Teaching Portfolio

Alex M. Maldonado, PhD

Contents

Teaching statement

Education is inherently transformative. Each day offers countless opportunities for growth, understanding, and self-discovery. Educators can harness and amplify this natural development process through intentional guidance in thoughtfully designed learning environments. This principle shapes my teaching philosophy: I strive to be reliable, affable, and malleable by creating inclusive spaces that employ evidence-based strategies and adapt to students' evolving needs.

Modern challenges

Computational biology education stands at the intersection of rapidly evolving technologies and the established frameworks of academia. Tackling its unique challenges demands a thoughtful integration of cutting-edge advancements with effective pedagogical strategies.

Highly interdisciplinary

Computational biology draws from calculus, computer science, data science, machine learning, statistics, statistical mechanics, physical chemistry, biochemistry, biophysics, genetics, molecular biology, and more. These diverse areas converge to address complex biological problems that require both breadth and depth of knowledge.

For example, molecular simulations require a deep understanding of Newton's laws of motion, molecular ensembles for capturing conformational dynamics, and physical chemistry principles to model noncovalent interactions. However, this theoretical knowledge must be complemented by practical skills in programming, big data, and high-performance computing to effectively simulate protein folding, dynamics, and interactions. Together, these elements form the backbone of advancements in fields such as drug discovery, protein engineering, and structural biology.

Transcriptomics integrates genetics, statistical modeling, and machine learning with data science and high-throughput computing to analyze gene expression data. Key techniques include dimensionality reduction, differential expression analysis, and computational pipelines for handling large-scale RNA sequencing datasets. These approaches enable insights into gene regulatory networks, functional genomics, and disease mechanisms, exemplifying the synergy of computational and biological sciences in modern research.

Limited educational resources

A central challenge in computational biology education is the scarcity of up-to-date, accessible resources tailored to undergraduates. The rapid pace of advancements in the field often outstrips the publication of textbooks, many prohibitively expensive and inaccessible to students from lower socioeconomic backgrounds. This issue is further compounded by the inherently multidisciplinary nature of computational biology. While excellent resources exist in fields like biology, physics, chemistry, and computer science, they often lack the integration necessary for real-world applications in computational biology. For example, a physics textbook might discuss noncovalent interactions but fail to explore their relevance to classical force fields in biomolecular modeling.

Computer science materials may cover FM-index or de Bruijn graphs without linking them to sequence alignment or genome assembly.

The DIKW (Data-Information-Knowledge-Wisdom) pyramid highlights a common challenge: students often understand individual concepts in computational biology but struggle to synthesize them into actionable insights. Bridging this gap requires cohesive teaching materials and interdisciplinary curriculum design that scaffold learning. By guiding students through the DIKW hierarchy, educators can help them progress from understanding data to applying integrated knowledge and wisdom in solving complex biological problems.

Insufficiency of classroom-only learning

While classroom instruction provides the foundation for computational biology education, it alone cannot fully equip students to tackle real-world challenges. Practical skills essential for success in computational biology are best developed through hands-on experience and exposure to industry practices. Though invaluable for gaining such exposure, internships are highly competitive, leaving many qualified students without access to these opportunities. Undergraduate research is crucial in bridging the gap between classroom learning and practical application, yet economic barriers often hinder students from pursuing unpaid or low-paying research roles. This disparity disproportionately affects those from lower socioeconomic backgrounds, who may lack the financial flexibility to take advantage of such opportunities. Additionally, mentoring undergraduates frequently falls on a small group of faculty, particularly in computational fields where mentors, resources, and time are already scarce.

Fragmentation of computational expertise

Computational methods have become indispensable across disciplines, including physics, chemistry, biology, and engineering. However, undergraduate students often face significant challenges in identifying resources and opportunities outside the classroom that could enhance their learning and career preparation. These difficulties are compounded by the siloed nature of many academic environments, where faculty with computational expertise are scattered across various departments. Computationally focused departments are relatively rare, and faculty in these departments often face constraints in capacity or funding, limiting their ability to mentor and train undergraduate students. This fragmentation creates barriers for students seeking mentorship, research opportunities, and skill-building projects aligned with their interests. Many students are also unaware of the range of computational tools, workshops, or interdisciplinary initiatives available, leaving them without clear pathways to engage with these resources.

Digital fluency paradox

In the early days of technology, using devices and software demanded a deep technical understanding, as they were not designed with accessibility in mind. Over time, advancements have prioritized user-friendliness, resulting in tools and devices that require little to no technical expertise. This shift has democratized technology, enabling widespread adoption beyond tech-savvy individuals. However, this ease of use often obscures the underlying complexity, leaving many users —especially students—unaware of the mechanisms driving the tools they rely on. As a result, the gap between technology users and its creators has widened. Bridging this divide calls for an educational shift that emphasizes not only how to use technology but also how to understand and create it, fostering a deeper connection between consumers and innovators in a computationally driven world.

Pedagogical approaches

I employ various pedagogical strategies rooted in evidence-based practices to address the challenges inherent in computational biology education and foster a transformative learning experience. These approaches are designed to create inclusive, engaging, and adaptable learning environments that empower students to develop critical skills, navigate interdisciplinary challenges, and cultivate a deep understanding of the subject.

Constructivist learning

Constructivist learning theory emphasizes that knowledge is most effectively acquired when learners actively integrate new information with their existing cognitive frameworks. Vygotsky articulated a key aspect of this theory: the Zone of Proximal Development (ZPD), which identifies where students can achieve more with guidance than they could independently. Building on this foundation, a scaffolding process that supports learners within their ZPD can enable them to tackle increasingly complex tasks as their understanding grows.

Guided by these principles, I design my courses to ensure students construct knowledge incrementally, building on their prior experiences while minimizing cognitive overload. Understanding students' evolving ZPD is critical to this process. Instead of relying solely on initial diagnostic assessments, I maintain a dynamic understanding of students' capabilities throughout the course. For example, I use anonymous, real-time Top Hat discussions during lectures to gather immediate feedback on student comprehension. This approach encourages active participation and allows me to identify discrepancies between my perception of students' understanding and their actual grasp of the material. When such discrepancies arise, I can pivot my teaching in real-time, using tools like a tablet and stylus to provide interactive, visual explanations that clarify complex concepts. This responsive strategy ensures that instructional scaffolds remain aligned with students' needs and facilitates their progression toward mastering advanced topics.

Furthermore, the role of scaffolding extends beyond content delivery to include strategic guidance and error anticipation. Cognitive biases, such as the curse of knowledge, can hinder instructors from effectively addressing student misconceptions. By anticipating common errors and explicitly acknowledging them in my teaching, I aim to preempt confusion and provide a clear, flexible framework for students to engage with new material. For example, when introducing time integration in molecular simulations, I use relatable analogies—like the iterative process of creating a stop-motion animation—to connect abstract computational methods with tangible experiences. Such analogies not only make complex topics accessible but also align with the constructivist principle of building on familiar ideas to facilitate learning.

Constructivism also recognizes the importance of challenge and exploration in the learning process. Authentic, situated learning occurs when students confront real-world problems that push the boundaries of their current understanding. I incorporate “learning scenarios” into each module to connect the material to pressing societal problems. For instance, I frame genomics topics like sequencing and genome assembly in the context of identifying antibiotic resistance in bacterial strains. In computational structural biology, I use the challenge of designing new antibiotics targeting dihydrofolate reductase as a central theme. These scenarios provide students with a tangible purpose for their learning.

While it is crucial to ground material in concepts that students already understand, creating opportunities for intellectual growth requires carefully balanced challenges. Recognizing that many students face significant pressures to achieve high grades for medical school, graduate programs, internships, and more, I have adapted a grading model that aligns with these realities. In this model, the easier questions on homework and exams (which assess fundamental concepts and essential understanding) are worth the most points. Meanwhile, challenging questions are worth fewer points, allowing students to push their boundaries and explore advanced topics without the anxiety of jeopardizing their grades. This approach fosters an environment where students feel supported and empowered to take intellectual risks, ensuring their engagement and growth.

Inquiry-based learning

Effective teaching invites students into the discovery process, and inquiry-based learning is a powerful approach for fostering this engagement. Inquiry-based learning transforms education into an active, meaningful exploration by placing students at the heart of real-world problems requiring application, analysis, and synthesis of knowledge. This is particularly impactful in computational biology, where practical exercises demonstrate how theoretical principles translate into biological insights.

My courses implement inquiry-based learning through real-world projects that immerse students in solving contemporary scientific challenges. For instance, one project addresses antibiotic resistance by targeting *Staphylococcus aureus* dihydrofolate reductase (DHFR) through protein-ligand docking. Using tools developed by the Durrant lab, such as MolModa, students explore how ligands interact with DHFR in varying structural environments and evaluate the impact of cofactors such as NADPH on binding efficacy. By analyzing docking results in the context of experimental data, they assess the limitations of computational predictions and gain a nuanced understanding of structure-based drug design.

Another project involves protein structure prediction, where students critically evaluate models generated by advanced methods like AlphaFold3, I-TASSER, and SWISS-MODEL. They compare predicted structures to experimental data, exploring conformational differences near key binding pockets and identifying implications for subsequent drug design studies. This task cultivates their ability to analyze structural data, interpret molecular interactions, and appreciate the intricacies of protein modeling.

By incorporating open-ended projects into my courses, I cultivate a learning environment where students develop self-driven research skills and see the direct relevance of their work to societal challenges. These projects bridge classroom learning with career readiness, equipping students with practical expertise in cutting-edge tools and methods. Presenting their findings hones their communication skills, preparing them for professional roles in academia, industry, or beyond. Thus, inquiry-based learning becomes not just an educational strategy but a transformative experience, empowering students to address pressing scientific problems with confidence and creativity.

