

Computing for Next Generation Biology

Thomas Girke Tutorial

Lab: Peter Morrell

Chaochih Liu

August 14, 2015

Novel Problems in Biology

1. Hundreds of gigabytes of data
2. Computationally-intense algorithms
3. Solution: Use a super computer!

MSI Quirks

Module system

- Local user: account on your computer
- login.msi.umn.edu: any of the MSI login nodes
- resource.msi.umn.edu: any one of the systems behind MSI login nodes
 - i.e. Lab, Mesabi, etc.

MSI queuing system (PBS Queue)

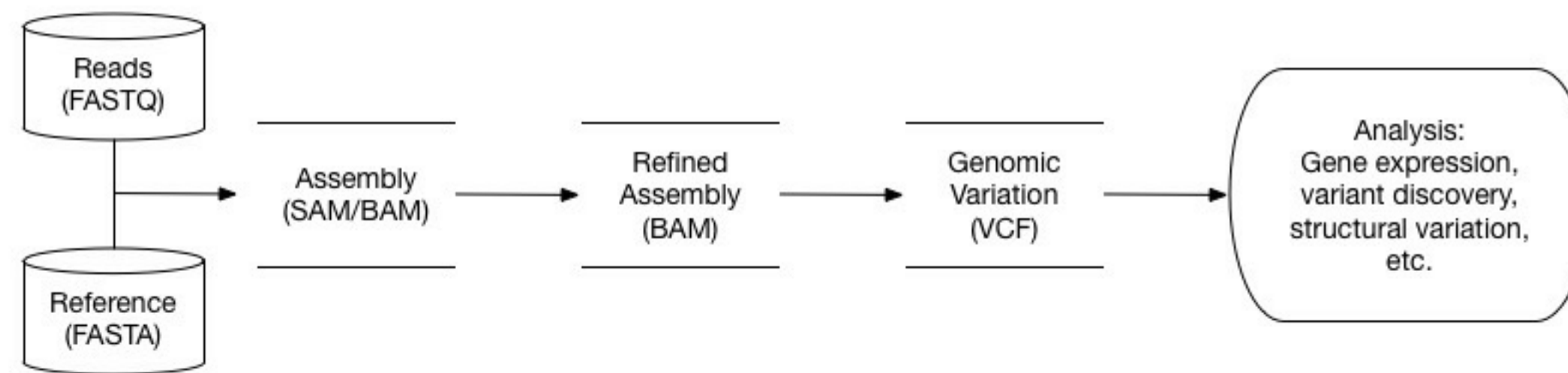
- Called PBS (Portable Batch System)
- Create PBS job scripts to submit a job
- Use qsub scriptname to submit a job
- Use qstat -u username to check on all jobs you have submitted
- Use qdel jobIdnumber to cancel a submitted job

Scripts

You will likely find yourself working in a UNIX-like environment (MSI runs Linux)

