Introduction to Ancient Metagenomics

2023-07-05

Table of contents

Introduction

Ancient metagenomics applies cutting-edge metagenomic methods to the degraded DNA content of archaeological and paleontological specimens. The rapidly growing field is currently uncovering a wealth of novel information for both human and natural history, from identifying the causes of devastating pandemics such as the Black Death, to revealing how past ecosystems changed in response to long-term climatic and anthropogenic change, to reconstructing the microbiomes of extinct human relatives. However, as the field grows, the techniques, methods, and workflows used to analyse such data are rapidly changing and improving.

In this book we will go through the main steps of ancient metagenomic bioinformatic workflows, familiarising students with the command line, demonstrating how to process next-generation-sequencing (NGS) data, and showing how to perform de novo metagenomic assembly. Focusing on host-associated ancient metagenomics, the book consists of a combination of theory and hands-on exercises, allowing readers to become familiar with the types of questions and data researchers work with.

By the end of the textbook, readers will have an understanding of how to effectively carry out the major bioinformatic components of an ancient metagenomic project in an open and transparent manner.

All material was originally developed for the SPAAM Summer School: Introduction to Ancient Metagenomics

Chapter 1

Introduction

This is a book created from markdown and executable code.

See (**knuth84?**) for additional discussion of literate programming.

```
1 + 1
```

```
[1] 2
```

Chapter 2

Authors

The creation of this text book was developed through a series of

Part I

Theory

Chapter 3

Introduction to NGS Sequencing

3.1 Introduction

In this section, I will introduce how we are able to convert DNA molecules to human readable sequences of A, C, T, and Gs, which we can subsequently can computationally analyse.

The field of Ancient DNA was revolutionised by the development of 'Next Generation Sequencing' (NGS), which relies on sequencing of millions of *short* fragments of DNA in parallel. The global leading DNA sequencing company is Illumina, and the technology used by Illumina is also most popular by palaeogeneticists. Therefore I will describe how the various technologies behind Illumina next-generation sequencing machines.

I will also describe some important differences in the way different models of Illumina sequences work, and how this can influence ancient DNA research. Finally I will introduce the structure of 'FASTQ' files, the most popular file format for representing the DNA sequence output of NGS sequencing machines.

3.2 Lecture

PDF version of these slides can be downloaded from here.

3.3 Readings

3.3.1 Reviews

(Schuster 2008)

(Shendure and Ji 2008)

(Slatko, Gardner, and Ausubel 2018)

(Dijk et al. 2014)

3.3.2 Sequencing Library Construction

(Kircher, Sawyer, and Meyer 2012)

(Meyer and Kircher 2010)

3.3.3 Errors and Considerations

(Ma et al. 2019)

(Sinha et al. 2017)

(Valk et al. 2019)

3.4 Questions to think about

- Why is Illumina sequencing technologies useful for aDNA?
- What problems can the 2-colour chemistry technology of NextSeq and NovaSeqs cause in downstream analysis?
- Why is 'Index-Hopping' a problem?
- What is good software to evaluate the quality of your sequencing runs?

Chapter 4

Introduction to Ancient DNA

Chapter 5

Introduction to Metagenomics

Chapter 6

Introduction to Microbial Genomics

Chapter 7

Introduction to Evolutionary Biology

Part II

Useful Skills

Chapter 8

Bare Bones Bash

Chapter 9

Introduction to R and the Tidyverse

Chapter 10

Introduction to Python and Pandas

Chapter 11

Introduction to Git(Hub)

11.1 Overview

As the size and complexity of metagenomic analyses continues to expand, effectively organizing and tracking changes to scripts, code, and even data, continues to be a critical part of ancient metagenomic analyses. Furthermore, this complexity is leading to ever more collaborative projects, with input from multiple researchers.

In this practical session, we will introduce 'Git', an extremely popular version control system used in bioinformatics and software development to store, track changes, and collaborate on scripts and code. We will also introduce, GitHub, a cloud-based service for Git repositories for sharing data and code, and where many bioinformatic tools are stored. We will learn how to access and navigate course materials stored on GitHub through the web interface as well as the command line, and we will create our own repositories to store and share the output of upcoming sessions.

11.1.0.1 Preparation

The conda environment `.yaml` file for this practical session can be downloaded from here: https://zenodo.org/record/6983120#.YxdEaOxBz0o. See instructions on page.