Agile education

Teaching in the rapidly evolving field of computational biology demands an approach that mirrors its pace of innovation. Revolutionary tools like AlphaFold have redefined our understanding of protein structure prediction, emphasizing the importance of curricula that adapt to such advancements. My teaching philosophy centers on equipping students with the foundational knowledge and adaptability required to excel in this dynamic landscape.

To bridge the gap between traditional education and industry expectations, I leverage multiple strategies to track developments in the pharmaceutical and biotechnology sectors. This includes analyzing trends in job postings, monitoring key announcements (e.g., NVIDIA's new AI Tech Community initiative in Pittsburgh), market trend reports and analysis, networking, and staying updated on emerging tools and methodologies. By incorporating these insights into my teaching, I ensure that my courses remain aligned with professional standards and reflect current demands in computational biology. For example, I discuss actual job postings in class to illustrate the skills and career paths most relevant to the field. This context-driven approach motivates students to engage deeply with the material and helps them understand its practical significance.

To provide foundational and advanced expertise, I expose students to cutting-edge tools like high-performance programming languages (e.g., Rust and Mojo), machine learning frameworks like PyTorch and scikit-learn, and collaborative platforms like GitHub. These tools enhance technical competence and expose students to the practices of interdisciplinary collaboration that are critical in today's professional environments. Comparing emerging protein prediction models like AlphaFold 3 with traditional approaches like I-TASSER further enhances students' understanding of the field's progression, fostering technical proficiency and critical insight.

My teaching approach focuses on iterative improvement and feedback. I regularly solicit student input to refine my teaching methods and adapt course content to meet their needs. By doing so, I aim to create an inclusive and responsive learning environment accommodating students from diverse backgrounds and learning styles. This iterative process ensures that my courses remain relevant and practical, reflecting both student aspirations and the broader demands of the field.

Open educational resources

I am actively developing comprehensive open educational resources (OERs) to address the limited availability of materials in computational biology. These resources aim to bridge gaps in undergraduate and graduate education by providing freely accessible, integrative materials tailored to the field's multidisciplinary demands.

To achieve this, I focus on creating interactive websites that host a variety of educational content, including lecture materials, homework assignments, projects, and supplementary readings. These websites are intentionally designed to be student-friendly under the Creative Commons CC BY-SA-NC 4.0 license. This licensing ensures the resources remain freely available while encouraging adaptation and sharing within the academic community. By adopting a copyleft license, I strive to democratize access to high-quality education, particularly for students from underrepresented or resource-limited backgrounds.

I prioritize making these resources widely accessible and keeping them up to date with the latest advancements in computational biology. The websites are repositories for static content and platforms for dynamic tools, such as interactive visualizations, code examples, and manipulable data sets. By integrating these features, I enable learners to connect theoretical knowledge with practical applications, fostering a deeper understanding and more robust skill development.

Building a supportive framework

Quality mentorship

Navigating the landscape of higher education can be daunting for students. Confusion, uncertainty, new responsibilities, and vague expectations can cloud their path, leaving many unsure how to proceed. I remember facing these challenges when I started my undergraduate studies at Western Michigan University. The advice I received initially was sparse and uninspiring: "Pick a marketable major and get good grades." It was actionable in the narrowest sense but failed to guide me in meaningfully engaging with my education.

This began to change when I met my mentor, Dr. Brian Young. His bioprocess engineering course expanded my awareness of chemical engineering and demonstrated the power of mentorship to transform a student's trajectory. One pivotal moment came when he asked me if I was familiar with the Louis Stokes Alliances for Minority Participation (LSAMP) program. I wasn't, but Dr. Young's ability to recognize my potential and connect me to this opportunity was life-changing. With his guidance, I applied, gained admission, and embarked on two years of research and mentorship that illuminated the transformative possibilities of higher education. This experience shaped the mentor I strive to be today: someone who tailors guidance to meet students where they are while preparing them for where they could go.

In my mentoring practice, I emphasize a tailored approach rooted in welcoming, compassion, and patience. Students often lack awareness of the resources and possibilities available to them. While they don't have to seize every opportunity, I aim to ensure they are prepared for the right one when it presents itself. This preparation often begins in my classroom, where I strive to be an approachable and authentic presence. By fostering a supportive and inclusive atmosphere, I encourage students to feel comfortable seeking advice or sharing their aspirations. I make myself available to guide and encourage them, helping them identify opportunities that align with their interests and goals. For example, I actively connect students to research labs that align with their academic interests. The course modules are designed to introduce students to contemporary tools and techniques, often overlapping with active research across the University. Students are encouraged to contact faculty for paid or credit-earning positions, using the skills they've developed in class as a bridge to practical applications.

Mentorship is about reciprocating Dr. Young's investment in me. It's about seeing students not as products to be churned out but as individuals with unique potential. By integrating mentorship into my teaching—whether through direct guidance, developmental advising, or fostering connections—I aim to illuminate pathways that students may not have realized were open to them, preparing them to seize opportunities and thrive when they do.

Cultivating belonging

Effective teaching transcends the simple dissemination of knowledge and skills. It requires creating an environment where students feel safe, welcome, and valued—a necessity in technically demanding fields like computational biology. I employ a two-layer approach to foster a sense of belonging to achieve this.

The first layer focuses on establishing both professional and personal connections with students. Hosting office hours in familiar, accessible spaces encourages students to seek help in a comfortable setting. Pre- and post-class discussions create informal opportunities to engage with students and build rapport. I also connect with students by sharing my favorite movie and music recommendations through Letterboxd and Spotify playlists, sparking conversations about shared interests. Discussions about Pittsburgh's best restaurants further build camaraderie, ensuring students see me as approachable and invested in their holistic well-being. Additionally, directing students to cultural navigators—peers, mentors, or staff members who can help them navigate campus resources—ensures they feel academically and socially supported.

The second layer involves tailoring assistance to meet individual needs. I take a holistic and proactive approach, regularly monitoring for subtle behavior changes (such as decreased participation, missing assignments, or irregular attendance) that might signal a student is struggling. To support this effort, I use Python scripts to analyze student trends in performance and attendance, allowing me to identify at-risk students before issues escalate. Once identified, I send a simple, personalized email to check in and ensure students know I care about their well-being. Often, this small gesture leads to meaningful conversations, during which I can better understand their challenges and guide them toward helpful resources, such as tutoring services or mental health support. This data-driven and empathetic strategy ensures that every student receives the support they need and strengthens the trust and rapport necessary for their academic and personal growth.

Another cornerstone of fostering belonging is addressing disparities in cultural capital: the knowledge, skills, and behaviors inherited from familial or community background. Students who lack certain types of cultural capital often face barriers to success in higher education. Rather than singling out specific groups, I provide additional resources for all students to enhance academic literacy. Examples include computer literacy training, detailed writing guidelines, and awareness of campus resources. This approach benefits all students while addressing gaps that might otherwise impede success.

I was fortunate to experience the transformative power of inclusive teaching firsthand. As a STEM major enrolled in a general education art course at Western Michigan University, I was initially apprehensive due to my lack of experience with drawing and painting. My instructor, Jim Hopfensperger, learned of my concerns through an introductory questionnaire. He had the entire class complete an exercise using their non-dominant hand, creating a shared starting point for everyone. This simple but thoughtful gesture immediately made me feel welcome and capable in an unfamiliar environment. It’s a memory that underscores the importance of fostering a sense of belonging in the classroom.

Teaching evaluations

The evaluations from Fall 2024 highlighted many aspects of my teaching that students appreciated, reflecting my dedication to fostering an engaging, inclusive, and rigorous learning environment. They also offered valuable insights into the strategies that resonated most effectively with students, guiding my efforts for continuous improvement.

Strengths identified by students

Students consistently praised the engaging nature of lectures and course content, describing them as entertaining, authentic, and thoughtfully structured. They noted that my lectures effectively blended modern tools and real-world examples, making the course highly relevant to their academic and career goals. Furthermore, students appreciated my ability to ground complex concepts in relatable examples, which helped demystify challenging material and provided a clear path to comprehension.

Accessibility was another frequently cited strength. Students felt comfortable approaching me with questions, both during and outside of class, and acknowledged the support I provided through office hours and email. They also valued the open-source and accessible nature of all course materials, which made learning convenient and transparent. This approach ensured that students had the resources they needed to succeed, regardless of their background or prior familiarity with computational biology.

The evaluations highlighted my efforts to foster a respectful and inclusive classroom environment, where all students felt valued and supported. My emphasis on providing useful and constructive feedback on assignments was noted as a positive influence on their learning experience. Several students expressed gratitude for the active discussions on scientific literature, which allowed them to form and refine their opinions on debated topics within computational biology, deepening their engagement with the subject matter.

Additionally, students recognized my adaptability and openness to feedback. They appreciated how I balanced implementing their suggestions while maintaining clear course objectives. This ability to adjust the course dynamically while ensuring pedagogical rigor was particularly valued. Many students remarked on my enthusiasm and passion for the subject, which they found infectious and motivating.

Areas for improvement

While the evaluations were largely positive, students provided constructive feedback highlighting areas where my teaching could improve. These suggestions offer a foundation for refining the course and enhancing its impact.

Some students wanted lecture slides to be shared before class to aid in preparation and note-taking. I plan to address this in future semesters by sharing lecture slides beforehand as the course becomes more established. This change will allow students to familiarize themselves with the content, making class discussions more productive and interactive.

A few students noted the lack of supplemental materials, especially given the challenging nature of computational biology. While my slides are intentionally concise to avoid resembling textbooks, I recognize the need for more comprehensive resources. I aim to continue writing open educational materials that supplement lectures and provide additional student support. These resources will also serve as valuable tools for review and self-paced learning.

Students found the extensive, high-stakes exams stressful and suggested incorporating more frequent, lower-stakes assessments. I plan to implement four or five 15-minute quizzes throughout the semester to reduce stress while maintaining rigor. This approach will provide students with consistent feedback on their progress and help them identify areas for improvement early on.

