

Lab 5:

Integrative epigenomics analysis

Bio5488

2/13/15

A few updates

- Assignments 4 & 5 due next Friday, 2/26 at 10am
 - No late submissions accepted
 - Assignments will be submitted *via Blackboard*

A few updates

- The original server has been retired (RIP 45.65.227.83 🙄)
- Download a local copy of your current/previous work from the *new* server
 - Username: same as before (your first name)
 - Password: same as before
 - Host: genomic.wustl.edu (note: no S in name)
- ~~We won't be using the *new* server for this class (at least yet) since IT is unable to install software on it.~~

Alternatives for moving forward

- **Option 0: use the new bio5488 server**
 - IT installed the necessary software

← This is probably now the
least headache
producing method!

- **Option 1: use Anaconda**
 - Anaconda is a popular Python distribution for scientific computing
 - *However*, you will need install some third-party command-line tools, e.g., Perl and bedtools
- **Option 2: roll your own**
 - You install python, modules, and third-party tools
- **Option 3: use the bio5488 virtual machine**
 - See instructions document on the syllabus
 - Installation demo

Thanks
Zhen ☺

Virtual machines

- A virtual machine allows you to run another operating system *alongside* your existing operating system.



A screenshot of a Mac (the **host operating system**) running a Windows virtual machine (the **guest operating system**):

VM installation

- See Instructions on blackboard [here](#)
- Instructions have been added on how to install sublime

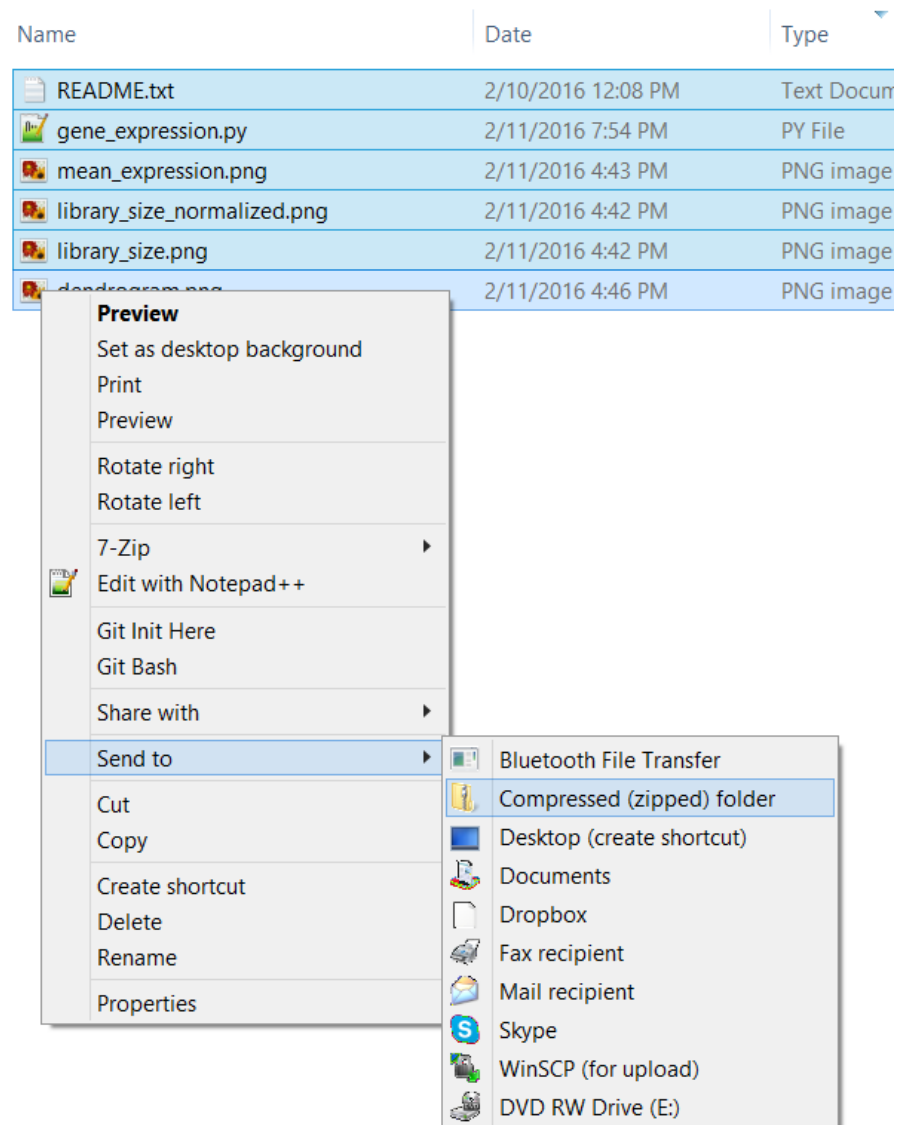
How to turn in assignments

- Now that the new server is more functional, you have 2 options for turning in assignments:
 - Method 1: place all your files in your `~/assignmentX/submission/` directory on the new server (genomic.wustl.edu)
 - Method 2: use Blackboard

