Principles and Applications of Modern DNA Sequencing

EEEB GU4055

Session 2: Genome Structure

Today's topics

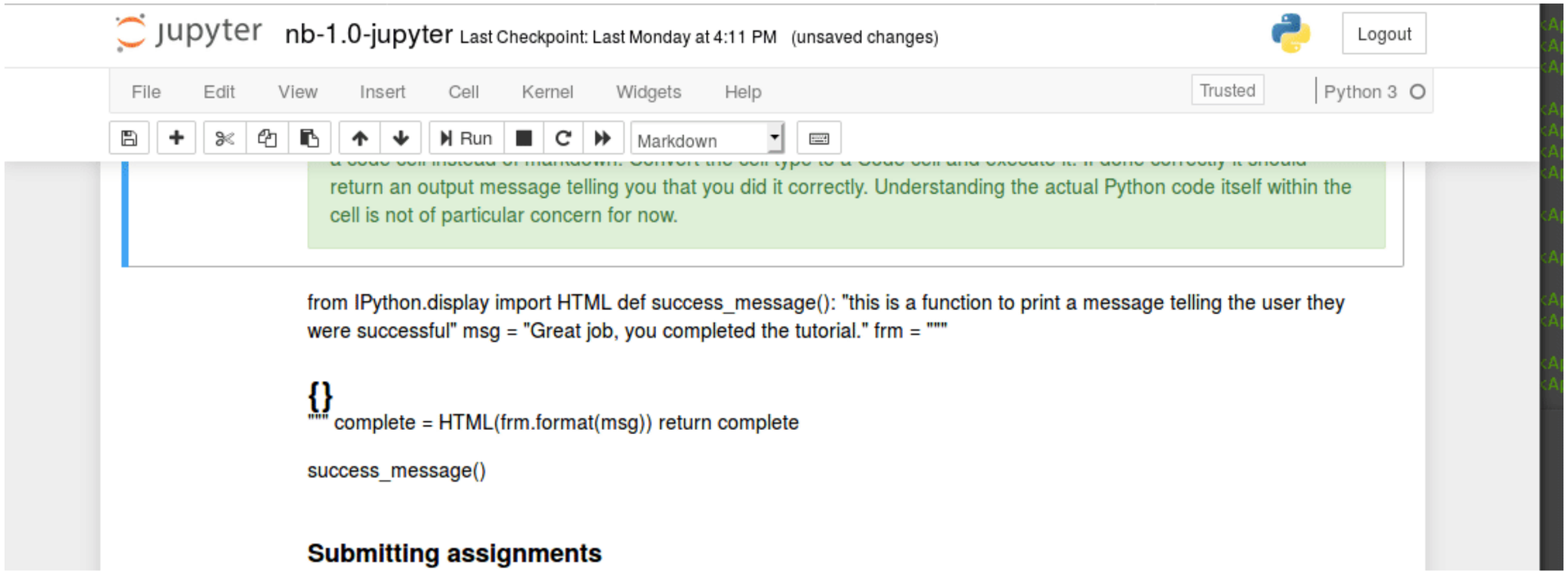
1. Review notebook assignments (bash)
2. Genome annotations (GFF table)
3. Assigned reading (history of genomics)
4. Introduction to Python

Clarification: readings are paired with notebooks

Your assignment for next session is to read a Python tutorial while completing notebooks that introduce Python coding with examples from genomics.

Notebook 1.0: Intro to jupyter

Executing code blocks, editing Markdown, saving notebooks. We covered this in class last time, but has anyone encountered any technical issues?



Interacting with a bash terminal

Lines starting with hash (#) are only comments.

```
# This is the general format of unix command line tools
$ program -option1 -option2 target
```

An example command line program:

```
# e.g., the 'pwd' program with no option or target prints your cur dir
$ pwd
```

```
/home/deren/
```

Interacting with a bash terminal

```
# The echo command prints text to the screen
$ echo "hello world"
```

```
hello world
```

```
# The -e option to echo renders special characters
$ echo -e "hello\tworld"
```

```
hello	world
```

Executing bash in jupyter

Jupyter notebooks can execute many different computer languages (sometimes requiring add-on installations). By default it supports both Python and bash. You can run a code cell in bash-mode by appending *%%bash* to the top.

```
%%bash
echo -e "hello\tworld"
```

```
hello     world
```