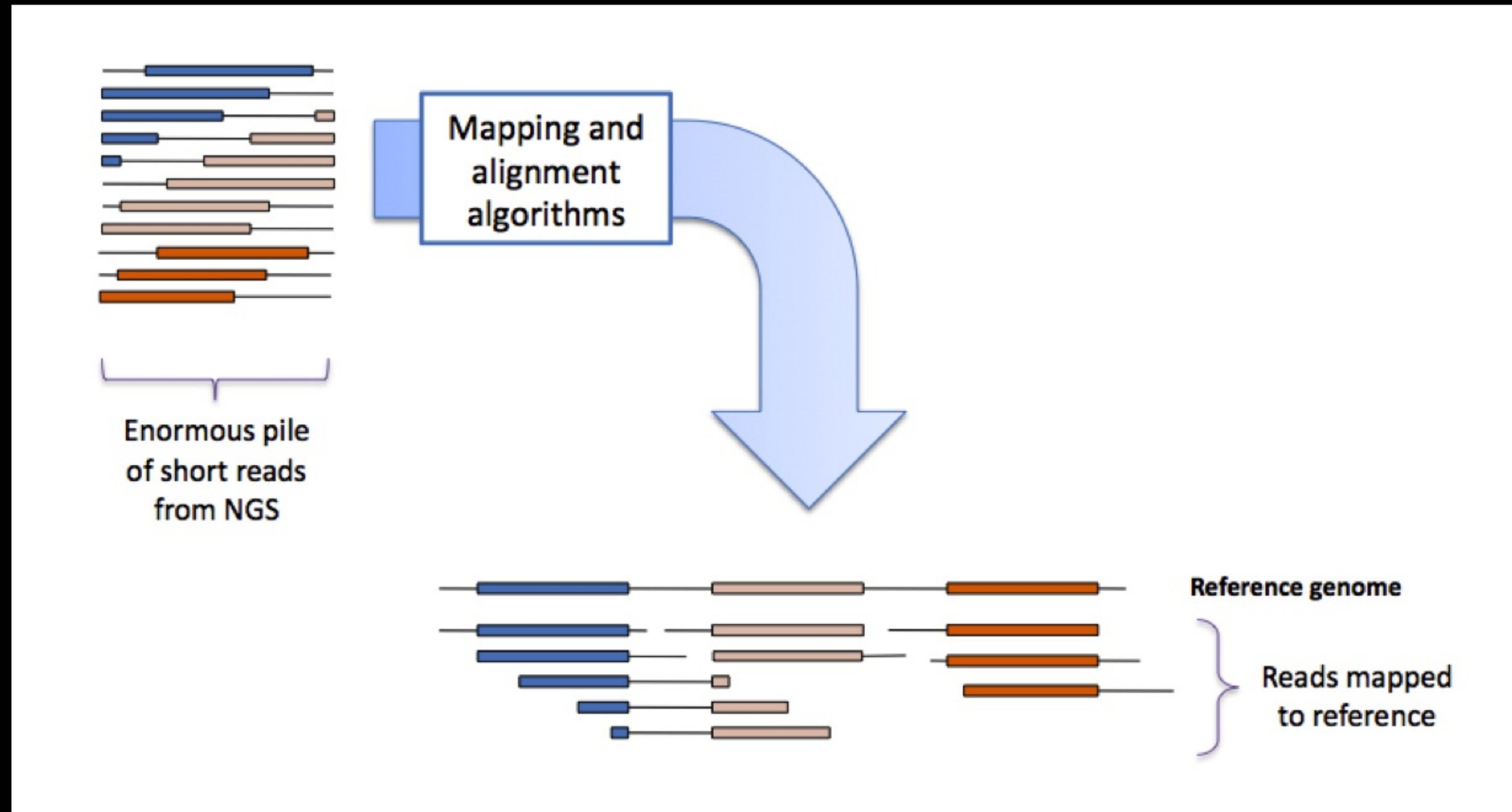
- Specific, optimized tools
- Easy-to-parse text files
- Pipeline-like flow of data between tools

General Workflow



Tools are given at the end of slides.

Mapping and Alignment



This figure was from GATK Broad Institute.

File Formats

Most are plain text

Well-defined and easy to parse

- Pre-built tools for parsing (SAMTools, VCFTools, etc...)

FASTQ File Format

Four Fields

```
@DJB775P1:248:D0MDGACXX:7:1202:12362:49613
TGCTTACTCTGCGTTGATACCACTGCTTAGATCGGAAGAGCACACGTCTGAA
+
JJJJJIIJJJJJJHHHHGHFFFFFFFCEEEEEEDBD?DDDDDDBDDBDDABDDCA
```

- Name (description line) - starts with @
- Sequence data
- End of sequence
- Quality data - same length as sequence

