github.io/git-novice/, speakerdeck.com/alicebartlett/git-for-humans, & rogerdudler.github.io/git-guide/
- **Probability and Statistics** | Nature Collection (Statistics for Biologists | Practical Guides | Points of Significance): www.nature.com/collections/qghhqm
- **Genetics and Molecular Biology** | learn.genetics.utah.edu/ & www.genomicseducation.hee.nhs.uk



Getting help

 ... so many excellent blog posts!

 **stackoverflow**

 **Biostars**
— BIOINFORMATICS EXPLAINED —

Several video lessons/courses on YouTube

No shame!

StackOverflow Importer

Do you ever feel like all you're doing is copy/pasting from Stack Overflow?

Let's take it one step further.

from stackoverflow import quick_sort will go through the search results of [python] quick sort looking for the largest code block that doesn't syntax error in the highest voted answer from the highest voted question and return it as a module. If that answer doesn't have any valid python code, it checks the next highest voted answer for code blocks.

```
>>> from stackoverflow import quick_sort, split_into_chunks

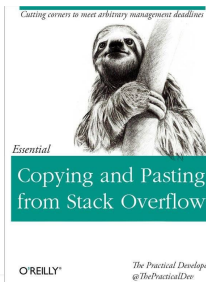
>>> print(quick_sort.sort([1, 3, 2, 5, 4]))
[1, 2, 3, 4, 5]

>>> print(list(split_into_chunks.chunk("very good chunk func")))
['very ', 'good ', 'chunk', ' func']

>>> print("I wonder who made split_into_chunks", split_into_chunks.__author__)
I wonder who made split_into_chunks https://stackoverflow.com/a/35107113

>>> print("but what's the license? Can I really use this?", quick_sort.__license__)
but what's the license? Can I really use this? CC BY-SA 3.0

>>> assert("nice, attribution!")
```



Getting help – Additional reading

- Checkout all the references cited in the slides.
- So you want to be a computational biologist? <https://www.nature.com/articles/nbt.2740>
- What Is the Key Best Practice for Collaborating with a Computational Biologist?
[https://www.cell.com/cell-systems/fulltext/S2405-4712\(16\)30223-X](https://www.cell.com/cell-systems/fulltext/S2405-4712(16)30223-X)
- A Quick Guide for Developing Effective Bioinformatics Programming Skills
<http://dx.plos.org/10.1371/journal.pcbi.1000589>
- Ten Simple Rules for Effective Computational Research
<http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1003506>
- Good Enough Practices in Scientific Computing <http://arxiv.org/abs/1609.00037>
- Ten simple rules for documenting scientific software
<https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1006561>

Getting help – Additional reading

- Fantastic resources on Reproducible code, Data management, Getting published, and Peer review
<http://www.britishecologicalsociety.org/publications/guides-to/>
- A Quick Guide to Organizing Computational Biology Projects
<https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1000424>
- A Quick Introduction to Version Control with Git and GitHub
<http://dx.plos.org/10.1371/journal.pcbi.1004668>
- Ten Simple Rules for Taking Advantage of Git and GitHub
<http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004947>

Teaching philosophy

- I think of each class I teach as a **collaborative learning community** where we can all work together to enhance each other's learning.
- To foster this community, my goal is to make sure that our class is a **space** where all of you can **join in, breathe, be seen & heard, be curious, and openly engage with the ideas**.
- **You absolutely belong here and you will be valued and respected.** Your unique background, training, and life experiences are your strength, and everyone including me will benefit from listening to you and learning from you.
- I have designed and am teaching this class to also maximize **my learning, fill gaps in my knowledge**, and find **better ways of discussing** each of the many **complex ideas**.