Errors and Exceptions

When an error is detected the Python interpreter will return a message to the cell output with a hint about the error. For ecample, if we tried to execute bash code in a Python-mode code cell it raises a SyntaxError:

```
# we forgot to add %%bash to the header of this cell
echo -e "hello\tworld"
```

```
File "ipython-input-458-239334a501c4", line 1
echo -e "hello\tworld"
                 ^
SyntaxError: invalid syntax
```

Notebook 1.1: bash and genomes

Index of ftp://ftp.ncbi.nlm.nih.gov/genomes/refseq/fungi/Saccharomyces_cerevisiae/latest_assembly_versions/GCF_000146045.2_R64/

Up to higher level directory

Name	Size	Last Modified	
File: GCF_000146045.2_R64_assembly_report.txt	3 KB	10/30/18	2:47:00 PM EDT
File: GCF_000146045.2_R64_assembly_stats.txt	16 KB	10/30/18	2:47:00 PM EDT
GCF_000146045.2_R64_assembly_structure		9/20/16	12:00:00 AM EDT
File: GCF_000146045.2_R64_cds_from_genomic.fna.gz	2946 KB	6/8/17	12:00:00 AM EDT
File: GCF_000146045.2_R64_feature_count.txt.gz	1 KB	12/24/17	12:00:00 AM EST
File: GCF_000146045.2_R64_feature_table.txt.gz	377 KB	12/24/17	12:00:00 AM EST
File: GCF_000146045.2_R64_genomic.fna.gz	3754 KB	4/9/18	12:00:00 AM EDT
File: GCF_000146045.2_R64_genomic.gbff.gz	8096 KB	4/9/18	12:00:00 AM EDT
File: GCF_000146045.2_R64_genomic.gff.gz	1424 KB	4/9/18	12:00:00 AM EDT
File: GCF_000146045.2_R64_protein.faa.gz	1800 KB	4/5/17	12:00:00 AM EDT
File: GCF_000146045.2_R64_protein.gpff.gz	3697 KB	4/9/18	12:00:00 AM EDT
File: GCF_000146045.2_R64_rm.out.gz	104 KB	6/16/16	12:00:00 AM EDT
File: GCF_000146045.2_R64_rm.run	1 KB	3/31/17	12:00:00 AM EDT
File: GCF_000146045.2_R64_rna.fna.gz	2712 KB	4/9/18	12:00:00 AM EDT
File: GCF_000146045.2_R64_rna.gbff.gz	7861 KB	4/9/18	12:00:00 AM EDT
File: GCF_000146045.2_R64_rna_from_genomic.fna.gz	2967 KB	5/1/17	12:00:00 AM EDT
File: GCF_000146045.2_R64_translated_cds.faa.gz	2071 KB	12/24/17	12:00:00 AM EST
README.txt		9/20/16	12:00:00 AM EDT
File: annotation_hashes.txt	1 KB	10/30/18	2:47:00 PM EDT
File: assembly_status.txt	1 KB	1/27/19	6:06:00 AM EST
File: md5checksums.txt	6 KB	10/30/18	2:47:00 PM EDT

Finding genome data online (NCBI example)

Published genomes are organized into a file system on NCBI where the compressed sequence data file, genome annotation file, and other data files are grouped into folders. You can right-click to get the URL of files to download with wget.

```
# create a new directory to store files in.
mkdir -p genomes/

# the URL link to the genome file, here stored to the variable 'url1'
url1="https://ftp.ncbi.nlm.nih.gov/genomes/refseq/viral/Pandoravirus_quercus/late

# run the wget program on the url with additional options
wget $url1 -q -O ./genomes/virus.fna.gz

# download GFF (genome feature file) file for Yeast assembly from URL
url2="https://ftp.ncbi.nlm.nih.gov/genomes/refseq/fungi/Saccharomyces_cerevisiae/
wget $url2 -q -O ./genomes/yeast.gff.gz
```