Method 2: Turning assignments with Blackboard

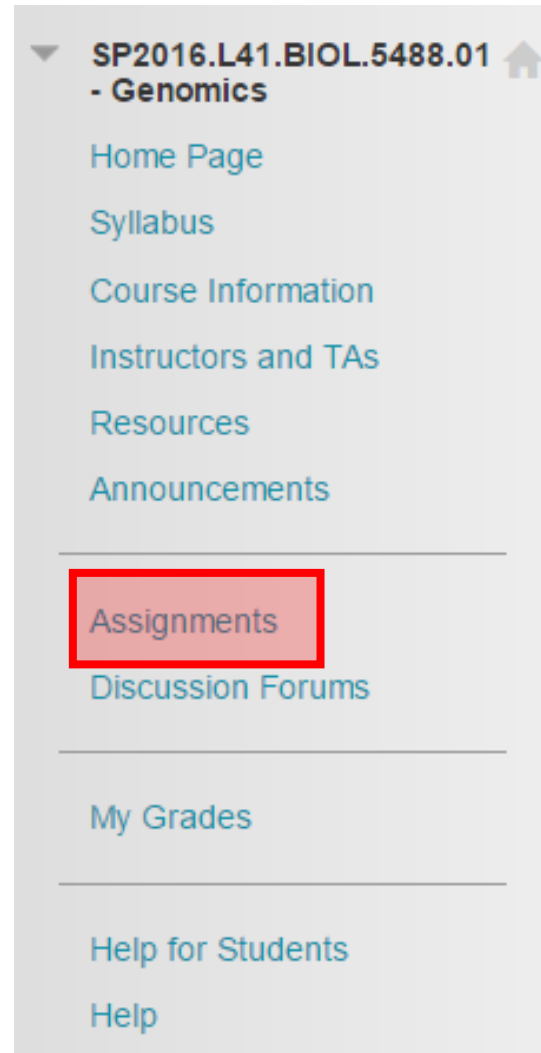
1. Zip all of your submission files together

1. Links to how to zip files for [windows](#) and [mac](#).



Method 2: Turning assignments with Blackboard

1. Zip all of your submission files together
 1. Links to how to zip files for [windows](#) and [mac](#).
2. Click Assignments



Method 2: Turning assignments with Blackboard

1. Zip all of your submission files together
 1. Links to how to zip files for [windows](#) and [mac](#).
2. Click Assignments
3. Click on *the assignment name*

Assignments



Assignment 4: Gene Expression

Attached Files: assignment4_files.zip (1.208 MB)
 student_assignment4_v4.pdf (299.342 KB)

Please zip all of the required files together.

Links to how to zip files for [windows](#) and [mac](#).



Assignment 5: Epigenomics

Attached Files: student_assignment5_v2.pdf (321.482 KB)
 assignment5_files.zip (18.503 MB)

Please zip all of the required files together.

Links to how to zip files for [windows](#) and [mac](#).

How to turn in assignments (2.0)

1. Zip all of your submission files together
 1. Links to how to zip files for [windows](#) and [mac](#).
2. Click Assignments
3. Click on *the assignment name*
4. Upload the zipped file and click Submit.

1. Assignment Information

Due Date Friday, February 26, 2016 10:00 AM	Points Possible 10
--	------------------------------

Please zip all of the required files together.
Links to how to zip files for [windows](#) and [mac](#).
[assignment4_files.zip](#) [student_assignment4_v4.pdf](#)

2. Assignment Submission

Text Submission	<input type="button" value="Write Submission"/>
Attach File	<input type="button" value="Browse My Computer"/> <input type="button" value="Browse Content Collection"/>

3. Add Comments

Comments

☒

Character count: 0

4. Submit

When finished, make sure to click **Submit**.
Optionally, click **Save as Draft** to save changes and continue working later, or click **Cancel** to quit without saving changes.