The exclusion of Python from the course sparked mixed reactions. Computational biology majors strongly advocated its inclusion, while others found it overwhelming. To strike a balance, I will gradually integrate Python in Spring 2025 by aligning it with fewer computational biology topics, ensuring students can learn both without being overburdened. Assignments will focus on reinforcing programming skills in tandem with conceptual learning.

Students desired more practice problems to reinforce their understanding. This is an area I intend to expand as I refine the course content and delivery over time. By creating targeted problem sets that align closely with lecture material, I can offer students opportunities to apply their knowledge in a structured manner, building their confidence and competence.

The open-ended nature of some questions and projects challenged students. While this approach encourages critical thinking and independence, I will provide clearer guidelines and examples to help students navigate these assignments when appropriate. Enhanced instructions will ensure students feel more confident tackling complex, exploratory tasks.

Some students noted delays in receiving graded homework. To address this, I plan to incorporate auto-graded Python-friendly assignments using tools like Gradescope to provide faster feedback. This will enhance the learning experience and allow students to track their progress more effectively.

A recurring theme was the perception of high expectations. To address this, I will continue leveraging input from undergraduate teaching assistants (UTAs) to ensure assessments remain fair and aligned with student preparedness. Additionally, I will communicate the purpose behind these expectations to help students appreciate the value of the challenges presented.

Example materials

BIOSC 1540

Here are some example materials from my most recent Computational Biology (BIOSC 1540) course. All other course materials can be found at pitt-biosc1540-2024f.oasci.org.

Syllabus

Semester	Fall 2024
Meeting time	Tuesdays and Thursdays from 4:00 - 5:15 pm
Location	1501 Posvar
Instructor	Alex Maldonado, PhD (he/him/his)
Email	`alex.maldonado@pitt.edu`
Office hours	Tuesdays (102 Clapp Hall) and Thursdays (315 Clapp Hall) from 11:30 am - 12:30 pm

Catalog description

This course gives students a broad understanding of how computational approaches can solve problems in biology. We will also explore the biological and computational underpinnings of the methods.

Prerequisites

To enroll, you must have received a minimum grade of `C` in Foundations of Biology 2 (BIOSC 0160, 0165, or 0716).

Outcomes

There are only two required courses in the Bachelor's degree in computational biology where learning computational biology is the focus. After this course, students must take computational genomics (BIOSC 1542) or simulation and modeling (BIOSC 1544). Outcomes for this course are geared towards introducing students to both subfields.

1. Understand the fundamental concepts and methodologies in genomics, transcriptomics, and computational structural biology.
2. Recognize the applications of computational methods in various biological fields.
3. Critically evaluate the strengths and limitations of different computational approaches in biology.
4. Interpret and analyze results from various computational biology methods.
5. Demonstrate enhanced problem-solving and critical thinking skills in the context of computational biology.

Schedule

We start with bioinformatics to learn fundamentals of genomics and transcriptomics. Afterwards, we will cover structure prediction, molecular simulations, and computer-aided drug design in our computational structural biology module. Our third module in scientific python will introduce how we can use Python in computational biology.

Module 01 - Bioinformatics

Week 1

Tuesday (Aug 27): Introduction to computational biology.

Thursday (Aug 29): DNA sequencing technologies.

Week 2

Tuesday (Sep 3): Quality control.

Thursday (Sep 5): De novo genome assembly.

Week 3

Tuesday (Sep 10): Gene annotation.

Thursday (Sep 12): Sequence alignment.

Week 4

Tuesday (Sep 17): Introduction to transcriptomics.

Thursday (Sep 19): Read mapping.

Week 5

Tuesday (Sep 24): Gene expression quantification.

Thursday (Sep 26): Differential gene expression.

Week 6

Tuesday (Oct 1): Review

Thursday (Oct 3): Exam 1

Module 02 - Computational structural biology

Week 7

Tuesday (Oct 8): Structural biology and determination.

Thursday (Oct 10): Protein structure prediction.

Week 8

Tuesday (Oct 15) **Fall break (no class)**

Thursday (Oct 17): Molecular simulation principles.

Week 9

Tuesday (Oct 22): Molecular system representations.

Thursday (Oct 24): Gaining atomistic insights.

Week 10

Tuesday (Oct 29): Structure-based drug design.

Thursday (Oct 31): Docking and virtual screening.

Week 11

Tuesday (Nov 5) Election day: Go vote (no class)

Thursday (Nov 7): Ligand-based drug design.

Week 12

Tuesday (Nov 12) Review

Thursday (Nov 14): Exam 2

Module 03 - Scientific python

These are optional lectures that will not negatively affect your final grade.

Week 13

Tuesday (Nov 19): Python basics

Thursday (Nov 21): Arrays and plotting

Thanksgiving break

No class on Nov 26 and 28.

Week 15

Tuesday (Dec 3): Predictive modeling

Thursday (Dec 5): Project work

Week 16

Tuesday (Dec 10): Project work

Final

Monday (Dec 16): An optional, cumulative final exam will be offered to replace lowest exam grade.

Distribution

The course will have the following point distribution.

- **Exams** (35%)
- **Project** (30%)
- **Assignments** (35%)

Late assignments and extensions

I am mindful of the diverse nature of deadlines, particularly in the scientific realm. Some are set in stone, while others exhibit more flexibility. It is noteworthy that the scientific community frequently submits manuscripts and reviews days, weeks, or months after the editor's request. Such practices are widely understood. Conversely, submitting a grant application even a minute past the deadline makes it ineligible for review.

I will use the following late assignment and extension policy. It encourages timely submissions while acknowledging the influence of external commitments and unforeseen circumstances.

- Each assignment has a specified due date and time.
- Assignments submitted after the due date will incur a late penalty.
- The late penalty is calculated using the function: % Penalty = 0.01 (1.4 x hours late)2</sup> rounded to the nearest tenth. This results in approximately:

Hours late	Penalty
6	0.7%
12	2.8%
24	11.3%
48	45.2%
72	100.0%

- Assignments will not be accepted more than 72 hours (3 days) after the due date.
- The penalty is applied to the assignment's total possible points. For example, if an assignment is worth 100 points and is submitted 36 hours late, the penalty would be approximately 13 points.
- To reward punctuality, each assignment submitted on time will earn a 2% bonus on that assignment's score.
- These on-time bonuses will accumulate throughout the semester and will be added to your final course grade.

Exceptions to this policy will be made on a case-by-case basis for extenuating circumstances. Please communicate with me as early as possible if you anticipate difficulties meeting a deadline.

Submitting your assignments on time can earn you up to a 2% bonus added to your final grade. For each assignment you submit on time, you will receive a proportional percentage of this bonus. For example, if you submit 7 out of 8 assignments on time, you would receive a 1.75% boost to your final grade. This bonus can help improve your final grade, but please note that I will not round up final grades.

Missed exams

Attendance and participation in all scheduled exams are crucial for your success in this course. To accommodate unforeseen circumstances while maintaining academic standards, please be aware of the following policy:

Due to time constraints and the difficulty of creating equivalent assessments, makeup exams will only be offered under the same circumstances. Instead, an optional cumulative final exam will be available at the end of the course. This final exam can replace your lowest midterm exam grade if it benefits your overall score. If the final exam score is lower than your existing midterm grades, it will not be counted, ensuring it cannot negatively impact your grade.

If you miss one of the midterm exams, you are highly encouraged to take the cumulative final exam. The score you achieve on the final exam will replace the missed midterm exam grade in calculating your final grade. Missing both midterm exams poses a significant challenge to course completion; in such cases, you must contact me immediately to discuss your situation and explore possible options.

Scale

Letter grades for this course will be assigned based on Pitt’s recommended scale (shown below).

Letter grade	Percentage	GPA
A +	97.0 - 100.0%	4.00
A	93.0 - 96.9%	4.00
A –	90.0 - 92.9%	3.75
B +	87.0 - 89.9%	3.25
B	83.0 - 86.9%	3.00
B –	80.0 - 82.9%	2.75
C +	77.0 - 79.9%	2.25
C	73.0 - 76.9%	2.20
C –	70.0 - 72.9%	1.75
D +	67.0 - 69.9%	1.25
D	63.0 - 66.9%	1.00
D –	60.0 - 62.9%	0.75
F	0.0 - 59.9%	0.00

Attendance

While attendance is not mandatory, active participation is strongly encouraged and integral to your course success. Regular class attendance offers several benefits:

1. Enhanced understanding of course material through engaging discussions;
2. Opportunities for collaborative learning through group activities;
3. Practical application of concepts via hands-on exercises.

Please note that while it is possible to achieve the maximum number of points without attending lectures, consistent attendance demonstrates commitment to the course. This commitment may be considered when addressing individual circumstances or requests for flexibility.

Policies

Generative AI

We are in an exciting area of generative AI development with the release of tools such as ChatGPT, DALL-E, GitHub Copilot, Bing Chat, Bard, Copy.ai, and many more. This course will permit these tools' ethical and responsible use except when explicitly noted. For example, you can use these tools as an on-demand tutor by explaining complex topics.

Other ways are undoubtedly possible, but any use should aid—not replace—your learning. You must also be aware of the following aspects of generative AI.

- **AI limitations:** While AI programs can be valuable resources, they may produce inaccurate, biased, or incomplete material. Each program has its unique limitations as well.
- **Bias and accuracy:** Scrutinizing each aspect of these enormous data sets used to train these products is infeasible. AI will inherit biases and inaccuracies from these sources and human influences in fine-tuning. You must be critical and skeptical of anything generated from these models and verify information from trusted sources.
- **Critical thinking:** Understand that AI is a tool, not a replacement for your analysis and critical thinking skills. AI to enhance your understanding and productivity, but remember that your development as a scholar depends on your ability to engage independently with the material.
- **Academic integrity:** Plagiarism extends to content generated by AI. Using AI-generated material without proper attribution is a violation of academic integrity policies. Always give credit to AI-generated content and adhere to citation rules.

