

Human Genetics 551: Computational Genomics Winter 2022

Course description

This course provides an introduction to the theory and practice of genomic data analysis including DNA sequence mapping, variant calling, gene expression analysis, and functional genomics. This course will introduce basic programming concepts using Python. A final course project will require applying standard genomics tools to existing data sets. This course is appropriate for students with no background in computer programming.

Students are required to use their own computers in this course. Students should have the ability to install software on their computer. We will also make heavy use Great Lakes, the campus wide scientific computing cluster.

Course Objectives

1. Learn the fundamentals of Python programming for manipulating genomics data
2. Develop an understanding of the computational principles underlying common genomics analyses
3. Apply computational tools to address important questions in genomics

Suggested Text Books and Other Resources

These books are not required and only listed as a reference. There are many good online resources for learning computer programming through Python. We will provide links to appropriate chapters.

Learning Python by Mark Lutz (O'Reilly publishing). We will focus on Python v 3.x (not 2.x).

Bioinformatics Data Skills by Vince Buffalo

Bioinformatics Programming Using Python by Mitchell Model

Introduction to Unix:

<http://www.ee.surrey.ac.uk/Teaching/Unix/index.html>

https://github.com/blahah/command_line_bootcamp

How to think like a computer scientist: <http://openbookproject.net/thinkcs/python/english3e/>

Python documentation: <https://docs.python.org/3/tutorial/>

Online resources with programming problems:

<http://rosalind.info/problems/locations/>

<http://opentechschoo.github.io/python-data-intro/>

There are many online forums with discussion about common programming problems and errors, as well as documentation on specific functions. A commonly used reference web page is <https://stackoverflow.com/>
Additional material including papers and web-resources will be made available using Canvas.

Start Date: Thursday, January 6, 2021

Time: 10:00am-11:20am (Tuesdays and Thursdays)

An optional practicum will also be scheduled based on student availability.

Location: TAUB LIB 6000 except for Tuesday April 5 and Thursday April 7, when course will be held in FSSC 2710 (below the 3rd floor of Med Sci II). Will also make use of a Slack channel for questions and discussion. Slack has several advantages including the ability to post screen shots or images of your code and any error messages.

Zoom info:

<https://umich.zoom.us/j/92993492671?pwd=VEpoQ0hGM0pSZG1CcThMeEFac1NiUT09>

Meeting ID: 929 9349 2671

Passcode: 039958

Note that the course is designed for interactive problem solving and that audio may not always come clearly over zoom.

Credits: 3

Instructors: Jeff Kidd, Ph.D. Jacob Kitzman, Ph.D.

Office: 3726 MS II 4811 MS II

Phone: 734-764-6922 734-764-9587

Email: jmkidd@umich.edu kitzmanj@umich.edu

Conflict of Interest: The instructors, Jeff Kidd and Jacob Kitzman, have no industry relationships.

GSI: Anthony Nguyen
antnguye@umich.edu

Grading

Course grades will be based on programming/analysis assignments (65%), a final course project and presentation (25%) and active and thoughtful participation during lectures and discussions, including in-class short answers to questions (10%). Dates for assignments are subject to change – consult Canvas and other announcements for up-to-date information. This course is planned for in person instruction in the Winter 2022 term and features time for inclass discussion and problem solving. The health of our community is of primary concern. Please do not attend class if you are unwell or are under investigation for symptoms based on the UM Covid screening policies. Please contact the instructors if you are unable to attend in person and options for remote participation (Zoom) may be activated.

Lecture Schedule (Note assignment dates and lecture topics are subject to change -- refer to Canvas for up-to-date information):

1. 1/6 (Th) Introduction to Computational Genomics
Motivation for learning computation, programs vs data, manipulating text files, file transfer, and using unix.

Reading:

Sean Eddy: High Throughput Sequencing for Neuroscience
<http://cryptogenomicon.org/2014/11/01/high-throughput-sequencing-for-neuroscience/>

Noble, WS *Plos Comp Biol* 2009
<http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1000424>

Introduction to UNIX commands
<https://swcarpentry.github.io/shell-novice/01-intro/>
<https://swcarpentry.github.io/shell-novice/02-filedir/>
<https://swcarpentry.github.io/shell-novice/03-create/>
<https://swcarpentry.github.io/shell-novice/04-pipefilter/>

Additional Unix tutorial (with built-in interface):
http://rik.smith-unna.com/command_line_bootcamp

Unix quick reference sheet:

<https://www.rain.org/~mkummel/unix.html>

2. 1/11 (Tu)

Intro to Unix and Intro to Python

Working with Unix, working with iPython notebooks, visualization, configuration, interactive vs saved-file computing. Variables, objects, concept of immutability. Numbers and strings

Reading:

How to Think Like a Computer Scientist: Learning with Python.