Git

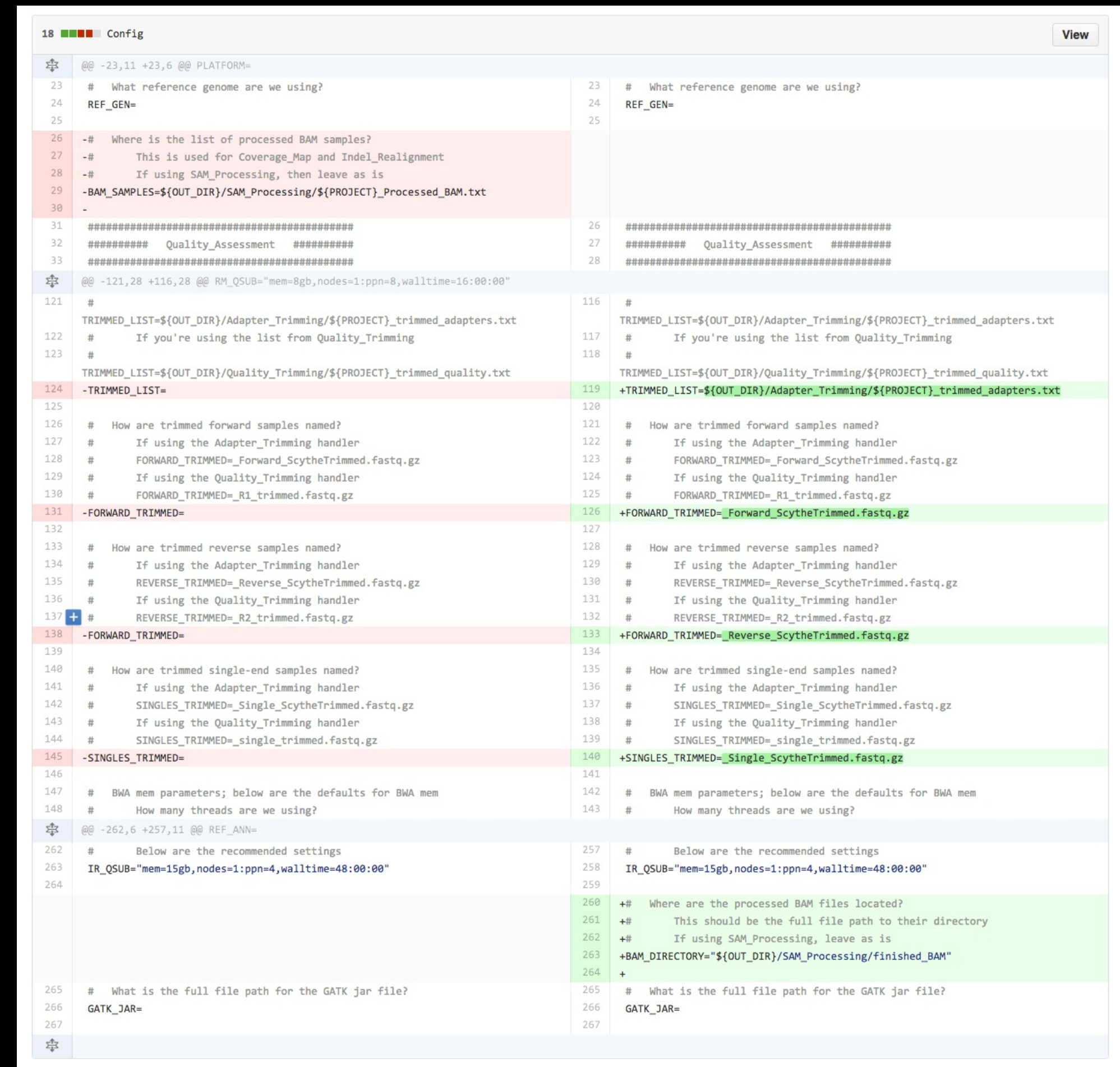
Version Control System

- Keep track of changes
- Use specific version of code
- Free software

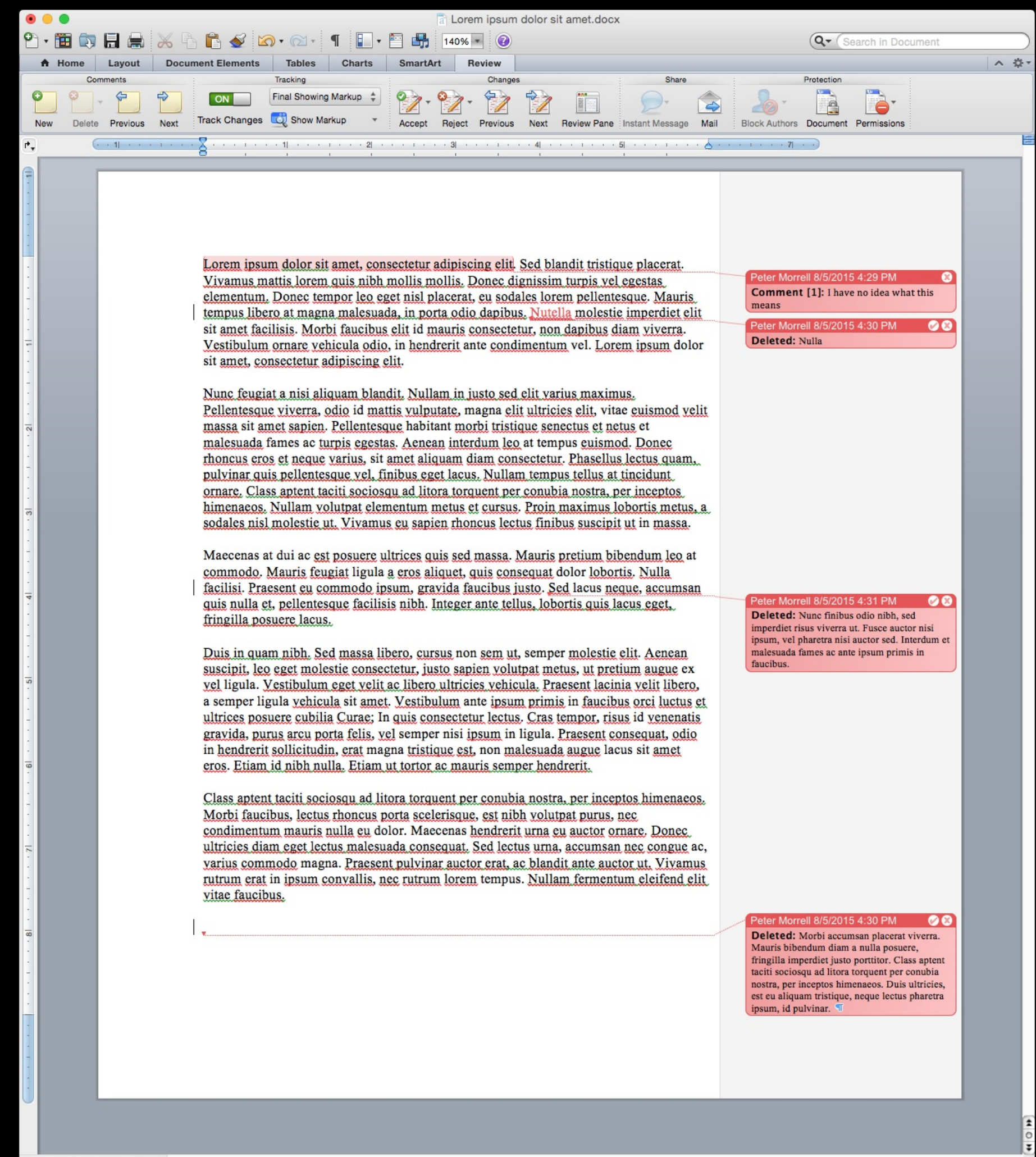
Command-line tool or GUI application

Git Vs. MS Word

Git Track Changes



MS Word Track Changes



Intro to GitHub

Collaboration

- Review changes
- Comment on code
- Report issues

Sharing of code


- Get feedback

User friendliness


- Web-based graphical interface
- READMEs and Wiki pages

An account is not necessary to view and download public repositories.