<input type="button" value="Cancel"/>	<input type="button" value="Save Draft"/>	<input type="button" value="Submit"/>
---------------------------------------	---	---------------------------------------

Lab 5:

Integrative epigenomics analysis

CpG Islands (CGIs)


- Vertebrate genomes are CpG-poor and contain mostly methylated CpGs
- However, there are exceptions to this rule: **CpG islands**
 - CGIs are short genomic regions that are GC-rich, CpG-rich, and predominantly unmethylated
- CGIs are important regulatory regions
 - Ex: ~70% of promoters contain CGIs
 - CGI promoters have distinctive patterns of transcription initiation and chromatin configuration

Lab 5: Integrative epigenomics analysis

- Goal
 - Explore the epigenetic profile of CGIs
 - Compare technologies for assessing the methylome
- Input
 - Epigenomic datasets from a human brain germinal matrix
 - Annotation files
- Output
 - Descriptions of methylation signal genome-wide, in CGIs, in promoter-CGIs and non-promoter CGIs
 - Correlation of methylation signals from different methylome technologies

Input datasets

Methylation datasets



Check out the
appendix for a
description of
each input file

File	Description
BGM_WGBS.bed	Whole genome bisulfite sequencing (WGBS)
BGM_MeDIP.bed	Methylated DNA immunoprecipitation sequencing (MeDIP-seq)
BGM_MRE.bed	Methyl-sensitive restriction enzyme sequencing (MRE-seq)

Annotation datasets

File	Description
CGI.bed	Locations of CGIs
CpG.bed	Locations of CpGs
refGen.bed	Locations of RefSeq genes

Datasets have been reduced to just chr22
Coordinates are based on hg19

Bed file format

- A concise and flexible file format used to define genomic features and annotations, e.g., CpGs
- .bed extension, e.g., CpG.bed
- Tab-delimited file
- Each feature is defined on one line
 - Required fields: genomic coordinate
 - Optional fields: feature name, score, strand & many others
- See <http://genome.ucsc.edu/FAQ/FAQformat.html> for a complete description

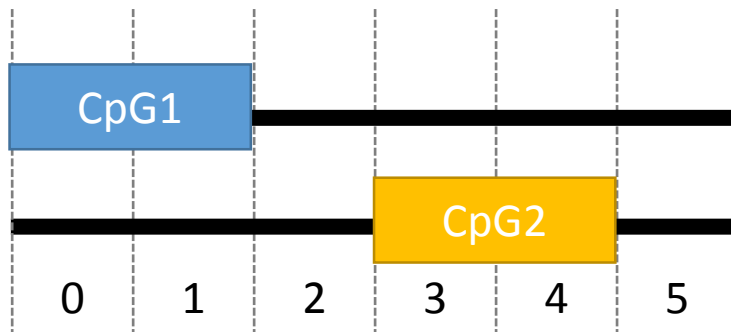
Example bed file

chr21	9411551	9411553
chr21	9411783	9411785
chr21	9412098	9412100

Genomic coordinates in bed files

Column	Description
0	Chromosome/contig name
1	Start coordinate <ul style="list-style-type: none">The left coordinate of the feature0-based and inclusive
2	End coordinate <ul style="list-style-type: none">The right coordinate of the feature0-based and exclusive
3	Name of feature
4	Strand of feature

Example



chromosome	start	end	name	strand
chr1	0	2	CpG1	+
chr1	3	5	CpG2	-



bedtools: a powerful toolset for “genome arithmetic”

- Bedtools is a useful command-line tool for manipulating bed files

- To see the help menu

```
$ bedtools --help
```

- Consists of a suite of sub-commands

```
$ bedtools [sub-command] [options]
```

- Example: find overlapping features

```
$ bedtools intersect -a a.bed -b b.bed
```

- The documentation is very informative (descriptions, examples, and figures, oh my!):

<http://bedtools.readthedocs.org/en/latest/>

Convention: optional command line arguments are enclosed in “[]”. When you specify the arguments, don’t type the “[]”

Quantifying methylation level with WGBS

- After bisulfite treatment:
 - Methylated Cs (mC) are “protected” and are sequenced as Cs
 - Unmethylated Cs are sequenced as Ts
- Calculating the methylation level at a C:

$$\text{methylation level at pos } i = \frac{\# \text{ C basecalls at pos } i}{\# \text{ C basecalls at pos } i + \# \text{ T basecalls at pos } i}$$

- Methylation level ranges from 0% to 100% methylation

Quantifying methylation level with WGBS (cont.)

- Calculating the methylation level at a C:

$$\text{methylation level at pos } i = \frac{\# \text{ C basecalls at pos } i}{\# \text{ C basecalls at pos } i + \# \text{ T basecalls at pos } i}$$

Position	0	1	2	3	4	5	6
Reference	C	G	A	C	G	C	A
Reads	C	G	A	T	G	T	A
	T	G	A	T	G	T	A
	T	G	A	T	G	T	A
	T	G	A	T	G	T	A
	$\frac{1}{1+3} = 0.25$			$\frac{0}{0+4} = 0$			

