Syllabus
AMNH-RGGS Comparative Genomics 2 — Informatics
Fall 2024

Course instructors
Robert DeSalle (desalle@amnh.org)
Jose Barba (jbarba@amnh.org)
Jessica Goodheart (jgoodheart@amnh.org)
Dean Bobo (dbobo@amnh.org)

Teaching assistant
Meghan Forcellati (mforcellati1@amnh.org)

Course logistics
Meets Thursdays from 2:00 to 4:00 PM in the ICG conference room.
Meets from September 5 to December 12, 2024 (15 sessions).
Office hours are available by appointment. Please email us to schedule a time.

Course credits
2 credits.

Course description
This project-oriented course introduces students to the computational tools necessary for interpreting modern molecular sequencing data. It covers whole genome and transcriptome assembly, comparative analyses using both de novo and reference-based methods, and phylogenomic inference. Designed for non-model systems, the course emphasizes documentation, reproducibility, and proficiency with computational tools. Students will address experimental questions starting from raw data, aiming to lay a foundation for a publishable final product. Students are encouraged to use their own data for assignments and a course project. Those without personal data should select an appropriate dataset to work with. Evaluation will be based on student participation, performance on quizzes and homework assignments, in-class presentations, and a final project. Students will receive individual guidance on applying methods relevant to their research and data. This support will include using their own datasets for tutorials and homework assignments, as well as consultations with instructors to discuss their data and appropriate bioinformatic methods.

Learning objectives
- Learn rigorous methods for selecting, conducting, and documenting computational approaches in genomics-based science.
- Increase comfort and competence with core computational tools, mastering essentials for a variety of informatics and computational methods.
- Acquire broad knowledge of both classic and innovative concepts and tools that form the foundation of genome science.
- Develop professional oral and written communication skills.

Course requirements
A computer capable of working in UNIX environments.
An account on the Huxley HPC, contact Sajesh Singh (ssingh@amnh.org).
Access to journals for research based on primary literature.

Grading
Evaluation will be based on class attendance and participation, completion of homework assignments, performance on quizzes, three in-class presentations, and a final project.

Paper presentations
Each student will choose an -omics paper that they find innovative, exciting, relevant to their work, or particularly interesting. On October 10, they will deliver a 10-minute presentation providing a concise overview of the research question addressed and a thorough explanation of the computational methods employed.

Mid-term project
On November 14, each student will deliver a 10-15-minute presentation on a cutting-edge genomics approach and lead a classroom discussion on the topic. Each student will present on a different topic. The objective is for students to gain insight into advanced methods that, while not directly applicable to their PhD work, are valuable for a comprehensive understanding of the field. The presentation should provide an overview of the chosen approach, including its fundamental techniques, associated costs, and requirements. It should also address the types of research questions the approach can effectively tackle, as well as its current limitations in terms of application or accessibility. For the discussion segment, the presenting student should prepare at least one question for the class to explore. Additionally, each student is required to submit an annotated bibliography of 3-10 relevant publications by the beginning of class on November 14.

Potential topics for mid-term project presentations
1. Long read sequencing
2. Single molecule sequencing
3. Single cell sequencing
4. Direct sequencing of RNA
5. Contact mapping
6. Epigenomics
7. Pan-genomics
8. Metagenomics
9. Metabolomics
10. Spatial transcriptomics
11. New -omics sequencing platforms
12. Gene/genome editing
13. Genomics and artificial intelligence/machine learning

Final project
This course is project-based, allowing students to develop and refine skills essential for analyzing data from their own PhD research. Whenever possible, the final project and homework assignments should incorporate the personal data of students to enhance relevance and practical application. However, if such data are not available or are insufficient, the instructors will guide the students in selecting an appropriate alternative dataset. The main goal of the final project is to provide students with experience in developing well-formulated research questions that can be addressed with -omics data, appropriate computational methods to address these questions, and fully documented, reproducible computational pipelines. Working with instructors, students will craft a research question, select the methods to use, and create a detailed tutorial for the necessary computational steps. A short description (~350 words) of the proposed research question should be submitted by November 14.

The final project will consist of a detailed tutorial, which will be shared on the student's GitHub account. Throughout the course, students will work incrementally to complete the project, adding and testing new steps to the pipeline as they are introduced in class. Each student will deliver a 10-minute presentation on their final project. The submitted GitHub tutorial must meet the following requirements:
1) An "about" section explaining the research goal, why the specific approach presented was chosen, and any alternatives that should be considered. If alternative approaches were tested and rejected, this should be described.
2) For every step of the analysis, provide a description of the tool being used, the available options, any caveats to using the tool, and the exact command used in the project. Even trivial steps like text/data manipulations should be documented.
3) Where relevant, elaborate on any challenges encountered, whether trivial or significant.
4) A brief bibliography of citations for the computational tools used.

5) There is no length requirement for the resulting document, but it must be detailed enough for anyone to fully reproduce your research using the tutorial. It must clearly describe all necessary steps.

Statement on academic integrity
Each student is responsible for observing the traditional standards of scholarly discourse, scientific research, and academic honesty. Plagiarism, cheating, and research fraud will not be tolerated. Students are expected to work individually unless specifically instructed to work in groups. The full Academic Integrity policy is detailed in the RGGS student handbook.

Course evaluations
You will be asked to complete two brief evaluative surveys to help us assess student learning. These surveys are required. Additionally, each student must complete an anonymous course evaluation at the end of the term. This evaluation serves as a valuable tool for faculty and administrators to enhance the student learning experience.