Vince Buffalo's *Bioinformatic Data Skills* Book

 This repository Search

Pull requestsIssuesGist

 vsbuffalo / bds-files

Watch 9Star 36Fork 19

Supplementary files for my book, "Bioinformatics Data Skills"

70 commits1 branch0 releases2 contributors

Branch: master bds-files / +

added one more untracked file

vsbuffalo authored on Jun 23latest commit 30d172dd40

chapter-00-preface	updated preface readme	5 months ago
chapter-01-ideology	updated readme for preface	5 months ago
chapter-02-bioinformatics-projects	updated preface readme	5 months ago
chapter-03-remedial-unix	changes that came up during book editing	2 months ago
chapter-04-working-with-remote-machines	added section on security of SSH forwarding	3 months ago
chapter-05-git-for-scientists	edits to resources section, new diff ex file	5 months ago
chapter-06-bioinformatics-data	changes that came up during book editing	2 months ago
chapter-07-unix-data-tools	better title for section	2 months ago
chapter-08-r	added one more untracked file	a month ago
chapter-09-working-with-range-data	added joins to main directory of ch13	4 months ago
chapter-10-sequence	more material	5 months ago
chapter-11-alignment	added joins to main directory of ch13	4 months ago
chapter-12-pipelines	minor typo corrections	4 months ago
chapter-13-out-of-memory	added section on security of SSH forwarding	3 months ago
chapter-conclusion	added section on security of SSH forwarding	3 months ago
LICENSE	changed to BSD license, permissive is good	6 months ago
README.md	readme markdown error updates	5 months ago

<> Code

Issues 0

Pull requests 0

Wiki

Pulse

Graphs

HTTPS clone URL

https://github.com/vsbuff

You can clone with HTTPS, SSH, or Subversion.

Clone in Desktop

Download ZIP

Useful Git Commands

Create Repositories

- To obtain a repository from an existing URL

```
git clone [url]
```

- To create a new repository

```
git init [project_name]
```

Synchronize Changes

- Downloads history and updates changes
- You must be in the repository before you run this command

Exercise 3.1: Downloading Dataset from GitHub

Dataset used: from Vince Buffalo's book *Bioinformatics Data Skills" - Chapters 10 and 11

Vince Buffalo's GitHub Page is open to the public

- Creating an account is free but not necessary for *today*

Use the following git commands in terminal:

```
# Make sure you are in the directory ~/Itasca
# To download files in Vince's GitHub repository (this may take a couple minutes)
git clone https://github.com/vsbuffalo/bds-files.git
```

Now check to make sure it has been downloaded to ~/Itasca

Exercise 3.2: Viewing FASTQ File

Now that we have a directory called bds-files in ~/Itasca with Vince's dataset, go into bds-files and see what's there (hint: use cd command).

We will use the find command we used in Lec 1 to find which directory the .fastq files are located in. Don't remember the exact command? Use a combination of history and grep commands to find it

We will search through our entire history with history and pipe the STDOUT to grep.

```
history | grep '*.fasta'
```

Once we find the command, we will change the *.fasta part to *.fastq

FASTQ File Format

This file format is used to store high-throughput sequencing data, reported with a per-base quality score indicating confidence of each base call.

Four Fields

```
@DJB775P1:248:D0MDGACXX:7:1202:12362:49613
TGCTTACTCTGCGTTGATACCACTGCTTAGATCGGAAGAGCACACGTCTGAA
+
JJJJJIIJJJJJJHHHHGHFFFFFFCEEEEEEDBD?DDDDDDBDDBDDABDDCA
```

- Name (description line) - starts with @
- Sequence data
- End of sequence
- Quality data - same length as sequence

BED/GFF3 Format

Describe regions or intervals.

