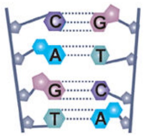


syllabus for

DNA

comp/bio 242

ATTGCGCGATGCGTAGGGC
ATGCATGGCAAAACGGTAT
ATATGCGTATTGCAAAGGC
TATGCT**ATCGC**ATTATGCT

Instructor: **Mark LeBlanc** mleblanc@wheatoncollege.edu

SC-1322 508.286.3970
Office Hours:
Tue 2:30-3:30, Wed 10:30-noon
Thur 9:30-10:30 or appt.

Meeting Times: Tuesday and Thursday 12:30 – 1:50
Room SC-1315, csLab
Text (online): http://interactivepython.org/runestone/static/thinkcspy/toc.html
Brad Miller and David Ranum
MOOC (online): https://www.udacity.com/course/tales-from-the-genome--bio110
Udacity.com

10 GOALS for this course

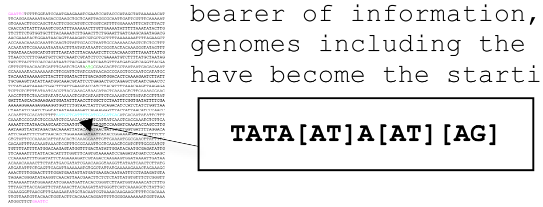
(0) You are at a cocktail party and the topic of genomes comes up. You are able to **recall** significant phrases, terms, and techniques and your **understanding** of the main ideas and concepts enables you to lead the conversation for a while … which causes your friends to raise their eyebrows.
(1) You learn to **identify** and **classify** problems that are candidates for a computer to handle; this is the start of "computational thinking" …
(2) You **demonstrate** the ability to **think algorithmically**, breaking what originally seems like an overly complicated problem into a series of smaller, manageable tasks.
(3) You learn to **craft creative solutions** by "**writing software**" ("to program", "to write software", "to script", "to cut code", …).
(4) You appreciate the importance of the **Human Microbiome Project**.
(5) You **design experiments** to first solve small computationally-intensive tasks (e.g., on one gene sequence) and then **scale** your solutions to very large sets of data (e.g., all genes in a genome).
(6) You **learn** to perform tasks in **the Linux (Unix)** operating system.
(7) You learn to professionally **document** your software (Methods) and produce quality scientific reports, including Results and Discussion.
(8) You begin to appreciate the (soon to be) revolution in **personalized medicine**, including knowing your way around a **23andMe** report.
(9) You feel empowered to **evaluate** the ethical implications of your work and learn to **appraise**, **critique**, and **defend** your own as well as the work of others.

Connections: The Bigger Picture

This course is part of the ***connection "Genes in Context"*** *with Philosophy 111 (Ethics) or Philosophy 241 (Bio-Ethics).* Throughout the semester, students will be exposed to the ethical aspects of living in a post-genomic world and the increasing use and challenges of sequenced genomes as applied to "**personalized medicine**". To reinforce the connection, teams of students will study one company (e.g., 23andMe) and tour around their professor's personalized genetic report from 23andMe, and then produce **one-minute youTube "commercials"** either for or against companies currently promoting and selling medical profiles based on individual genomes. The commercials will be framed from one of two points of view: (i) from the point of view of the company or (ii) from a consumer advocacy point of view.

Catalog Description

An amazing blend of science, computing, and mathematics emerges when considering the molecule "Deoxyribonucleic Acid" (DNA). DNA is the blueprint of life for all organisms on Earth and throughout evolutionary time. Its distinctive and beautiful physical nature, a double helix of four bases, maps onto its functionality as a bearer of information, generation after generation. Fully sequenced genomes including the human genome and hundreds of microbial genomes have become the starting point for attempts to answer a wide range of biological and quantitative questions.



This DNA-242 course satisfies the **Quantitative Analysis (QA**) requirement.

Exact pages to read (in our online, interactive text) and specific use of web sites will be **assigned in class.** URLs are listed as assigned/needed in the onCourse (Moodle) website.