Session schedule

Session	Date	Instructor	Topic
1	09–05–2024	Jose Barba	– Course overview – Reproducibility and organization in computational biology – Introduction to Unix and Unix-like operating systems – Basic navigation in the terminal
2	09–12–2024	Jose Barba	– Command line computing basics 1 – Connecting to remote servers using the terminal and a package manager
3	09–19–2024	Jose Barba	– Command line computing basics 2 – Introduction to GitHub
4	09–26–2024	Jose Barba	– Working on remote servers: HPC architecture, using a scheduler, submitting jobs – Scripting – Version control with GitHub
5	10–03–2024	Jose Barba	– Introduction to R and Python for phylogenomics
6	10–10–2024	Jose Barba	– Quiz 1 – Paper presentations
7	10–17–2024	Dean Bobo	– Getting data from public repositories – Processing Illumina reads: QC, trimming, and filtering
8	10–24–2024	Meghan Forcellati	– Genome assembly from Illumina reads – Assembly assessment – Transcriptome assembly from Illumina reads – Transcriptome assessment
9	11–01–2024	Jessica Goodhart	– Structural annotation of genomes and transcriptomes – Repeat identification and masking – Homology-based functional annotation – The principles and practice of BLAST searches
10	11–07–2024	Robert DeSalle (TBC)	– Domain-based functional annotation – InterProScan and other protein databases – GO annotation
11	11–14–2024	Jose Barba	– Final project description due – Quiz 2 – Mid-term project presentations
12	11–21–2024	Jose Barba	– Phylogenomic inference from NGS data
13	11–28–2024	Robert DeSalle	– Detecting sequence variants by mapping to a reference genome – Detecting selection in sequence variants

14	12–05–2024	Jose Barba	– Molecular clock dating for phylogenomic data
15	12–12–2024	Jose Barba	– GitHub tutorial due – Final project presentations

A collection of links for useful tools, tutorials, and repositories

RGGS CG2 — Informatics

- Course GitHub repository: https://github.com/josebarbamontoya/rggs_comparative_genomics_2

Unix

- Unix basic commands: https://sandbox.bio/tutorials/terminal-basics/
- Unix shell: https://swcarpentry.github.io/shell-novice/01-intro.html
- Data exploration with awk: https://sandbox.bio/tutorials/awk-intro
- Loops: https://swcarpentry.github.io/shell-novice/05-loop.html

R

- Introduction to R: https://swirlstats.com
- Programming with R: http://swcarpentry.github.io/r-novice-inflammation/
- Data visualization in R using ggplot2: http://varianceexplained.org/RData/lessons/lesson2/

Python

- Introduction to Python: https://python.land/python-tutorial
- Programming with Python: https://swcarpentry.github.io/python-novice-inflammation/
- Data visualization in Python: https://python-graph-gallery.com

Genomic data processing and analysis

- Data Collection and Quality Control
  - FastQC: https://www.bioinformatics.babraham.ac.uk/projects/fastqc/
- Data Preprocessing
  - AdapterRemoval: https://adapterremoval.readthedocs.io/en/stable/
  - Trimmomatic: http://www.usadellab.org/cms/?page=trimmomatic
  - Cutadapt: https://cutadapt.readthedocs.io/en/stable/
  - SPAdes: https://cab.spbu.ru/software/spades/
  - BFC: https://github.com/lh3/bfc
- Genome Assembly
  - SPAdes: https://cab.spbu.ru/software/spades/
  - Canu: https://github.com/marbl/canu
  - BWA: http://bio-bwa.sourceforge.net/
  - Bowtie2: http://bowtie-bio.sourceforge.net/bowtie2/index.shtml
- Annotation
  - AUGUSTUS: http://bioinf.uni-greifswald.de/augustus/
  - GeneMark: http://exon.gatech.edu/GeneMark/
  - BLAST: https://blast.ncbi.nlm.nih.gov/Blast.cgi
  - InterProScan: https://www.ebi.ac.uk/interpro/search/sequence/
  - Pfam: https://pfam.xfam.org/
- Variant Calling
  - BWA: http://bio-bwa.sourceforge.net/
  - Bowtie2: http://bowtie-bio.sourceforge.net/bowtie2/index.shtml
  - GATK: https://gatk.broadinstitute.org/hc/en-us
  - FreeBayes: https://github.com/freebayes/freebayes
  - SAMtools: http://www.htslib.org/
- Comparative Genomics
  - MUSCLE: https://www.drive5.com/muscle/
  - MAFFT: https://mafft.cbrc.jp/alignment/software/
  - RAxML: https://cme.h-its.org/exelixis/web/software/raxml/
  - IQ-TREE: http://www.iqtree.org
  - PAML: http://abacus.gene.ucl.ac.uk/software/paml.html
- Functional Genomics
  - HISAT2: https://daehwankimlab.github.io/hisat2/

    - StringTie: http://ccb.jhu.edu/software/stringtie/
    - DESeq2: https://bioconductor.org/packages/release/bioc/html/DESeq2.html
    - Bismark: https://www.bioinformatics.babraham.ac.uk/projects/bismark/
    - MACS: https://github.com/macs3-project/MACS
- Data Visualization
    - UCSC Genome Browser: https://genome.ucsc.edu/
    - IGV (Integrative Genomics Viewer): http://software.broadinstitute.org/software/igv/
    - ggplot2 (R): https://ggplot2.tidyverse.org/
    - Matplotlib (Python): https://matplotlib.org/
- Data Storage and Management
    - NCBI: https://www.ncbi.nlm.nih.gov/
    - EMBL-EBI: https://www.ebi.ac.uk/