11.1.0.2 Introduction

In this walkthrough, we will introduce the version control system **Git** as well as **Github**, a remote hosting service for version controlled repositories. Git and Github are increasingly popular tools for tracking data, collaborating on research projects, and sharing data and code, and learning to use them will help in many aspects of your own research. For more information on the benefits of using version control systems, see the slides.

11.1.0.3 SSH setup

To begin, you will set up an SSH key to facilitate easier authentication when transferring data between local and remote repositories. In other words, follow this section of the tutorial so that you never have to type in your github password again! Begin by activating the conda environment for this section (see **Preparation** above).

```
conda activate git-eager
```

Next, generate your own ssh key, replacing the email below with your own address.

```
ssh-keygen -t ed25519 -C "your_email@example.com"
```

I recommend saving the file to the default location and skipping passphrase setup. To do this, simply press enter without typing anything.

You should now (hopefully!) have generated an ssh key. To check that it worked, run the following commands to list the files containing your public and private keys and check that the ssh program is running.

```
cd ~/.ssh/
ls id*
eval "$(ssh-agent -s)"
```

Now you need to give ssh your key to record:

```
ssh-add ~/.ssh/id_ed15519
```

Next, open your webbrowser and navigate to your github account. Go to settings -> SSH & GPG Keys -> New SSH Key. Give you key a title and paste the public key that you just generated on your local machine.

```
cat ~/.ssh/id_ed15519
```

Finally, press Add SSH key. To check that it worked, run the following command on your local machine. You should see a message telling you that you've successfully authenticated.

```
ssh -T git@github.com
```

For more information about setting up the SSH key, including instructions for different operating systems, check out github's documentation: https://docs.github.com/es/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent.

11.1.0.4 The only 6 commands you really need to know

Now that you have set up your own SSH key, we can begin working on some version controlled data! Navigate to your github homepage and create a new repository. You can choose any name for your new repo (including the default). Add a README file, then select Create Repository.

Note: For the remainder of the session, replace the name of my repository (vigilant-octo-journey) with your own repo name.

Change into the directory where you would like to work, and let's get started! First, we will learn to **clone** a remote repository onto your local machine. Navigate to your new repo, select the *Code* dropdown menu, select SSH, and copy the address as shown below.

Back at your command line, clone the repo as follows:

```
git clone git@github.com:meganemichel/vigilant-octo-journey.git
```

Next, let's **add** a new or modified file to our 'staging area' on our local machine.

```
cd vigilant-octo-journey
echo "test_file" > file_A.txt
echo "Just an example repo" >> README.md
git add file_A.txt
```

Now we can check what files have been locally changed, staged, etc. with **status**.

```
git status
```

You should see that `file_A.txt` is staged to be committed, but `README.md` is NOT. Try adding `README.md` and check the status again.

Now we need to package or save the changes into a **commit** with a message describing the changes we've made. Each commit comes with a unique hash ID and will be stored forever in git history.

```
git commit -m "Add example file"
```

Finally, let's **push** our local commit back to our remote repository.

```
git push
```

What if we want to download new commits from our remote to our local repository?

```
git pull
```

You should see that your repository is already up-to-date, since we have not made new changes to the remote repo. Let's try making a change to the remote repository's README file (as below). Then, back on the command line, pull the repository again.

11.1.0.5 Working collaboratively

Github facilitates simultaneous work by small teams through branching, which generates a copy of the main repository within the repository. This can be edited without breaking the 'master' version. First, back on github, make a new branch of your repository.

From the command line, you can create a new branch as follows:

```
git switch -c new_branch
```

To switch back to the main branch, use

```
git switch main
```

Note that you **must commit changes** for them to be saved to the desired branch!

11.1.0.6 Pull requests

A **Pull request** (aka PR) is used to propose changes to a branch from another branch. Others can comment and make suggestinos before your changes are merged into the main branch. For more information on creating a pull request, see github's documentation: https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/creating-a-pull-request.

11.1.1 Resources

- https://www.atlassian.com/git/tutorials
- https://ohshitgit.com/

11.1.2 Readings

- Chacon, Scott, and Ben Straub. 2022. Pro Git. Second Edition. The Expert's Voice. Apress.