A reference genome (fasta file format)

```
>NC_001133.9 Saccharomyces cerevisiae S288C chromosome I, complete sequence
ccacaccacacccacacacccacacaccacaccacacaccacaccacacccacacacacacatCCTAACACTAC
ACAGCCCTAATCTAACCCTGGCCAACCTGTCTCTCAACTTACCCTCCATTACCCTGCCTCCACTCGTTACCCTG
TCAACCATACCACTCCGAACCACCATCCATCCCTCTACTTACTACCACTCACCCACCGTTACCCTCCAATTACC
CAACCCACTGCCACTTACCCTACCATTACCCTACCATCCACCATGACCTACTCACCATACTGTTCTTCTACCCA
TGAAACGCTAACAAATGATCGTAAATAACACACACGTGCTTACCCTACCACTTTATACCACCACCACATGCCAT
CCTCACTTGTATACTGATTTTACGTACGCACACGGATGCTACAGTATATACCATCTCAAACTTACCCTACTCTC
CACTTCACTCCATGGCCCATCTCTCACTGAATCAGTACCAAATGCACTCACATCATTATGCACGGCACTTGCCT
TCTATACCCTGTGCCATTTACCCATAACGCCCATCATTATCCACATTTTGATATCTATATCTCATTCGGCGGTc
attgtataaCTGCCCTTAATACATACGTTATACCACTTTTGCACCATATACTTACCACTCCATTTATATACACT
AATATTACAGAAAAATCCCCACAAAAATCacctaaacataaaaatattctacttttcaacaataataCATAAAC
GCTTGTGGTAGCAACACTATCATGGTATCACTAACGTAAAAGTTCCTCAATATTGCAATTTGCTTGAACGGATG
CAGAATATTTCGTACTTACACAGGCCATACATTAGAATAATATGTCACATCACTGTCGTAACACTCTTTATTCA
AATAATACGGTAGTGGCTCAAACTCATGCGGGTGCTATGATACAATTATATCTTATTTCCATTCCCATATGCTA
ATATCCTAAAAGCATAACTGATGCATCTTTAATCTTGTATGTGACACTACTCATACGAAGGGACTATATCTAGT
GATACTGTGATAGGTACGTTATTTAATAGGATCTATAACGAAATgtcaaataattttacgGTAATATAACTTAT
...
```

A genome annotation (GFF) tabular file

```
##gff-version 3
#!gff-spec-version 1.21
#!processor NCBI annotwriter
#!genome-build R64
#!genome-build-accession NCBI_Assembly:GCF_000146045.2
#!annotation-source SGD R64-2-1
##sequence-region NC_001133.9 1 230218
##species https://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi?id=559292
NC_001133.9 RefSeq  region  1   230218  .   +   .   ID=NC_001133.9:1..230218;Db
NC_001133.9 RefSeq  telomere    1   801 .   -   .   ID=id-NC_001133.9:1..801;Db
NC_001133.9 RefSeq  origin_of_replication   707 776 .   +   .   ID=id-NC_00113
NC_001133.9 RefSeq  gene    1807    2169    .   -   .   ID=gene-YAL068C;Dbxref=
NC_001133.9 RefSeq  mRNA    1807    2169    .   -   .   ID=rna-NM_001180043.1;
NC_001133.9 RefSeq  exon    1807    2169    .   -   .   ID=exon-NM_001180043.1-
...
```

Reading a (big) genome fasta file

```
# zcat decompresses and reads the whole file, pipe to head to show only top
$ zcat genomes/virus.fna.gz | head -n 10
```

```
>NC_037667.1 Pandoravirus quercus, complete genome
CCGGTACAGTGAGCGGTTCACGGCCTGGCCACGGTCGACGGAGTGCCGTGCGATGCCATCGGCGACGGCCG
CGCGGGCATTCGCACGTGCGACCACAGCCGTCAGTGGTACTGGCGGGACGAGGCCGTCGGGGTGACGGACG
ACCTGCTCGATGCCATCACACGATGCGCCGAGTACGCGCACGATACCATCAGGGCGCCGTTGGCGAGCAAA
GAGATTATGGAGTTCAGCGTCCGTTGCACCCGCCAGGCGGCGGCCGGAGGCGACGACGTCACGGACCCCAT
GGACGCGAGGCCAGGCGCACGTGGCGCGCCTATCGCATGCACGCGCGCGTGTTCAGCGCCATCGCGTTGCT
ACCGCTGAGCATGATGGCGACGGCGGGTCTGCCCTTCTATGACGTGCGCCGGTACGCGCTGGTGGCGGCCC
GCCGCGCCGAACGCGCGTCGAGCCTGCTCCCAACACGCGTGCGACCAGACACCCTTGCGCACGAGGTGATG
...
```