  Furthermore, text from AI tools should be treated as someone else's work—because it is. You should never copy and paste text directly.
- **AI detection:** As discussed here, the University Center for Teaching and Learning does not recommend using AI detection tools like turnitin due to high false positive rates. I will not use AI detection tools in any capacity for this course and trust that you will use these tools responsibly when permitted and desired.

Remember that generative AI is helpful when used responsibly. You can ethically benefit from these technological advances by adhering to these guidelines. Embrace this opportunity to expand your skill set and engage thoughtfully with emerging technologies. If you have any questions about AI tool usage, please get in touch with me for clarification and guidance.

BIOSC 1540 - Computational Biology

Final Exam

Dec 16, 2024

100 points

Please read the following instructions carefully before beginning your assessment.

- **Time limit:** You have 150 minutes to complete and turn in this assessment.
- **Open note:** You may use notes, but with the following restrictions:
  - Notes must be hand-written on either (1) paper or (2) a tablet with a stylus, then printed.
  - You may use a maximum of one sheet of 8.5 × 11 in. paper for notes (front and back allowed).
  - Notes must be your own work. Sharing or copying notes from others is strictly prohibited.
  - Your name must be clearly written on your notes.
- **No digital devices:** The use of digital devices, including calculators, is not allowed.
- **Submission requirements:** You must submit both your completed assessment and all notes.

I agree to follow the above instructions. I affirm that all work on this assessment will be my own and that I will not give or receive any unauthorized assistance. To have your assessment graded, you must write your name, sign, and provide your student ID below.

Name

Signature

Student ID

Recommendation: More challenging problems are worth fewer points, so answer the easier problems first. Choose the best answer for each problem unless specified otherwise.

Problem 1

Build a De Bruijn graph with k_{edge} = 3 with all the following reads: GGATT, GATTA, TACAG, AGATT, TACCG. Using your De Bruijn graph, write down the optimal contig (there is only one). Assume that you can only walk along each edge up to two times.

(4 points)

Problem 2

In RNA-seq analysis, what is the most significant problem caused by adapter contamination?

(2 points)

Ⓐ False splice junctions during transcript mapping.

Ⓑ Reduction in overall sequencing depth.

Ⓒ Bias in GC content calculations.

Ⓓ Alteration of read quality scores.

Problem 3

What key advantage does RPKM (Reads Per Kilobase Million) provide in RNA-seq analysis?

(3 points)

Ⓐ It corrects for sequence-specific biases in PCR amplification.

Ⓑ It adjusts for differences in RNA degradation between samples.

Ⓒ It enables direct comparison of expression levels between different gene lengths.

Ⓓ It normalizes for differences in GC content between genes.

Problem 4

How does increasing sequencing coverage from 10x to 30x most significantly impact variant calling in genomic analysis?

(2 points)

Ⓐ It increases the maximum length of insertions that can be detected.

Ⓑ It improves confidence in identifying heterozygous variants.

Ⓒ It reduces the time required for computational analysis.

Ⓓ It enables detection of more complex structural variants.

Problem 5

In de Bruijn graph construction for genome assembly, what is the relationship between k-mer size and graph complexity?

(1 point)

Ⓐ Larger k-mers result in simpler graphs with fewer branching points but require higher coverage.

Ⓑ Smaller k-mers produce simpler graphs and require less coverage for accurate assembly.

Ⓒ K-mer size has no effect on graph complexity, which is determined solely by genome size.

Ⓓ Larger k-mers always produce more accurate assemblies regardless of coverage.

Problem 6

During genome assembly graph traversal, which strategy would most likely lead to inefficient or incorrect assembly?

(1 point)

Ⓐ Breaking cycles in the graph by identifying repeat regions.

Ⓑ Implementing backtracking when encountering branching points in the graph.

Ⓒ Starting traversal from nodes with high coverage and extending in both directions.

Ⓓ Using a depth-first random walk without considering coverage information.

Problem 7

Which feature of prokaryotic genes most significantly influences the accuracy of computational open reading frame (ORF) detection?

(4 points)

Ⓐ The presence of well-defined ribosomal binding sites upstream of start codons.

Ⓑ The length distribution of intergenic regions.

Ⓒ The presence of transcription termination sequences.

Ⓓ The GC content of coding regions.

Problem 8

In RNA-seq analysis, which scenarios would require a negative binomial model rather than a simpler Poisson model? Select all that apply.

(1 point)

Ⓐ When biological replicates show higher variance than expected from sampling alone.

Ⓑ When samples are sequenced at different depths.

Ⓒ When samples come from different experimental batches.

Ⓓ When analyzing differential expression between conditions.

Ⓔ When analyzing technical replicates from the same sample.

Problem 9

What is the main advantage of Salmon's online phase in transcript quantification?

(2 points)

Ⓐ It performs complete alignment of all reads against the transcriptome.

Ⓑ It eliminates the need for further refinement in abundance estimation.

Ⓒ It identifies all possible splice variants for each gene.

Ⓓ It provides rapid initial abundance estimates.

Problem 10

Why do Sanger sequencing reads typically show higher quality scores at the middle positions compared to both ends of the read?

(3 points)

Ⓐ DNA polymerase has higher accuracy in synthesizing medium-length fragments.

Ⓑ Base-calling algorithms are optimized for the middle of reads.

Ⓒ The middle positions have more balanced representation of fragment lengths in the reaction.

Ⓓ PCR amplification is more efficient for medium-length fragments.

Problem 11

In which scenario would protein threading be more likely to succeed than homology modeling?

(2 points)

Ⓐ When the protein has 20% sequence identity but conserved fold patterns with known structures.

Ⓑ When the protein shares 60% sequence identity with a known structure.

Ⓒ When the protein contains multiple domains with varying levels of conservation.

Ⓓ When crystal structures exist for close homologs of the target protein.

Problem 12

What makes correlated mutations in protein sequences valuable for predicting three-dimensional protein structure?

(3 points)

Ⓐ They determine the evolutionary age of specific protein domains.

Ⓑ They reveal which amino acids are most conserved across species.

Ⓒ They predict the rate of protein folding in different cellular environments.

Ⓓ They identify pairs of residues that maintain physical contact.

Problem 13

Perform a Smith-Waterman alignment with the following sequences: `GCATATACGC` and `TCGTAGCTA`. Use a scoring scheme of 1 for match, −2 for mismatch, and −1 for gap. Show all possible tracebacks and their respective alignments.

(4 points)

		G	C	A	T	A	T	A	C	G	C
T											
C											
G											
T											
A											
G											
C											
T											
A											

Problem 14

When developing a new force field for protein simulations, which approach would provide the most reliable parameter validation?

(2 points)

Ⓐ Comparing protein-ligand binding free energies with experimental measurements.

Ⓑ Testing if the force field can fold a protein from a random coil.

Ⓒ Measuring how well the force field reproduces quantum mechanical energies.

Ⓓ Counting how many hydrogen bonds form during a simulation.

Problem 15

Which challenge best explains why multiple types of scoring functions are often used together in molecular docking?

(3 points)

Ⓐ Individual scoring functions are too computationally expensive to use alone.

Ⓑ Different scoring functions capture complementary aspects of protein-ligand binding.

Ⓒ Single scoring functions cannot handle different protein structures.

Ⓓ Using multiple scoring functions increases the speed of virtual screening.

Problem 16

During a protein-ligand binding simulation, a researcher observes that important binding events are too rare to study effectively. Which simulation approach would be most appropriate to address this?

(3 points)

Ⓐ Change to a different force field parameter set.

Ⓑ Increase the simulation box size to include more solvent molecules.

Ⓒ Reduce the simulation temperature to slow molecular motions.

Ⓓ Sample along a collective variable between bound and unbound states.

Problem 17

Why is the minimum image convention essential when implementing periodic boundary conditions in molecular dynamics simulations?

(2 points)

Ⓐ It prevents particles from interacting with multiple copies of themselves.

Ⓑ It reduces the computational cost by eliminating the need for calculating forces for all atoms.

Ⓒ It allows particles to move freely between different simulation boxes.

Ⓓ It increases the accuracy of long-range electrostatic calculations.

Problem 18

Why does X-ray crystallography require multiple protein molecules arranged in a crystal lattice rather than a single protein molecule?

(4 points)

Ⓐ Individual protein molecules move too quickly to be analyzed by X-rays.

Ⓑ The crystal structure prevents radiation damage to the protein.

Ⓒ The regular arrangement of proteins amplifies weak X-ray scattering signals to detectable levels.

Ⓓ Crystallization removes water molecules that interfere with X-ray diffraction.

Problem 19

What typically happens if a molecular dynamics simulation uses a time step that is too large relative to the fastest molecular motions?

(3 points)

Ⓐ During equilibration, the simulation will adjust the time step to maintain stability.

Ⓑ The simulation will have unrealistic atomic movements and energy conservation violations.

Ⓒ The simulation slows down to compensate for the large time step.

Ⓓ The simulation runs more efficiently and sample more conformations.

Problem 20

Why are Fourier series particularly well-suited for modeling dihedral angle potentials in force fields?
(2 points)

Ⓐ They naturally capture the periodic nature of rotation around chemical bonds.

Ⓑ They require fewer computational resources than other mathematical functions.

Ⓒ They allow for direct incorporation of quantum mechanical data.

Ⓓ They automatically adjust to different types of chemical bonds.

Problem 21

In molecular dynamics simulations, why are multiple independent simulation runs considered more statistically robust than a single long simulation?
(2 points)

Ⓐ They allow for parallel processing of different initial molecular configurations, reducing overall computational time.

Ⓑ Multiple runs provide a more comprehensive sampling of the system's conformational space by exploring different initial microstates.