Chapter 1: The way of the program

http://openbookproject.net/thinkcs/python/english3e/way_of_the_program.html

One of many Unix tutorials:

<http://matt.might.net/articles/basic-unix/>

3. 1/13 (Th)

Python I

Variables, objects, concept of immutability. Numbers and strings

Reading:

Chapter 2: Variables, expressions, and statements.

http://openbookproject.net/thinkcs/python/english3e/variables_expressions_statements.html

Assignment:

Python programming assignment #1

4. 1/18 (Tu)

Computational Topic: Intro to Python II

Flow control: if, elif, else, loops.

Data structures: lists and tuples, and list comprehensions.

Mutable vs immutable variables

Reading:

How to Think Like a Computer Scientist: Learning with Python.

Chapter 4: Conditionals

Chapter 6: Iteration

Chapter 7: Strings

Chapter 9: Lists

<http://openbookproject.net/thinkcs/python/english3e/>

Optionally also see:

Python Tutorial sections 4.1-4.5, 5.1

<https://docs.python.org/3/tutorial/>

Assignment: Python exercise

5. 1/20 (Th)

Genomics Topic: Storing genomics coordinate data

coordinate intervals, 0-based vs 1-based, UCSC-bed format, bedtools

Computational Topic: Running programs and manipulating output in Unix

Reading:

Reminder -- Introduction to UNIX commands

<https://swcarpentry.github.io/shell-novice/01-intro/>
<https://swcarpentry.github.io/shell-novice/02-filedir/>
<https://swcarpentry.github.io/shell-novice/03-create/>
<https://swcarpentry.github.io/shell-novice/04-pipeline/>

Additional Unix tutorial (with built-in interface):

http://rik.smith-unna.com/command_line_bootcamp

Unix quick reference sheet:

<https://www.rain.org/~mkummel/unix.html>

Portions of Bedtools PDF manual (particularly Chapter 5, posted on Canvas)

Assignment:

Unix and bedtools exercise

6. 1/25 (Tu)Genomics Topic: *-seq

Assays for variation and function using sequencing as a read-out

Computational Topic: Intro to Python III

loops, iterables, methods, reading and writing files

Reading:

Shendure and Ji, *Nat Biotech*, 2008

<http://www.nature.com/nbt/journal/v26/n10/pdf/nbt1486.pdf>

Shendure and Lieberman Aiden, *Nat Biotech*, 2012

<http://www.nature.com/nbt/journal/v30/n11/pdf/nbt.2421.pdf>

Python file I/O

http://www.tutorialspoint.com/python/python_files_io.htm

How to Think Like a Computer Scientist: Learning with Python.

Chapter 13: Files

Assignment: Python programming assignment - file and string manipulation

7. 1/27 (Th)Genomics Topic: Illumina Fastq Files

Sequence representation and quality scores, storing of paired-end reads

Computational Topic: Intro to Python IV

Functions, classes, dictionaries and sets

Reading:

FASTQ Format (Wikipedia)

https://en.wikipedia.org/wiki/FASTQ_format

Python:

How to Think Like a Computer Scientist: Learning with Python.

Chapter 3: Functions

Chapter 5: Fruitful functions

Chapter 12: Dictionaries

See also dictionaries documentation

<https://docs.python.org/3/tutorial/datastructures.html#dictionaries>

Assignment:

fastq file analysis exercise

8. 2/1 (Tu)

Genomics Topic: Sequence alignment and Dynamic Programming

Local vs global alignments, Burrows-Wheeler transform

Computational Topic: Intro to Python V

Basic python file I/O with tabular data

Reading in a matrix to a dictionary

Reading:

Eddy, *Nat Biotech*, 2014

<http://www.nature.com/nbt/journal/v22/n7/full/nbt0704-909.html>

Assignment:

No assignment

9. 2/3 (Tu)

Genomics Topic: Significance of sequence matches

Dynamic programming, continued. Local vs global alignments.

Computational Topic: Intro to Python VI

Basic python file I/O with tabular data

Reading in a matrix to a dictionary

Assignment

Dynamic programming and sequence alignment (by hand)

10. 2/8 (Tu)

Genomics Topic: Assessing Significance

Basic parametric and nonparametric statistical tests

Computational Topic: Statistics in Python I

Array manipulation and numeric computing with numpy/scipy. Random number generation for simulations, and statistical testing in python.

