**Lesson Plan for BMS Bootcamp Session: Coding in Biomedical Research**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Section | Time | Content | Outcomes | Interactive |
| Introduction | 2:00 | Outline, goals | Establish interest in session | Ask about students’ background |
| Coding in Practice | 10:00 | Examples, languages, how to write a script | Familiarity with terms of the field; understand relevance; relate to coding as a new skill | Ask students to fill in the blanks as we relate write a script to doing a microarray |
| Paper exercise | 15:00 | FizzBuzz test | Thinking like a programmer | Students will code on paper |
| Computer exercise | 15:00 | Write a Python script that prints the bridge to Hey Ya | Learn basic coding concepts; more comfortable with Python | Students will code along with the instructor |
| Resources, part 1 | 3:00 | Python resources, finding help online | Ability to continue to learn after class | Handout |
| Break | 15:00 | --- | --- | --- |
| Bioinformatics workflow | 15:00 | Steps, considerations, collaboration, writing good code | Familiarity with bioinformatics methods, understanding of what makes a project pipeline good or bad | Ask students to fill in the blanks as we discuss considerations |
| Bioinformatics exercise | 25:00 | From a paper, find data, load into R, find methods to handle data & recreate a figure | Awareness of GEO/online datasets; more comfortable with R; able to find coding help online | Students will code along with the instructor |
| Resources, part 2 | 3:00 | Campus resources, R & package resources | Ability to continue to learn after class | Handout |
| Conclusion | 2:00 | Outline, outcomes achieved | Establish interest in and comfort with learning to code | Ask for thumbs up/down on confidence in each area covered |

**Challenge Yourself (Exam Questions)**

To start out, you can use the online interpreter: http://repl.it. Remember to use good code practices: useful variable & function names, modular & reusable code, readable code (lots of tabs and empty lines), documentation, and reproducibility.

1. Modify the above script to add the following topics: error messages, reading from and writing to a file, structures, classes, float operations

2. Write a script which randomly generates a DNA sequences of a given length. Expand your script to give the complementary sequence. Remember to follow the pipeline: choose your algorithm, choose your language, prototype, debug, and keep records.

3. Try the “Python Village” challenges and beyond at http://rosalind.info/problems/locations/

4. Try making a script to rename files in a folder. For instance, you could rename all your “Image002.jpg” files to “WesternBlot\_[date blot imaged].jpg”.