Ⓒ Independent runs enable direct comparison of simulation outcomes to identify systematic biases in the computational method.

Ⓓ They provide redundant data points that can be averaged to reduce statistical noise in the simulation results.

Problem 22

In molecular binding processes, entropy is best described as:
(3 points)

Ⓐ A static property that determines molecular interactions based on molecular size and shape.

Ⓑ A measure of thermal energy transfer between molecules during complex formation.

Ⓒ The quantitative change in molecular degrees of freedom upon binding.

Ⓓ An exclusively enthalpic phenomenon that predicts the stability of molecular complexes.

Problem 23

In computational drug discovery, the primary goal of generating molecular fingerprints through hashing is to:

(1 point)

Ⓐ Simulate molecular interactions through mathematical transformations.

Ⓑ Generate unique identifiers that perfectly capture a molecule's three-dimensional structure.

Ⓒ Compress complex molecular structural information into a computationally manageable format.

Ⓓ Standardize molecular representations for rapid computational comparison.

Problem 24

In molecular binding processes, Gibbs free energy (ΔG_{bind}) fundamentally represents:

(4 points)

Ⓐ The total energy required to initiate molecular interactions under standard conditions.

Ⓑ The maximum work that can be extracted from a molecular binding process at constant temperature and pressure.

Ⓒ A measure of the spontaneity and energetic favorability of molecular association.

Ⓓ The precise mechanical work needed to overcome intermolecular repulsive forces.

Problem 25

In molecular interactions, which type of noncovalent force typically provides the most important energetic contribution to specific molecular recognition?

(1 point)

Ⓐ Van der Waals interactions that depend on temporary electron density fluctuations.

Ⓑ Specialized chemical interactions that form precise, directional patterns.

Ⓒ Charge-based interactions that create long-range attraction between molecular partners.

Ⓓ Quantum mechanical coupling effects between molecular electronic structures.

Problem 26

In computational molecular simulations, the representation of chemical bonds as a mechanical model primarily aims to:

(3 points)

Ⓐ Provide a computationally efficient approximation of molecular bond dynamics.

Ⓑ Capture the exact quantum mechanical behavior of electron interactions.

Ⓒ Simulate the complete breaking and reformation of chemical bonds during interactions.

Ⓓ Replicate the precise vibrational modes of molecular structures.

Problem 27

In structural biology, constructing an accurate protein model from experimental electron density data involves:

(3 points)

Ⓐ Systematically comparing theoretical atomic models with observed experimental data.

Ⓑ Using computational algorithms to predict protein folding based on sequence information.

Ⓒ Manually tracing electron density contours to determine molecular structure.

Ⓓ Generating multiple independent structural models to capture protein variability.

Problem 28

Given the Burrows-Wheeler Transform (BWT) `TGAT$AGAG`, determine the original string. Show all of your work.

(4 points)

Problem 29

Perform the Burrows-Wheeler Transform (BWT) of the string `TAGTGAGA`. Show all intermediate steps.

(4 points)

Problem 30

In transcriptomics, the fragment assignment matrix represents a critical step in:

(2 points)

- Ⓐ Precisely determining the genomic origin of sequencing fragments with deterministic mapping.
- Ⓑ Probabilistically resolving fragment compatibility across multiple potential transcripts.
- Ⓒ Generating a comprehensive catalog of all possible transcript variants.
- Ⓓ Calculating absolute fragment count distributions for each transcript.

Problem 31

In transcriptomics, a generative model's primary purpose is to:

(2 points)

Ⓐ Create a predictive framework for experimental design.

Ⓑ Develop a comprehensive mapping of transcript diversity.

Ⓒ Reconstruct the precise molecular pathway of RNA synthesis.

Ⓓ Simulate the probabilistic process of RNA sequencing fragment production.

Problem 32

In molecular biology research, the primary purpose of assessing RNA sample quality involves:

(4 points)

Ⓐ Quantifying the molecular characteristics that impact downstream experimental reliability.

Ⓑ Determining the potential for accurate gene expression measurements.

Ⓒ Establishing the biochemical potential of RNA molecules for experimental use.

Ⓓ Identifying the structural stability of RNA for long-term storage.

Problem 33

In genomic research, functional annotation primarily aims to:

(4 points)

Ⓐ Classify genes based on their evolutionary conservation patterns.

Ⓑ Determine the structural characteristics of genomic regions.

Ⓒ Predict the associated molecular interactions and cellular processes.

Ⓓ Quantify the expression levels of newly identified genes.

Problem 34

In genome assembly algorithms, which parameter would have the least direct impact on determining the most reliable contig path?

(4 points)

Ⓐ Computational resource requirements for path exploration.

Ⓑ Diversity of sequencing reads contributing to the path.

Ⓒ Statistical confidence in path connectivity.

Ⓓ Potential for introducing sequencing artifacts.

Problem 35

Why do ddNTPs cause chain termination in DNA synthesis while dNTPs allow continued elongation?
(4 points)

Ⓐ ddNTPs lack the 3' hydroxyl group necessary for forming the next phosphodiester bond in DNA elongation.

Ⓑ ddNTPs form weaker hydrogen bonds with template DNA, causing the polymerase to release the growing strand.

Ⓒ ddNTPs change the conformation of DNA polymerase, preventing it from adding more nucleotides.

Ⓓ ddNTPs block the binding site for the next incoming nucleotide, physically preventing further additions.

Problem 36

In Illumina sequencing, what happens to DNA fragments that lack properly ligated adapters?
(4 points)

Ⓐ They bind to the flow cell but cannot form clusters due to incomplete bridge formation.

Ⓑ They fail to bind to the flow cell surface and are washed away during bridge amplification.

Ⓒ They form clusters but cannot be sequenced due to missing primer binding sites.

Ⓓ They produce weak signals during sequencing due to reduced fluorescent marker incorporation.

Genome assembly project

You have been assigned a set of Illumina sequencing reads from a *Staphylococcus aureus* isolate. Your task is to assemble and annotate the genome, and then identify the dihydrofolate reductase gene, which is a common target for antibiotics. This project will give you experience with real-world bioinformatics tools and workflows used in genomics research.

Learning objectives

By completing this project, you will:

1. Gain practical experience with a complete genome assembly and annotation workflow.
2. Learn to use and interpret results from key bioinformatics tools.
3. Understand the process of identifying potential drug targets from genomic data.
4. Develop skills in data analysis and interpretation in a genomics context.

Materials

You will be assigned one of 114 whole-genome sequencing runs from the BioProject PRJNA741582. Each run contains Illumina HiSeq 2000 paired-end reads of 250 nucleotides in length. Go to Canvas to find your SRA accession number.

Instructions

You will use a series of bioinformatics tools to process and analyze your assigned sequencing data. Follow these steps carefully, documenting your process and results at each stage.

Setting up your Galaxy project

Before beginning the analysis, you need to set up an account on Galaxy and create a project for this assignment. Galaxy is a web-based platform for data-intensive biomedical research that will allow you to perform all the necessary analyses for this project.

1. Go to https://usegalaxy.org.
2. Click on the “Login or Register” button in the top middle.
3. If you don’t have an account, click on “Register here” and fill out the registration form. Make sure to use a valid email address as you’ll need to verify it.
4. Once you’ve registered and logged in, click “Create new history+” and name your new project “BIOSC 1540 Project” and provide a brief description if you wish. Click `Create`.

Always ensure you’re working within your “BIOSC 1540 Project” for this assignment. This will help you keep your work organized and easily accessible.

Galaxy will also save your work automatically, but it’s a good practice to regularly check that your analyses are being saved correctly.

If you encounter any issues with Galaxy, check their support and training.

Getting data

In this step, you'll download the assigned sequencing data for your *S. aureus* isolate and upload it to Galaxy. Follow these instructions carefully:

1. Your instructor will provide you with a unique SRA (Sequence Read Archive) accession number for your assigned *S. aureus* isolate. We will use `SRX11246059` as an example.
2. In the Galaxy interface, click on `Tools` in the left sidebar.
3. In the search bar at the top of the Tools panel, type `Download and Extract Reads in FASTQ format from NCBI SRA` and click on it.
4. In the tool interface:
   - For `select input type`, choose `SRR accession`.
   - In the `Accession` field, enter your assigned SRA accession number.
   - Leave all other settings as default.
5. Click `Run Tool`.

Galaxy will now download and extract your sequencing data. You can monitor the progress in the right sidebar, and will turn from blanched almond to green once finished.

1. Once complete, you should see two new items in your history:
   - `Single-end data (fastq-dump)`
   - `Paired-end data (fastq-dump)`
2. There will be no data in `Single-end data (fastq-dump)`, so you can delete this.
3. To find the `forward.fastqc` file, you need to view job information for `Paired-end data (fastq-dump)`, click on your accession number (i.e., `SRX11246059`), then on `forward` to preview its contents. You should see sequences in FASTQ format.

Include the following information:

- Your SRA accession number that starts with `SRX`.
- The first five FASTQ entries in the `forward.fastq` file.

Quality control with FastQC

Quality control is a crucial step in any sequencing data analysis. It helps you identify any issues with your sequencing data that might affect downstream analyses. We'll use FastQC, a widely used tool for quality control of high throughput sequencing data.

1. In the Galaxy interface, search for "FastQC" in the tools panel.
2. In the tool interface:
   - For `Raw read data from your current history`, select `Dataset Collection` and choose `Paired-end data (fastq-dump)`.
   - Leave all other settings as default.
3. Click `Run Tool`.

Once the job is complete, you'll see new items in your history for each FastQC report (Webpage and RawData).

Click on the `FastQC on collection: Webpage`, your accession number, and then both `forward` and `reverse` eye icons. The report contains several modules, each assessing a different aspect of your sequence data.