Must be paired with a reference.

BED: usually just intervals. Useful for masking or extracting pieces of sequence.

- BED12 format is the more modern version and more descriptive

GFF3: intervals and qualifiers. Used in genome browsers to list mRNA, CDS, UTR, repeat regions, restriction sites...

BED file format

Example BED file from Vince's Chapter 11

```
1 216596556 216596738
1 216595194 216595882
1 216591856 216592021
1 216538295 216538427
1 216500933 216500996
1 216498647 216498941
1 216497510 216497694
1 216496816 216497037
1 216495225 216495318
1 216465517 216465712
1 216462622 216462752
1 216424245 216424440
1 216419927 216420568
1 216405295 216405478
1 216390729 216390892
1 216380615 216380773
1 216373363 216373463
```

GFF File Format

Columns in Morex_Annotations_WithPhase.gff:

1. Seqname: name of chromosome
2. Source: name of program that generated feature
3. Feature type name: i.e. Gene, Variation, Similarity
4. Start position
5. End position
6. Score – floating point value
7. Forward (+) or reverse(-) strand
8. Frame: either 0 (first base of codon), 1 (2nd base of codon) or 2
9. Additional info about each feature

Example GFF file:

```
morex_contig_1083669 mips predicted_gene 1 212 . - . ID=MLOC_1000
morex_contig_1083669 mips mRNA 1 212 . - .
ID=MLOC_1000.1;Parent=MLOC_1000
morex_contig_1083669 mips exon 1 212
```

Exercise 3.3: View BED Files

Let's try viewing a bed file:

```
# Go back to the bds-files directory
cd ..

# Run the same find command we've been using to find where the BED file is located
find `pwd` -name "*.bed" | sort
```

Where are the .bed files located?

Exercise 3.3: View BED Files

We see the .bed files are located in chapters 6, 7, 8, 9 and 11. We'll use the .bed file in chapter-11-alignment called USH2A_exons.bed.

```
# First, go into that directory and find a file called 'USH2A_exons.bed'
```

```
# View the file using the 'less' command  
less USH2A_exons.bed
```

SAM/BAM Format

Contains alignment information

- Start/end coordinates
- Mapping quality
- Mates
- And so on

SAM: Plain text human readable format (cut, grep, sed, etc...)

- Use SAMTools view to view SAM files

BAM: Binary(Compressed)

BAM Format

Order of magnitude smaller than SAM file

Computer readable

Here is what a BAM file looks like if you try to view it with head (not very useful to us):

?????#~x?[]??δ?>????*?i?q????zž?R???]s?O\??/S?l??h?g???i??.??j??P?*??×??:??O\f?~?js?+_?]x?[]?yW??
 W?}??O^s?K???5*L?'?Y????@??'??8?V???W?&???~?Z??)?b©?~??=?m??8?A?w?\;;T?R?w???Y?o?:???+??
 4?OV????:?`??R??1CXpo[]3??OX???Δb?/?<q???V\?pe?8???_?i
 ????J??úG???o???f[?
 ?c2?bQ
 6? s??>5??wl???~???33??8y??q!??u??D????4?
 .r?N?4;??C?(l?~2?R??b<pn?+?/
 Br?|8,??s??#7
 ?Z???~?QS²?????;??m_
 []?; .????}??7??
 ?BOKpV/_L??`l??G?????)e?w????l?ee??r??_Qad??Bc?#

SAM Format

@SQ header: stores info about the reference sequence

@RG header: contains read group and sample metadata

@PG header: metadata about programs used to create/process SAM/BAM files

First line of alignment section – doesn't begin with '@'

Note: some programs need SAM files to have an @HD header at the beginning to work