Reading a tabular genome feature (GFF) file

cut, grep, awk and other bash tools are fast and powerful methods for selecting columns or rows of data tables. We will soon learn to do this more easily in Python.

```
# read file | exclude lines start w/ # | get cols 1-5 | show first 10 lines
zcat genomes/yeast.gff.gz | grep -v "^#" | cut -f 1-5 | head -n 10
```

```
NC_001133.9 RefSeq  region  1   230218
NC_001133.9 RefSeq  telomere    1   801
NC_001133.9 RefSeq  origin_of_replication   707 776
NC_001133.9 RefSeq  gene    1807    2169
NC_001133.9 RefSeq  mRNA    1807    2169
NC_001133.9 RefSeq  exon    1807    2169
NC_001133.9 RefSeq  CDS 1807    2169
NC_001133.9 RefSeq  gene    2480    2707
NC_001133.9 RefSeq  mRNA    2480    2707
...
```

The GFF file format

We will revisit this file format in association with the next reading assignment; it introduces how genomic features are related (e.g., gene -> mRNA transcript -> exon -> CDS). For now, we are using it to practice reading and parsing a tab-delimited file.

General feature format (GFF/GTF)

There are several formats for storing identified genome features called GFF or GTF or GFF3 files. In this tabular formatted file feature names are mapped to coordinates of the genome in terms of the scaffold or chromosome names and their start:stop positions. Features are named according to a specific naming system (ontology) that **we will discuss more in the future**. But for now, take note that it is a hierarchical system (subunits nested within higher level units), as represented by the figure below. All of the elements below gene1 are parts that make up the unit we are calling gene 1, which includes messenger RNAs, exons, and coding sequences (exons - introns - UTRs).

```
gene1            ================================    ID=gene1
mRNA1            ================================    ID=mRNA1;Parent=gene1
five_prime_UTR   ==                                  Parent=mRNA1
CDS1               ==....=====...........==          Parent=mRNA1 (3 rows)
three_prime_UTR                          ======      Parent=mRNA1
mRNA2            ================================    ID=mRNA2;Parent=gene1
exon             ====                                Parent=mRNA2
CDS2               ==....................==          Parent=mRNA2 (2 rows)
exon                                     ========    Parent=mRNA2
```

The grep tool

grep is one of the most commonly used bash tools. It can be used like a filter on lines of text to include or exclude them based on their contents. In conjunction with the cut tool, you can select rows (lines) and columns of text in a file.

```
# pipe zcat output to grep
zcat genomes/yeast.fna.gz | grep ">"
```

```
>NC_001133.9 Saccharomyces cerevisiae S288C chromosome I, complete sequence
>NC_001134.8 Saccharomyces cerevisiae S288C chromosome II, complete sequence
>NC_001135.5 Saccharomyces cerevisiae S288C chromosome III, complete sequence
>NC_001136.10 Saccharomyces cerevisiae S288C chromosome IV, complete sequence
>NC_001137.3 Saccharomyces cerevisiae S288C chromosome V, complete sequence
>NC_001138.5 Saccharomyces cerevisiae S288C chromosome VI, complete sequence
>NC_001139.9 Saccharomyces cerevisiae S288C chromosome VII, complete sequence
...
```

grep and cut to parse tabular data

cut, grep, awk and other bash tools are fast and powerful methods for selecting columns or rows of data tables. We will soon learn to do this more easily in Python.

```
# read file | exclude lines start w/ # | get cols 1-5 | show first 10 lines
zcat genomes/yeast.gff.gz | grep -v "^#" | cut -f 1-5 | head -n 10
```

```
NC_001133.9 RefSeq  region  1   230218
NC_001133.9 RefSeq  telomere    1   801
NC_001133.9 RefSeq  origin_of_replication   707 776
NC_001133.9 RefSeq  gene    1807    2169
NC_001133.9 RefSeq  mRNA    1807    2169
NC_001133.9 RefSeq  exon    1807    2169
NC_001133.9 RefSeq  CDS 1807    2169
NC_001133.9 RefSeq  gene    2480    2707
NC_001133.9 RefSeq  mRNA    2480    2707
...
```

extracting and counting features

By combining these simple tools we can accomplish complex tasks, like asking 'how many genes does the yeast genome contain?' From studying the GFF format we know that the 3rd column contains feature types. Let's select all rows with the term 'gene' in column 3.

```
# read file | get 3rd field | grep -w to match word -c to count
zcat genomes/yeast.gff.gz | cut -f 3 | grep -wc "gene"
```

```
6427
```

how many genes in yeast

All | Images | News | Videos | Shopping | More | Settings | Tools

About 67,900,000 results (0.56 seconds)