In your report, address the following questions for your `forward` reads:

1. What is the overall quality of your sequencing data? Provide evidence from the FastQC report (i.e., screenshots) to support your assessment.
2. Are there any concerning issues identified by FastQC? If so, what are they and how might they impact your downstream analyses?
3. Based on the "Per base sequence quality" plot, is there evidence of quality degradation towards the end of the reads? If so, how might this inform your approach to read trimming?
4. Do you see any evidence of adapter contamination or overrepresented sequences? If so, what steps might you take to address this?
5. How do the results for your forward reads compare to your reverse reads? Are there any notable differences?

Remember, some warnings or failures in the FastQC report don't necessarily mean your data is unusable. The context of your experiment and the specific analyses you plan to perform should guide your interpretation of these results.

Adapter trimming with fastp

After assessing the quality of your raw sequencing data, the next step is to trim adapters and perform quality control. We'll use fastp, a fast all-in-one preprocessing tool for FASTQ files.

1. In the Galaxy interface, search for "fastp" in the tools panel.
2. Click on "fastp: fast all-in-one preprocessing for FASTQ files".
3. In the tool interface, set the following parameters:
    - "Single-end or paired reads": Select "Paired Collection".
    - "Select paired collection(s)": Choose your paired-end reads.
4. Click "Run Tool".

Click on the eye icon next to the HTML report to view it. The report contains information about the trimming and filtering process.

In your report, address the following questions with either your fastp or FastQC report:

1. How many reads were removed during the filtering process? What percentage of reads passed the filters?
2. Did fastp detect and trim any adapters? If so, which ones?
3. How did the quality scores change after filtering? Provide specific examples from the report.
4. Were there any issues with base composition (e.g., GC bias) before or after filtering?

5. What was the duplication rate? Is this what you would expect for your type of sequencing data?
6. Compare the fastp results to your earlier FastQC results. How do they complement each other? Are there any discrepancies?

Remember, the goal of this step is to improve the overall quality of your sequencing data by removing low-quality reads, trimming adapters, and addressing any other issues identified in the FastQC step. The fastp results should show an improvement in data quality compared to the raw reads.

De novo genome assembly with SPAdes

After trimming and quality control, the next step is to assemble the processed reads into contigs. We'll use SPAdes, a versatile genome assembler designed for both small genomes and single-cell projects.

1. In the Galaxy interface, search for "SPAdes" in the tools panel.
2. Click on "SPAdes genome assembler for genomes of regular and single-cell projects".
3. In the tool interface, set the following parameters:
   - "Operation mode": Select "Assembly and error correction"
   - "Single-end or paired-end short-reads": Choose "Paired-end: list of dataset pairs"
   - "FASTA/FASTQ file(s)": Select your paired-end output from the fastp step
4. Click "Run Tool".

Once complete, you'll see new items in your history, including the assembly graph, contigs, and scaffolds.

To get more detailed statistics about your assembly, we'll use the Bandage Info tool:

1. In the Galaxy interface, search for `Bandage Info` in the tools panel.
2. Click on `Bandage Info`.
3. In the tool interface, set the following parameters:
   - `Graphical Fragment Assembly`: Select the `SPAdes: Assembly Graph with Scaffolds`.
   - Leave all other settings as default.
4. Click "Run Tool".
5. Once complete, you'll see a new item in your history with the Bandage Info output.
6. Next, repeat the same process with `Bandage Image` and include this in your report.

In your report, address the following questions:

1. How many contigs (i.e., nodes) were produced by the assembly?
2. What is the total length of the assembly? How does this compare to the expected genome size of Staphylococcus aureus (approximately 2,800,000 base bp)? (Note: You can also use quast to check this number.)
3. Calculate the difference between the "Total length" and "Total length no overlaps". What does this difference represent, and why is it important to consider?

4. What is the N50 of your assembly? How does this value compare to the median node length? What does this tell you about the distribution of contig sizes in your assembly?
5. Analyze the “dead ends” in your assembly. How many dead ends are there? What percentage of the total possible ends (2 * number of nodes) do these represent? What might a high percentage of dead ends indicate about your assembly?
6. Based on all these results, how would you assess the overall quality of your genome assembly? Consider factors such as completeness, contiguity, and potential issues.
7. How do you think the “careful” mode in SPAdes might havfinale influenced these assembly statistics compared to the default mode?

Gene annotation with Prokka

After assembling the genome, the next step is to annotate it to identify genes and their potential functions. We’ll use Prokka, a rapid prokaryotic genome annotation tool.

1. In the Galaxy interface, search for `Prokka` in the tools panel.
2. Click on `Prokka: Prokaryotic genome annotation`.
3. In the tool interface, set the following parameters:
   - “Contigs to annotate”: Select the Scaffolds file from your SPAdes output
   - “Genus name”: Enter “Staphylococcus”
   - “Species name”: Enter “aureus”
4. Click “Run Tool”.

After the annotation is complete, examine the output files. Pay particular attention to:

- **GFF3 file (.gff)**: This contains the annotations in a standard format. You can use this file to visualize the annotations in a genome browser.
- **GenBank file (.gbk)**: This file contains both the sequence and the annotations. It can be viewed in many sequence analysis tools.
- **Statistics file (.txt)**: This provides a summary of the annotation process. Look at the number and types of features identified.
- **Protein FASTA file (.faa)**: This contains the amino acid sequences of predicted proteins. You’ll use this to find your gene of interest (dihydrofolate reductase).

In your report, address the following questions:

1. How many coding sequences (CDS) did Prokka identify in your genome?
2. How many rRNA and tRNA genes were annotated?
3. What other types of features did Prokka identify, and how many of each?
4. How does the number of annotated genes compare to what you would expect for a *Staphylococcus aureus* genome?
5. Search the `.faa` file for the dihydrofolate reductase gene. Did Prokka identify this gene? If so, what is its protein ID? What is the amino acid sequence of your protein?

6. Based on these results, how would you assess the quality and completeness of your genome annotation?
7. How might the genus and species names you provided influence the annotation process?

Remember, a typical *S. aureus* genome contains around 2,500 to 2,900 genes. The presence or absence of certain genes can provide insights into the strain's potential characteristics or capabilities.

Comparing to UniProt

After annotating your genome and identifying the dihydrofolate reductase gene, the next step is to compare your sequence to a known reference. This comparison will help you verify your annotation and identify any potential variations in your sequence.

1. Go to the UniProt website (https://www.uniprot.org/).
2. Find the dihydrofolate reductase protein (folA gene) entry for *Staphylococcus aureus* (Taxon ID `1280`).

In your report, address the following questions:

1. What is the UniProt ID?
2. What is the length of your protein sequence? How does it compare to the UniProt reference (159 amino acids)?
3. Calculate the percentage identity between your sequence and the UniProt reference. Are they identical, or are there differences?
4. If there are differences, list their positions and the amino acid changes. Are these changes conservative (similar amino acids) or non-conservative?

Docking project

In the ongoing battle against antibiotic-resistant bacteria, understanding the molecular interactions between drugs and their targets is crucial. *Staphylococcus aureus* dihydrofolate reductase (DHFR) is a well-established target for several antibiotics due to its essential role in bacterial folate synthesis. This project focuses on protein-ligand docking studies of *S. aureus* DHFR using MolModa, a powerful tool for virtual screening and molecular modeling.

You will perform docking simulations to predict how various ligands bind to DHFR within different protein environments. By comparing these predictions with experimental data, you will assess the reliability of computational docking as a method for identifying potential antibiotic candidates. Additionally, you will explore the impact of cofactors like NADPH on ligand binding, providing insights into the complexities of protein-ligand interactions in a cellular context.

Learning objectives

By completing this project, you will:

1. Comprehend the principles and significance of molecular docking in drug discovery.
2. Navigate and utilize MolModa for preparing proteins and ligands for docking.
3. Configure and execute docking simulations with appropriate parameters.
4. Compare computational docking scores with experimental binding affinities (IC50, Ki, MIC).
5. Assess the impact of cofactors, such as NADPH, on ligand binding and docking outcomes.
6. Evaluate the correlation between docking predictions and experimental data.
7. Generate and interpret visual representations of ligand poses and binding interactions.
8. Analyze how decoys are scored and the implications for screening accuracy.

Instructions

We will be using MolModa to perform protein-ligand docking of *S. aureus* DHFR in the following proteins.

PDB ID	**Additional ligand**
3FRD	Folate
6PRA	None
3FRD	OWS

MolModa, unfortunately, cannot dock in the presence of cofactors (e.g. NADPH), which could impact results of our virtual screening. Thus, the instructor will provide docking results with the NADPH cofactor while you use MolModa to examine the impact of its removal.

We recommend that you use Google Chrome. Some browsers (e.g., Firefox) have been shown to exhibit more bugs and not work smoothly.

Protein preparation

Here are the general steps in MolModa to prepare the protein for docking.

1. Download this specific `3FRD` PDB structure (click this link) and import it into MolModa using `File` -> `Open`.
2. Remove all non-polymer atoms (e.g., water molecules and all co-crystallized ligands) from the protein structure.
3. Protonate the protein at a pH of 7.4 (the default).
4. Add a region with a center of (-9, 34, −4) and dimensions of (26, 25, 27).

Ligand preparation

Here are the general steps in MolModa to prepare the ligands for docking.

1. Download the active molecules' SMILES and load them into MolModa by using `File` -> `Open` or by drag and drop.
2. Protonate all compounds at a pH of 7.4 while regenerating coordinates with the "recommended" 3D coordinates generation option.

Docking

Dock the protonated compounds into the `3FRD` protein using an exhaustiveness of `24` in MolModa. These are your docking results for the active compounds without NADPH.

You need to download your docked ligand poses by going to `File` -> `Save`, uncheck "Save in MolModa format", and use these settings.q

This will download a zip file that you can extract and have all of your docked poses as `mol2` files. You can load these compounds into PyMOL.