Notes on grading, due dates, and submission procedures

You must attend all classes. When **homework** is "due in class", this means at the beginning of the class on that particular day, e.g., homework will be collected at the start of class. Homework that arrives after the class and up to one day late will lose 10%. No homework is accepted after one-day late.

Programs are due on various dates (see syllabus for approximate due dates; but all *real* due dates will be listed on the program directions, the specification or "spec"). Since I know from experience that many students like to use the last night for testing, I will allow you to submit your programs until 5am the following day. For example, Program #1 is **due FRI, Sept 11th, but you can submit it electronically until 5am SAT, Sept 12th** -- Careful! The course website (onCourse) makes it appear as if the program is due on **Saturday**, but remember, **Saturday at 5am**! In addition to **electronically submitting** your Python programs (more instructions will be given in class as to how to do this via onCourse), you must also submit (i) a **hardcopy printout** of your program (your "source code") and (ii) a copy of your program's OUTPUT [all stapled!] in class after it is due.

Note that Python programs must *always* be **printed** in **landscape mode** and your source listing must be **stapled** to your sample **output.** (Your professor will mutter: "@%^$#&*" and will deduct points if you do not print your code in landscape (sideways) mode or if you do not staple your code and output together).

It is expected that you spend *at least* **2 hours** on **reading and practice problems** for every hour of lecture. This computes to *at least* **6 hours of work outside of class per week.** This should be done throughout the semester. Please assume all deadlines are fixed. Obviously, see your professor if you know of a conflict beforehand.

Honor Code Revisited: It goes without saying that all submitted work will be the student's own, in keeping with the Wheaton Honor Code. For in-class labs, you may get help from fellow classmates, but remember that all submitted work must be your own. All homework, Python programs, and other hand-written work must be your own from beginning to end unless otherwise noted in the instructions (e.g., paired work).

YOUR GRADE

5% **ATTENDANCE/PARTICIPATION**	- based on attendance/participation in all sessions
10% **QUIZZES** (about 5)	- see onCourse schedule (and notes at start of class)

15% **HOMEWORKS**

Sept 03	Homework #1 –	1 point	- Building Quiz #1
TBD	Homework #2 –	6 points	- Regular Expressions "RegEx" (Parts I and II)
TBD	Homework #3 –	8 points	- 1-minute "personalized medicine" YouTube video

50% **PROGRAMS** (six)

Fri Sept 11	Program #1	(7%)	- String Play
Fri Sept 18	Program #2	(7%)	- Chargaffin Counts
Fri Oct 09	Program #3	(12%)	- Gene Finder
Fri Oct 23	Program #4	(12%)	- "eLmer" – a motif finder
Fri Nov 20	Program #5	(12%)	- Building a Concordance

20% Final **Project**

Tue Nov 24	DRAFT Written Proposal (Intro & Methods)	(5%)
Dec 8rd or 10th	Oral Presentation	(5%)
Dec 10th	Final Paper, Software, and Results	(10%)

Throughout the semester, you will have the opportunity to show Superior Effort, for example, a homework or programming assignment has additional (extra credit) steps. Remember, I will use the following metric when determining each assignment and your final, overall grade: an 'A' will be assigned for superior effort, 'B' shows above average effort, 'C' is average effort, 'D' is incomplete, and 'F' is no-credit for no-work.

As you can see, the final projects are a significant part of your final grade (20%). Your professor will determine the team pairs. When working on teams, the collective team will be given a certain number of points and it will be up to the team members to determine how those points should be allocated. For example, if a pair was given 150 points, then the pair could decide that Person A, who did more of the work, should get 80 and Person B should get 70. If the pair felt they each did the same amount of effort, then each would get 75 points. For your final project, each of the three parts (Proposal, Oral Presentation, and Final Paper which includes your software and results) will receive a separate grade.

At the end of the day, this is a course to teach you how to think algorithmically, automate your problem solving, and learn how to write your own programs to process your own data in your own experiments … some day, down the road.