Not in a CpG context so the base ignored

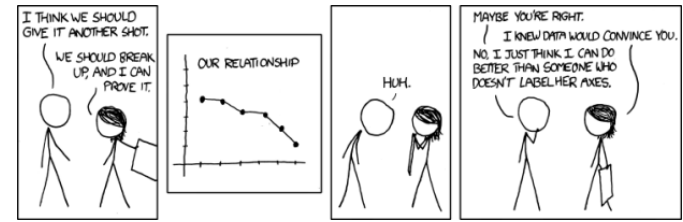
Plotting in Python

- If your script creates a plot, **always** include the following lines **in this order**:

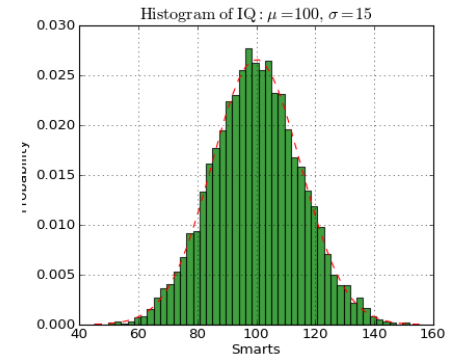
```
1  # Setup plotting
2  import matplotlib
3  matplotlib.use('Agg')
4  import matplotlib.pyplot as plt
```

- This tells matplotlib to save the images as files rather than display the image in a popup window
 - The 'Agg' (Anti-Grain Geometry) tells matplotlib to save the files as pngs.
- This is called “configuring the backend”
- Read more about it here:
http://matplotlib.org/faq/usage_faq.html#what-is-a-backend

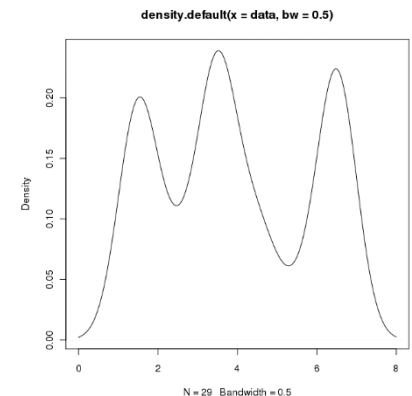
Data visualization



Example histogram



Example density plot



- All plots **must** have informative axis labels, titles, etc.



TIP: check out the matplotlib documentation:

http://matplotlib.org/api/pyplot_api.html



TIP: when plotting distributions, e.g., with histograms, check that the data is not over or under smoothed

- Histograms: adjust the # of bins
- Density plots: adjust the bandwidth

Good practice: Create dynamic output filenames

- Suppose we have a script that takes in command line argument, e.g., a file, and creates an output file

	Description	
	Utility	
	An “it works” solution	A better solution
	<ul style="list-style-type: none">• Hardcode the name of the output file	<ul style="list-style-type: none">• Name the output file based on the command line argument
	<ul style="list-style-type: none">• The output file is always overwritten• You can’t tell from the output filename alone, which command line argument was used to generate the output file	<ul style="list-style-type: none">• If you run the script with different command line arguments, the output is not overwritten!• Output filenames are descriptive!

Good practice: Create dynamic output filenames

What we have: `/home/assignments/assignment5/BGM_WGBS.bed`

```
>>> import os
>>> path_to_file = "/home/assignments/assignment5/BGM_WGBS.bed"
>>> path_to_file
'/home/assignments/assignment5/BGM_WGBS.bed'
```

```
>>> # Grab just the filename
... basename = os.path.basename(path_to_file)
>>> basename
'BGM_WGBS.bed'
```

```
>>> # Remove the extension
... basename_no_extension = os.path.splitext(basename)[0]
>>> basename_no_extension
'BGM_WGBS'
```

```
>>> # Create plot name
... plot_name = basename_no_extension + ".png"
>>> plot_name
'BGM_WGBS.png'
```


Writing to output files in Python

- Previously, we printed output text to the terminal
- Alternatively, we can print output text to a file
 - Analogous to performing a “save as” operation

Example

```
1 # Open output file for writing
2 output_fileobject = open("results.txt", "w")
3
4 # Write to the file
5 print("Genomics rules!", file=output_fileobject)
6
7 # Close the file
8 output_fileobject.close()
```

Code template

```
<fileobject> = open(<filename>, "w")
print(<string>, file=<fileobject>)
<fileobject>.close()
```

- **If you write to a file that already exists, you will overwrite it!**
- Only call open and close 1X each
- Call print every time you want to print a line to a file

Assignment 5: requirements



TIP: start early!

- Due next Friday (2/26/16) at 10am
- Comment your code
- Your submission folder should contain all scripts that you wrote, output files, and a README
 - See assignment PDF for the list of files 😊

Assignment 5: Optional extra credit

- Examine H3K4me3 across different types of CGIs
 - See the assignment PDF for instructions
- Due when assignment 5 is due
- Comment your code
- Your submission folder should contain all scripts that you wrote, output files
 - *Include your comments and commands in the same README as assignment 5*
 - See assignment document for list of files 😊