To view SAM file headers use `samtools view -H file_name.sam | head -n 10`:

```
@SQ SN:I   LN:15072434
```

```
@SQ SN:II  LN:15279421
```

```
@SQ SN:III LN:13783801
```


Exercise 3.4: View SAM/BAM Files

Use the same find command we've been using to find where the .sam file is located:

```
find `pwd` -name "*.sam" | sort
```

We see the .sam files are located in chapter-11-alignment. Go into chapter-11-alignment and look for .sam file.

```
# View head of SAM file celegans.sam with 'head' command  
head celegans.sam
```

Now try to view celegans.bam using the head command. This doesn't output information that is very useful to us. Try using samtools view celegans.bam | head -n 10 to view the .bam file.

Note: .sam and .bam files are typically relatively large in real datasets so using less

VCF/BCF Files

Describes mismatches from reference.

Must be paired with reference sequence.

Information includes:

- Type (SNP, MNP, Indel...)
- Chromosome (or contig)
- Position
- Quality, depth...

Exercise 3.5: View VCF Files

If you are not already in the directory, cd into the directory containing .vcf file.

Since the file extension is .gz, we will have to unzip the file first. To unzip the gzipped file use zcat command (If using a local Mac, use gunzip instead) then pipe to head to view beginning of file.

```
zcat NA12891_CEU_sample.vcf.gz | head
```

Run the same command without piping to head and see what happens.

Tools

Aligners

- BWA: software package for mapping sequences against large reference genomes.
- Bowtie/Bowtie2: fast and memory efficient tool for aligning sequencing reads to long reference sequences.
- MOSAIC-aligner: a reference-guided aligner for NGS technologies.

SAM/BAM handlers

- SAMtools: includes various utilities for sorting, merging, indexing and generating alignments in the SAM format.
- Picard: provides command line tools for manipulating high-throughput sequencing data and file formats such as SAM/BAM/CRAM and VCF.
- BamTools: toolkit for handling BAM files.

Refining Assemblies

PacBio RS II

Newer sequencing technology than NGS.

Single Molecule, Real-Time (SMRT) DNA sequencing system.

- High consensus accuracy
- Long read lengths

Ideal for:

- *de novo* assembly
- Characterization of genetic variation
- Methylation analysis
- Microbiology studies
- And more...

Exercise 3.6: More Useful Commands to Know

Go into directory with contam.fastq file. We'll use the one from chapter-10-sequence.

Now we'll copy contam.fastq from chapter-10-sequence to ~/Itasca and rename the file. Use the cp command to copy. We will rename by specifying directory the file will be copied to and giving it a new name contam1.fastq

```
cp contam.fastq ~/Itasca/contam1.fastq
```

Next, go into ~/Itasca to view the copied file.

There is already an existing file contam.fastq that we used wget to download earlier. Let's see if there are any differences between the file we just copied and the original file. We will use diff command to compare differences between files. The format will be diff -y file_1 file_2

Exercise 3.6: More Useful Commands to Know

Let's create differences and use diff again to compare the files. We will be using Vim as our text editor for today. To do this, type vim followed by the file you want to edit.

```
vim contam1.fastq
```

In Vim you won't be able to use your mouse to make changes.

- Type : to start entering command.
- Hit i to enter INSERT MODE to edit file

We will delete lines 16-19, but first let's jump to line 16:

```
:16
```

Use dd to delete current line and repeat this 4 times.

Exercise 3.6: More Useful Commands to Know

If you accidentally delete too many lines, use the undo command to undo your last change.

```
:u
```

If you type `:u` again it will undo 2 changes before that and so on.

Now let's redo our last undo. Use `CTRL+r` to redo last two changes. Now that we have made changes to our file, we will save and exit.

- `:w` saves the file
- `:q` exits out of Vim

```
:wq
```


Additional Resources

Vince Buffalo's *Bioinformatics Data Skills* - Ch. 10 and 11

Answer key to today's exercises can be viewed and downloaded from my [Gist repository](#).