In your submission, answer the following questions:

1. Download and extract on your computer these docking results that were performed in the presence of NADPH using AutoDock Vina. In PyMOL, load the `3frd.pdb` structure and all active `.pdbqt` files in the `poses/` directory. These `.pdbqt` files contain the top 9 poses of the identified binding modes. To see their scores, look at the respective log file in the `logs/` directory. Choose any three active compounds and compare these poses to your results—you can download your ligand poses from MolModa by clicking `File` -> `Save` and unchecking "Save project in .molmoda format". Be sure to look at more than just the top pose from the NADPH results. Use screenshots to justify your observations.

Analysis

As mentioned in most CSB lectures, the purpose of docking is to be able to identify potentially active compounds out of the massively large chemical space. Below is a table from Muddala et al. that elucidates the binding affinity and antibacterial properties of novel ligands to *S. aureus* DHFR.

Label	Inhibitor	Average IC50 ± SEM (nM)	Average Ki ± SEM (nM)	MIC (µg/mL)
11a	CHEMBL2042372	6.3 ± 0.3	1.2 ± 0.1	0.0625-0.25
11b	CHEMBL2042373	5.5 ± 1.0	1.1 ± 0.2	0.0469-0.0938
11c	CHEMBL2042374	8.7 ± 2.1	1.7 ± 0.4	0.0938-0.1875
11d	CHEMBL2042375	8.2 ± 0.5	1.6 ± 0.1	0.0938-0.25
11e	CHEMBL2042376	9.2 ± 0.5	1.8 ± 0.1	0.125-0.25
11f	CHEMBL2042377	13.1 ± 3.1	2.6 ± 0.6	0.5
11g	CHEMBL551038	4.6 ± 0.7	0.9 ± 0.1	0.1875-0.375
11h	CHEMBL565020	4.1 ± 0.4	0.8 ± 0.1	0.125-0.25
11i	CHEMBL2042475	5.8 ± 0.3	1.1 ± 0.1	0.125-0.25
11j	CHEMBL550083	8.4 ± 1.7	1.7 ± 0.3	0.0469-0.1875
11k	CHEMBL2042476	7.2 ± 0.8	1.4 ± 0.2	0.25
11l	CHEMBL2042477	4.8 ± 0.8	0.9 ± 0.2	0.125
11m	CHEMBL2042478	5.7 ± 0.6	1.1 ± 0.1	0.125-0.375
11n	CHEMBL2042479	7.0 ± 0.3	1.4 ± 0.1	0.25-0.5
11o	CHEMBL2042480	11.6 ± 2.2	2.3 ± 0.4	0.5
11p	CHEMBL2042481	6.3 ± 1.9	1.2 ± 0.4	0.5
11q	CHEMBL2042482	9.6 ± 0.5	1.9 ± 0.1	1
12a	N/A	7.7 ± 2.9	1.5 ± 0.6	1
12b	CHEMBL4799218	5.8 ± 3.3	1.1 ± 0.7	1
12c	CHEMBL4781757	3.3 ± 1.7	0.7 ± 0.3	1
12d	N/A	6.7 ± 3.9	1.3 ± 0.8	0.5-1
12e	N/A	6.3 ± 2.8	1.2 ± 0.6	0.5
12f	N/A	5.6 ± 2.8	1.1 ± 0.6	0.5

12g	N/A	4.5 ± 2.6	0.9 ± 0.5	1
12h	N/A	4.5 ± 2.2	0.9 ± 0.4	1
12i	N/A	5.4 ± 3.1	1.1 ± 0.6	0.5
12j	N/A	5.2 ± 2.3	1.0 ± 0.5	0.5
13a	N/A	8.4 ± 3.8	1.7 ± 0.7	0.5-1
13b	CHEMBL4790501	6.1 ± 2.7	1.2 ± 0.5	0.5-1
13c	N/A	5.2 ± 3.0	1.0 ± 0.6	0.5
13d	N/A	7.2 ± 3.2	1.4 ± 0.6	0.5
13e	N/A	7.4 ± 4.2	1.5 ± 0.8	0.5
13f	N/A	6.1 ± 2.7	1.2 ± 0.5	0.5

Your instructor has docked all of these active ligands (in addition to the canonical reactant, DFH) and 100 randomly selected decoys from FDA approved drugs into `3FRD`, `6PRA`, and `6PR6`. You can download the results here: 3FRD-results.zip, 6PRA-results.zip, 6PR6-results.zip.

In your submission, answer the following questions:

1. What is the meaning/significance of IC50, Ki, and MIC? For an active molecule, what are the ideal values for these properties?
2. `3FRD` is a crystal structure with NADPH and DHF, which should be an optimal protein structure to design a DHF competitive inhibitor. Is there a correlation between AutoDock Vina docking scores (with NADPH) and experimental Ki values? Quantify this correlation with an R^2 value.
3. `6PRA` is a crystal structure that does not have DHF. Explain how this could impact docking calculations. Compare the docking scores of active ligands from `6PRA` and `3FRD`; are there significant differences?
4. `6PR6` was crystallized with the active compound `11j`. Compare the crystallized ligand pose to the top 9 poses and scores of the same `11j` ligand. What do you observe? Support your claims with screenshots.
5. Decoy ligands provide a mechanism to ensure that virtual screening workflows work as expected. Using the `3FRD` results, examine the decoy scores and poses with the top three best and worst scores. Does our pipeline accurately sort these decoy molecules? Why or why not? Examine the top poses of these best and worst decoys. Are there any noncovalent interaction trends? Support your claims with screenshots.

Assignment 07

Be concise and focus on critical concepts. For each question or subpart, your response should be between 50 and 100 words.

Q01

Identifying the correct protein target is a crucial step in drug development.

a) Explain why this identification is so important.

b) List and describe two criteria that make a protein a suitable drug target.

Q02

Gibbs free energy plays a significant role in the binding of a protein to a ligand.

a) Describe this role.

b) How do enthalpic and entropic contributions influence the binding affinity between a protein and a ligand?

Q03

Noncovalent interactions contribute to the binding enthalpy in protein-ligand complexes.

a) List and briefly describe four types of these interactions.

b) Explain why these interactions are essential for binding specificity.

Q04

Entropy influences the binding of a ligand to a protein.

a) Explain how entropy affects this process.

b) Discuss why considering entropy merely as “disorder” is an oversimplification in the context of protein-ligand binding.

Q05

Binding pocket detection is important in molecular docking studies.

a) Describe the importance of detecting binding pockets.

b) What are the different types of binding sites, and how do they differ from each other?

Q06

There are differences between systematic and stochastic search algorithms used in molecular docking.

a) Explain these differences.

b) Why is stochastic sampling often preferred for larger molecules in docking studies?

Q07

Ligand pose optimization is a critical step in docking studies.

a) Define ligand pose optimization.

b) Why is it critical for accurately predicting binding affinity?

Q08

Discuss how data-driven approaches, such as machine learning, can improve the efficiency and accuracy of virtual screening in drug discovery.

Q09

Scoring functions are essential in molecular docking.

a) Explain their purpose.

b) How do machine learning-based scoring functions differ from traditional physics-based methods? Provide examples of advantages and limitations of each approach.

Q10

Alchemical simulations are used to estimate the free energy of binding between a protein and a ligand.

a) Describe how these simulations work.

b) Explain why these simulations are computationally expensive, and discuss scenarios where they might be preferred over simpler docking methods.

Q11

Thermodynamic integration is a concept used in alchemical simulations.

a) Explain this concept.

b) How does thermodynamic integration contribute to estimating free energy differences between bound and unbound states in protein-ligand interactions?

BIOSC 1630

Here are some example materials from my most recent Computational Biology Seminar (BIOSC 1630) course. All other course materials can be found at pitt-biosc1630-2024f.oasci.org.

Syllabus

Semester	Fall 2024
Meeting time	Wednesdays from 1:00 - 3:30 pm
Location	302 Cathedral of Learning
Instructor	Alex Maldonado, PhD (he/him/his)
Email	`alex.maldonado@pitt.edu`
Office hours	No routine office hours will be provided. Meetings can be scheduled on a case-by-case basis.

Catalog description

Topics in computational biology will be explored using primary literature. Students will present research articles orally, as well as complete a series of writing assignments that will culminate in the production of a literature review paper.

Prerequisites

You must be a Junior or Senior on the CBUAS-BS and CBUSCI-BS plan. You must also have completed BIOSC 1540 with a minimum grade of C **and** one of the following courses:

- ENG 0102;
- ENGCMP (0002 or 0006 or 0020 or 0200 or 0203 or 0205 or 0207 or 0208 or 0210 or 0212);
- ENGFLM 0210;
- FP (0003 or 0006).

Writing intensive

BIOSC 1630 will fulfill the requirement for a writing-intensive course in the Computational Biology major. The Dietrich School of Arts & Sciences Writing Institute defines this as:

- a course in which students engage with writing substantively throughout the term;
- they write and revise throughout the term (not just at the end);
- they write a total of 5750–6250 words;
- they get feedback from their teacher and their peers.

Five hundred words is approximately one page single-spaced and two pages double-spaced. You are expected to write 11–13 single-spaced pages or 23–25 double-spaced pages in this course.

Outcomes

After successfully completing this course, students should be able to do the following.

1. Efficiently search for and identify relevant scientific literature.
2. Effectively summarize scientific literature’s motivation, methods, and critical findings.
3. Critically evaluate the robustness and validity of methods and analyses.
4. Interpret and draw meaningful conclusions from computational data.
5. Assess the transparency, reproducibility, and adherence to open science practices.
6. Understand the interplay between computational and experimental corroboration.
7. Communicate scientific ideas and data effectively through clear, concise, and well-structured writing.
8. Deliver engaging and informative presentations on scientific literature.

Distribution

The course will have the following point distribution.