11.1.3 Questions to think about

1. Why is using a version control software for tracking data and code important?
2. How can using Git(Hub) help me to collaborate on group projects?

Part III

data.qmd - introduction-to-ancientmetagenomedir.qmd

Chapter 12

Summary

In summary, this book has no content whatsoever.

```
1 + 1
```

```
[1] 2
```

Part IV

Appendicies

Chapter 13

Resources

13.1 Introduction to NGS Sequence

- https://www.youtube.com/watch?v=fCd6B5HRaZ8
- https://emea.illumina.com/content/dam/illumina-marketing/documents/products/illumina_sequencing_introduction.pdf

References

Dijk, Erwin L van, Hélène Auger, Yan Jaszczyszyn, and Claude Thermes. 2014. "Ten Years of Next-Generation Sequencing Technology." *Trends in Genetics* 30 (9): 418–26. https://doi.org/10.1016/j.tig.2014.07.001.

Kircher, Martin, Susanna Sawyer, and Matthias Meyer. 2012. "Double Indexing Overcomes Inaccuracies in Multiplex Sequencing on the Illumina Platform." *Nucleic Acids Research* 40 (1): e3. https://doi.org/10.1093/nar/gkr771.

Ma, Xiaotu, Ying Shao, Liqing Tian, Diane A Flasch, Heather L Mulder, Michael N Edmonson, Yu Liu, et al. 2019. "Analysis of Error Profiles in Deep Next-Generation Sequencing Data." *Genome Biology* 20 (1): 50. https://doi.org/10.1186/s13059-019-1659-6.

Meyer, Matthias, and Martin Kircher. 2010. "Illumina Sequencing Library Preparation for Highly Multiplexed Target Capture and Sequencing." *Cold Spring Harbor Protocols* 2010 (6): db.prot5448. https://doi.org/10.1101/pdb.prot5448.

Schuster, Stephan C. 2008. "Next-Generation Sequencing Transforms Today's Biology." *Nature Methods* 5 (1): 16–18. https://doi.org/10.1038/nmeth1156.

Shendure, Jay, and Hanlee Ji. 2008. "Next-Generation DNA Sequencing." *Nature Biotechnology* 26 (10): 1135–45. https://doi.org/10.1038/nbt1486.

Sinha, Rahul, Geoff Stanley, Gunsagar Singh Gulati, Camille Ezran, Kyle Joseph Travaglini, Eric Wei, Charles Kwok Fai Chan, et al. 2017. "Index Switching Causes 'Spreading-of-Signal' Among Multiplexed Samples in Illumina HiSeq 4000 DNA Sequencing." *bioRxiv*. https://doi.org/10.1101/125724.

Slatko, Barton E, Andrew F Gardner, and Frederick M Ausubel. 2018. "Overview of Next-Generation Sequencing Technologies." *Current Protocols in Molecular Biology / Edited by Frederick M. Ausubel ... [Et Al.]* 122 (1): e59. https://doi.org/10.1002/cpmb.59.

Valk, Tom van der, Francesco Vezzi, Mattias Ormestad, Love Dalén, and Katerina Guschanski. 2019. "Index Hopping on the Illumina HiseqX Platform and Its Consequences for Ancient DNA Studies." *Molecular Ecology Resources*, March. https://doi.org/10.1111/1755-0998.13009.

Chapter 14

Tools

Chapter 15

Acknowledgements

We would like to thank

15.1 Financial Support

WSS

WERNER SIEMENS-STIFTUNG

The content of this textbook was developed from the SPAAM Summer School: Introduction to Ancient Metagenomics summer school series, sponsored by the Werner Siemens-Stiftung (Grant: Paleobiotechnology, awarded to Pierre Stallforth, Hans-Knöll Institute, and Christina Warinner, Max Planck Institute for Evolutionary Anthropology)

15.2 Institutional Support

15.3 Infrastructural Support

de.NBI

GERMAN NETWORK FOR BIOINFORMATICS INFRASTRUCTURE

The practical sessions of the summers schools work was supported by the BMBF-funded de.NBI Cloud within the German Network for Bioinformatics Infrastructure (de.NBI) (031A532B, 031A533A, 031A533B, 031A534A, 031A535A, 031A537A, 031A537B, 031A537C, 031A537D, 031A538A). z

Chapter 16

Index

Index