The S. cerevisiae genome is composed of about 12,156,677 base pairs and 6,275 **genes**, compactly organized on 16 chromosomes. Only about 5,800 of these **genes** are believed to be functional. It is estimated at least 31% of **yeast genes** have homologs in the human genome.

en.wikipedia.org › wiki › Saccharomyces_cerevisiae

Saccharomyces cerevisiae - Wikipedia

About Featured Snippets | Feedback

Challenge from notebook 1.1

Return a tab-delimited table with positions of all telomeres in the Yeast genome.

Each line should have the following information: seqid, type, start, stop.

```
# read file | not lines start w/ # | fields 1,3,4,5 | only w/ 'telomere'
zcat genomes/yeast.gff.gz | \
    grep -v "^#" | \
    cut -f 1,3-5 | \
    grep -w 'telomere'
```

```
NC_001133.9 telomere    1   801
NC_001133.9 telomere    229411  230218
NC_001134.8 telomere    1   6608
NC_001134.8 telomere    812379  813184
NC_001135.5 telomere    1   1098
NC_001135.5 telomere    315783  316620
...
```

Public genome databases

You visited the NCBI FTP site to view published genome files and metadata. You were asked to select any genome in the refseq/ directory to find statistics in the 'assembly_stats.txt' file. Below is an example stats file for Corn (Zea Mays).

```
# Assembly Statistics Report
# Assembly name:  B73 RefGen_v4
# Description:    Zm-B73-REFERENCE-GRAMENE-4.0
# Organism name:  Zea mays (maize)
# Infraspecific name:  cultivar=B73
# Taxid:          4577
# BioSample:      SAMN04296295
# BioProject:     PRJNA10769
# Submitter:      maizesequence
# Date:           2017-02-07
# Assembly type:  haploid
# Release type:   major
# Assembly level: Chromosome
# Genome representation: full
```

POLL

Assigned reading

REVIEW

doi:10.1038/nature24286

DNA sequencing at 40: past, present and future

Jay Shendure[1,2], Shankar Balasubramanian[3,4], George M. Church[5], Walter Gilbert[6], Jane Rogers[7], Jeffery A. Schloss[8] & Robert H. Waterston[1]

This review commemorates the 40th anniversary of DNA sequencing, a period in which we have already witnessed multiple technological revolutions and a growth in scale from a few kilobases to the first human genome, and now to millions of human and a myriad of other genomes. DNA sequencing has been extensively and creatively repurposed, including as a 'counter' for a vast range of molecular phenomena. We predict that in the long view of history, the impact of DNA sequencing will be on a par with that of the microscope.

DNA sequencing has two intertwined histories—that of the underlying technologies and that of the breadth of problems for which it has proven useful. Here we first review major developments in the history of DNA sequencing technologies (Fig. 1). Next we consider the trajectory of DNA sequencing applications (Fig. 2). Finally, we discuss the future of DNA sequencing.

visualization in two dimensions, with the resulting positions diagnostic of their size and sequence[5].

The invention of DNA sequencing

Early attempts to sequence DNA were cumbersome. In 1968, Wu reported the use of primer extension methods to determine 12 bases of the cohesive ends of bacteriophage lambda[6]. In 1973, Gilbert and Maxam reported 24

Python

Why Python, is it fast, is it easy to learn?

Python

Easy to use, easy to read, extendable (e.g., C++ binding), mature. Python is the glue that binds programs/code/web together.

Interactive Modern Python (IPython)

Although it has been around for decades, Python has exploded in popularity in the last few years owing to its well developed data science libraries and interactive scripting tools. We will be learning modern interactive Python usage.

```
In [463]: # create a string variable called dna
          dna = "ACTGCATCAGACTAGCAT"
```

```
In [ ]:

In [ ]:

In [ ]:
```

Assignment

Complete Reading and notebooks for session 2 at https://eaton-lab.org/slides/genomics. Note that the reading is different from that listed in the syllabus. You only need to read chapters 1, 3, and 4.

```
In [463]: # create a string variable called dna
          dna = "ACTGCATCAGACTAGCAT"
```

```
In [ ]:

In [ ]:

In [ ]:
```