From Zero to Hero: Six steps to learn computational biology

Ming Tommy Tang

3/25/2023

Computational biology is an interdisciplinary field that combines computer science, statistics, and biology to solve biological problems. If you are interested in learning computational biology on your own, these six steps will provide you with a structured approach to get started.

Step 1: Get Familiar with Unix

why Unix?

1. Many computational tools are written to run in the command line.
2. For high-performance computing clusters or cloud computing, there is no GUI for you to use.
3. Unix commands are powerful. You can combine different tools to wrangle data files.

Where to start:

- Unix Shell youtube videos Make sure you watch all ten episodes.
- Introduction to the Command Line for Genomics
- The Linux Command Line
- The Unix workbench

Step 2: Learn the Basics of Programming

To become a computational biologist, you need to learn how to program. Python and R are the most widely used language in computational biology, and it's a great place to start. You can find many online resources to learn python and R, such as Codecademy, Coursera, and edX. Once you have a basic understanding of Python and R, you can move on to more advanced topics such as data structures and algorithms.

Where to start

Statistics (R focused)

- R for data science by Garrett Grolemund and Hadley Wickham. Learning `tidyverse` is a great investment.
- Advanced R by Hadley Wickham.
- R packages by Hadley Wickham. If you want to transit from an R user to a developer, writing an R package will get you started.

- Efficient R programming
- Data analysis for the life science with R by Micheal Love and Rafael A. Irizarry. I took the course on edx for 3 times! learned a ton! You can buy a paper book at here.
- Computational Genomics with R by Altuna Akalin.
- Mordern statistics for modern biology by Susan Holmes and Wolfgang Huber.

Python programming

- Python for Biologists: A complete programming course for beginners
- Python for data analysis
- Data science from scratch

Step 3: Learn the Basics of Biology

To understand biological problems, you need to have a basic understanding of biology. You can start with a free online course, such as Khan Academy's Biology course. This course covers topics such as cells, genetics, and evolution. You can also read textbooks, such as "Molecular Biology of the Cell" by Bruce Alberts, to gain a deeper understanding of biology.

- Tales from the Genome A course by Udacity and 23andMe.
- Learn Genetics from University of Utah learning center.
- iBiology offers several different types of courses
- courses from khanacademy.org

Step 4: Learn Bioinformatics Tools

Bioinformatics tools are software programs that help you analyze biological data. You need to be aware what kind of tools are out there for your problem. Most of the time, someone has written a tool for your problem. Do not re-invent the wheel, but also do not trust the tool blindly. Read the documentation, check the source code, is it from a reputable lab?

Common tools for genomics (Just give you several examples):

- BWA, STAR for reads alignments
- bedtools for genomics interval analysis, MACS for ChIP-seq peak calling.
- variant calling: GATK, Dragene, Sentieon. QIAGEN also have a recent release of a WGS analysis pipeline run for 25 mins.
- Bioconductor. e.g, DESeq2 for differential gene expression. It has an R package for almost every genomics assay that you may work on.

Bioinformatics is a rapid evolving field. There are many tools published every week. How to stay the current of bioinformatics? I recommend you all to read Steven Turner's blog posts:

- How to Stay Current in Bioinformatics/Genomics 2012 Edition

- Staying Current in Bioinformatics & Genomics: 2017 Edition

I am very grateful for Steven's posts. After reading the 2012 post, I registered a twitter account and started my blog: Diving into Genetics and Genomics. Why on twitter? Because you can follow the latest papers and tools to stay the forefront of the field. Moreover, you can learn a lot from the tweeps and even make friends. Follow me on twitter.

I got a lot of information on twitter, but I did one step further: curating the materials:

- My RNA-seq analysis notes
- My ChIP-seq analysis notes
- My DNA-seq analysis notes
- My single-cell RNAseq analysis notes
- My DNA methylation analysis notes
- My single-cell ATACseq analysis notes
- My spatial transcriptome analysis notes
- My TCR and BCR sequencing analysis notes

Step 5: Practice, Practice, Practice

As with any skill, practice makes perfect. Start by working on small projects and gradually work your way up to more complex problems. There are many public datasets available for you to practice on, such as the Cancer Genome Atlas (TCGA) and the Genotype-Tissue Expression (GTEx) project.

You can use these datasets to practice your programming, data analysis, and visualization skills. You can also participate in bioinformatics challenges, such as the Rosalind problems to gain experience in solving real-world problems. One other thing I love to do is to reproduce figures in published papers.

Where to get data

There are so many publicly available data. I am proud to consider myself as a research parasite to mine the public data.

Two main repositories to get the published data:

- Gene Expression Omnibus (GEO) GEO hosts files in its own SRA format, you need to use fasterq-dump or parallel version https://github.com/rvalieris/parallel-fastq-dump to convert the sra files to the fastq files.
- European Nucleotide Archive (ENA)

Step 6: Join a Community

Joining a community of computational biologists can help you learn from others and stay up-to-date with the latest developments in the field. You can join online communities, such as the twitter, such as BioStars, Seqanswers, Reddit's Bioinformatics community, and the Computational Biology Stack Exchange. You can ask questions, share your work, and learn from others in the community and you can also attend conferences, such as the ISMB conference and Bioconductor conference, to learn from experts in the field and present your research.

Conclusion

By following these six steps, you will be well on your way to becoming a computational biologist. Don't be afraid to ask questions and seek help when you need it. Remember, the key to success is perseverance and hard work. No doubt, there will be difficulties. I was there and I fully understand you. However, with dedication and practice, you can become a proficient computational biologist and contribute to the advancement of biological research. More importantly, you can transform to the people you want to be:)

Computational biology is a rapidly growing field, and there are many exciting opportunities for those who are passionate about using data and technology to solve biological problems.

Happy Learning!

Bonus

Ten courses to get you started with bioinformatics/computational Biology

- HarvardX Biomedical Data Science Open Online Training by Rafa
- Applied Computational Genomics Course at UU: by Aaron Quinlan, the creator of `bedtools` and many other cool tools.
- Bioinformatics Algorithms. You can find the video classes on Coursera
- Applied Bioinformatics by Istvan Albert, the creator of biostars.
- Introduction to Bioinformatics and Computational Biology by Shirley Liu in Dana-Farber, Harvard. I am glad to contribute a little myself.
- Data carpentry Genomics workshops I am honored to serve as the curriculum committee chair
- Computational Genomics: Applied Comparative Genomics by Michael Schatz.
- Introduction to Computational Biology by Mike Love
- MIT Computational Biology: Genomes, Networks, Evolution, Health - Fall 2018 - 6.047/6.878/HST.507 by Manolis Kellis