Reading:

Points of Significance series from *Nature Methods*:

Importance of being uncertain

<http://www.nature.com/nmeth/journal/v10/n9/full/nmeth.2613.html>

and

Non-parametric tests

<http://www.nature.com/nmeth/journal/v11/n5/full/nmeth.2937.html>

The rest of this excellent series is at:

<http://www.nature.com/collections/qghhqm/pointsofsignificance>

More lengthy: *A biologist's guide to statistical thinking and analysis*
<http://www.ncbi.nlm.nih.gov/books/NBK153593/>

- 11. 2/10 (Th)** Computational Topic I: Statistics in Python (continued)
- Computational Topic II: Python data visualization I
 Matplotlib – basic plot construction
- Assignment:
 Python statistics and plotting assignment
- 12. 2/15 (Tu)** Data Visualization
 Data Visualization Exercises with GTEX
 Description of basic practices for job submission the Great Lakes Cluster
- Assignment:
 Cluster computing exercise
- 13. 2/17 (Th)** Genomics Topic: The multiple testing problem
 p-values, False discovery rate, q-values.
- Reading:
 Storey and Tibshirani, *PNAS*, 2003
<http://www.pnas.org/content/100/16/9440.full>
 Online plotting reference
- 14. 2/22 (Tu)** Genomics Topic: Sequence Analysis I
 Properties of Illumina data, introduction to mapping reads and the BAM format
- Computational Topic: BWA and samtools
 Manipulating BAM files using samtools and pysam
- Reading:
 Li et al *Bioinformatics*, 2009
<http://bioinformatics.oxfordjournals.org/content/25/16/2078.long>
- SAM/BAM specification: <http://samtools.github.io/hts-specs/SAMv1.pdf>
- 15. 2/24 (Tu)** Genomics Topic: Sequence Analysis II
 Theory and implementation of short read alignment, including Burrows-Wheeler read mapping.
- Computational Topic: Picard tools
 Read mapping with bwa on Great Lakes
- Reading:

Trapnell and Salzberg, *Nature Biotech*, 2009
<http://www.nature.com/nbt/journal/v27/n5/abs/nbt0509-455.html>

March 1 and March 3 – No class

- 16. 3/8 (Tu)** Genomics Topic: Sequence Analysis III (Final Project Groups Due)
 Variant calling, statistical models for joint vs individual calling. Considering different scenarios, somatic vs germline, selecting variants, hard filters vs machine learning approaches, VCF format
- Computational Topic: GATK and samtools
 Variant calling models
- Reading:
 Li, *Bioinformatics*, 2011
<http://bioinformatics.oxfordjournals.org/content/27/21/2987.long>
- DePristo et al., *Nature Genetics*, 2011
<http://www.nature.com/ng/journal/v43/n5/full/ng.806.html>
- Danecek et al. *Bioinformatics*, 2011
<http://bioinformatics.oxfordjournals.org/content/27/15/2156>
- VCF format specification:
<https://samtools.github.io/hts-specs/VCFv4.2.pdf>
- 17. 3/10 (Tu)** Genomics Topic: RNA-seq (I)
 Experimental strategies for expression profiling using sequencing. Algorithmic aspects of RNA read mapping, isoform detection, and quantification at the gene, isoform, and exon levels. Normalization.
- Computational Topic: RNA-seq read mapping
 Alignment vs alignment-free methods.
- Reading:
 Oshlack et al., *Genome Biol*, 2010
<https://genomebiology.biomedcentral.com/articles/10.1186/gb-2010-11-12-220>
- Conesa et al., *Genome Biol*, 2016
<https://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-0881-8>
- 18. 3/15 (Tu)** Genomics Topic: RNA-Seq II (Final Project Ideas Due)
 Measures of abundance (FPKM, TPM). Expression quantitative trait locus (QTL) statistical framework for studies of genetic variants' impact upon expression.

Computational Topic: Bowtie, STAR, and alignment-free methods

Reading:

Lior Pachter, “Estimating number of transcripts from RNA-Seq measurements (and why I believe in paywall)”

<https://liorpachter.wordpress.com/2014/04/30/estimating-number-of-transcripts-from-rna-seq-measurements-and-why-i-believe-in-paywall/>

Trapnell et al. *Nature Biotech*, 2013

<http://www.nature.com/nbt/journal/v31/n1/full/nbt.2450.html>

Assignment:

eQTL analysis exercise

19. 3/17 (Th)

Python Data Visualization II - Advanced Topics

Pandas and Advanced Plotting
eQTL exercise

Reading:

To be determined

20. 3/22 (Tu)

Clustering (Ryan Mills, Ph.D, guest-lecture)

Algorithms and applications for grouping data based on similarity

21. 3/24 (Th)

Genomics Topic: Functional Genomics

Chromatin profiling data, chip-seq, Peak calling, chromatin confirmation assays
HMMs, Segmentation, Machine Learning, and Statistical Inference

Reading:

To be determined

22. 3/29 (Tu)

Version Control, gitHub, and Reproducible Analyses

Reading:

To be determined

Assignment:

Work on final projects

23. 3/31 (Th)

TBD

24. 4/5 (Tu)

TBD

25. 4/7 (Th)

TBD

26. 4/12 (Tu)

Final Project Lab

27. 4/14 (Th)

Final Project Presentations

28. 4/19 (Tu)

Final Project Presentations