- **Paper**
  - **Theme analysis**: 6%
  - **Literature review**: 8%
  - **Introduction**: 6%
  - **Field overview**: 12%
  - **Analysis**: 12%
  - **Draft**: 8%
  - **Final draft**: 18%
- **Pre-class assignments**: 10%
- **Activities**: 20%

Late assignments and extensions

I am mindful of the diverse nature of deadlines, particularly in the scientific realm. Some are set in stone, while others exhibit more flexibility. It is noteworthy that the scientific community frequently submits manuscripts and reviews days, weeks, or months after the editor’s request. Such practices are widely understood. Conversely, submitting a grant application even a minute past the deadline makes it ineligible for review.

I will use the following late assignment and extension policy. It encourages timely submissions while acknowledging the influence of external commitments and unforeseen circumstances.

- Each assignment has a specified due date and time.

- Assignments submitted after the due date will incur a late penalty.

- The late penalty is calculated using the function: % Penalty = 0.01 (1.4 x hours late)2</sup> rounded to the nearest tenth. This results in approximately:

Hours late	Penalty
6	0.7%
12	2.8%
24	11.3%
48	45.2%
72	100.0%

- Assignments will not be accepted more than 72 hours (3 days) after the due date.
- The penalty is applied to the assignment's total possible points. For example, if an assignment is worth 100 points and is submitted 36 hours late, the penalty would be approximately 13 points.
- To reward punctuality, each assignment submitted on time will earn a 2% bonus on that assignment's score.
- These on-time bonuses will accumulate throughout the semester and will be added to your final course grade.

Exceptions to this policy will be made on a case-by-case basis for extenuating circumstances. Please communicate with me as early as possible if you anticipate difficulties meeting a deadline.

Submitting your assignments on time can earn you up to a 2% bonus added to your final grade. For each assignment you submit on time, you will receive a proportional percentage of this bonus. For example, if you submit 7 out of 8 assignments on time, you would receive a 1.75% boost to your final grade. This bonus can help improve your final grade, but please note that I will not round up final grades.

Scale

Letter grades for this course will be assigned based on Pitt's recommended scale (shown below).

Letter grade	Percentage	GPA
A +	97.0 - 100.0%	4.00
A	93.0 - 96.9%	4.00
A –	90.0 - 92.9%	3.75
B +	87.0 - 89.9%	3.25
B	83.0 - 86.9%	3.00
B –	80.0 - 82.9%	2.75
C +	77.0 - 79.9%	2.25
C	73.0 - 76.9%	2.20
C –	70.0 - 72.9%	1.75
D +	67.0 - 69.9%	1.25
D	63.0 - 66.9%	1.00

D –	60.0 - 62.9%	0.75
F	0.0 - 59.9%	0.00

Policies

Generative AI

We are in an exciting area of generative AI development with the release of tools such as ChatGPT, DALL-E, GitHub Copilot, Bing Chat, Bard, Copy.ai, and many more. This course will permit these tools' ethical and responsible use except when explicitly noted. For example, you can use these tools as an on-demand tutor by explaining complex topics.

Other ways are undoubtedly possible, but any use should aid—not replace—your learning. You must also be aware of the following aspects of generative AI.

- **AI limitations:** While AI programs can be valuable resources, they may produce inaccurate, biased, or incomplete material. Each program has its unique limitations as well.
- **Bias and accuracy:** Scrutinizing each aspect of these enormous data sets used to train these products is infeasible. AI will inherit biases and inaccuracies from these sources and human influences in fine-tuning. You must be critical and skeptical of anything generated from these models and verify information from trusted sources.
- **Critical thinking:** Understand that AI is a tool, not a replacement for your analysis and critical thinking skills. AI to enhance your understanding and productivity, but remember that your development as a scholar depends on your ability to engage independently with the material.
- **Academic integrity:** Plagiarism extends to content generated by AI. Using AI-generated material without proper attribution is a violation of academic integrity policies. Always give credit to AI-generated content and adhere to citation rules.

  Furthermore, text from AI tools should be treated as someone else's work—because it is. You should never copy and paste text directly.
- **AI detection:** As discussed here, the University Center for Teaching and Learning does not recommend using AI detection tools like turnitin due to high false positive rates. I will not use AI detection tools in any capacity for this course and trust that you will use these tools responsibly when permitted and desired.

Remember that generative AI is helpful when used responsibly. You can ethically benefit from these technological advances by adhering to these guidelines. Embrace this opportunity to expand your skill set and engage thoughtfully with emerging technologies. If you have any questions about AI tool usage, please get in touch with me for clarification and guidance.

Perspective

In this course, you will write a perspective article on a critical debate in computational biology. This assignment aims to develop your ability to analyze complex scientific issues, form well-reasoned arguments, and articulate your viewpoint supported by current research. You will:

- Choose one of the provided perspective primers, each presenting a nuanced question in computational biology.
- Write a detailed, well-supported perspective on your chosen topic (5-7 pages, excluding cover page, references, tables, and figures).

Objective

Your task is to research your chosen topic thoroughly, present a balanced argument considering multiple viewpoints, and provide a well-reasoned perspective supported by current scientific evidence. This paper should demonstrate your critical thinking skills and ability to synthesize complex information in computational biology.

Key components

While you have flexibility in structuring your paper, consider including these elements:

1. **Introduction:** Set the context, highlight the topic's significance, and present your main argument.
2. **Background:** Provide necessary scientific context and key concepts.
3. **Current State of Research:** Summarize relevant findings and identify trends or conflicts in the literature.
4. **Your Perspective:** Present your viewpoint, supported by evidence from your research.
5. **Implications and Future Directions:** Discuss the potential impact of your perspective and suggest areas for further research.
6. **Conclusion:** Synthesize your main points and reinforce your stance.

Remember to maintain a scholarly tone while engaging your audience of peers and instructors who have foundational knowledge of computational biology.

Evaluation criteria

Your paper will be evaluated based on:

- Depth of research and understanding of the topic
- Quality and relevance of supporting evidence
- Clarity and coherence of your argument
- Critical analysis and original thinking
- Adherence to academic writing standards and provided guidelines

We encourage you to approach this assignment as an opportunity to contribute meaningfully to ongoing debates in computational biology. Good luck with your research and writing!

Primers

You must choose one of the following perspective primers and write a detailed, well-supported perspective on the topic. Each primer presents a nuanced question with no clear right or wrong answer, encouraging you to explore the literature, form your own opinion, and justify your stance. Your task is to research your chosen topic thoroughly, present a balanced argument, and provide a well-reasoned perspective supported by current scientific evidence.

Protein structure prediction

Primer: Are *ab initio* protein structure prediction algorithms still relevant in the deep learning era?

Ab initio protein structure prediction algorithms can determine the three-dimensional structures of proteins from their amino acid sequences without relying on homologous structures. These methods often involve intensive computational processes and can be time-consuming. However, recent advances in deep learning, exemplified by tools like AlphaFold, have dramatically improved the accuracy and efficiency of protein structure predictions, challenging the relevance of traditional *ab initio* approaches.

In the era of deep learning, especially with sophisticated models that leverage vast amounts of data and computational power, protein structure prediction has seen unprecedented advancements. The question arises: do *ab initio* methods still hold value, or have they been rendered obsolete by these newer, data-driven approaches? This perspective should touch on the balance between traditional algorithmic approaches and cutting-edge machine learning techniques and their implications for the future of computational structural biology.

Possible discussion points:

- **Accuracy and Reliability:** Compare the accuracy and reliability of *ab initio* methods with deep learning-based predictions. Evaluate situations where one method may outperform the other.
- **Computational Resources:** Assess the computational demands of *ab initio* methods versus deep learning models, considering accessibility for different research institutions.
- **Data Dependence:** Discuss the dependence of deep learning models on large datasets and the potential limitations this may impose compared to *ab initio* methods, which do not exclusively rely on prior data.
- **Innovation and Integration:** Explore how traditional *ab initio* methods can be integrated with deep learning approaches to enhance prediction accuracy and reliability.
- **Case Studies:** Examine specific case studies where *ab initio* methods have provided unique insights or deep learning models have significantly outperformed traditional approaches.
- **Future Prospects:** Consider the future of protein structure prediction, including potential advancements in *ab initio* and deep learning methods and their implications for the field.

One can argue for the continued relevance of *ab initio* methods based on their foundational principles, independence from large training datasets, and potential for integration with new technologies.

Conversely, others may emphasize deep learning's transformative impact, highlighting its superior accuracy, efficiency, and the paradigm shift in the field.

Here are some scientific articles to help get you started.

Primary

- Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., … & Jumper, J. M. (2024). Accurate structure prediction of biomolecular interactions with AlphaFold3. *Nature*, 1-3. DOI: 10.1038/s41586-024-07487-w
- Baek, M., DiMaio, F., Anishchenko, I., Dauparas, J., Ovchinnikov, S., Lee, G. R., … & Baker, D. (2021). Accurate prediction of protein structures and interactions using a three-track neural network. *Science, 373*(6557), 871-876. DOI: 10.1126/science.abj8754
- Zhou, X., Zheng, W., Li, Y., Pearce, R., Zhang, C., Bell, E. W., … & Zhang, Y. (2022). I-TASSER-MTD: a deep-learning-based platform for multi-domain protein structure and function prediction. *Nature Protocols, 17*(10), 2326-2353. DOI: 10.1038/s41596-022-00728-0

Opinion

- Outeiral, C., Nissley, D. A., & Deane, C. M. (2022). Current structure predictors are not learning the physics of protein folding. *Bioinformatics, 38*(7), 1881-1887. DOI: 10.1093/bioinformatics/btab881
- Kuhlman, B., & Bradley, P. (2019). Advances in protein structure prediction and design. *Nature reviews molecular cell biology, 20*(11), 681-697. DOI: 10.1038/s41580-019-0163-x
- Doga, H., Raubenolt, B., Cumbo, F., Joshi, J., DiFilippo, F. P., Qin, J., … & Shehab, O. (2024). A perspective on protein structure prediction using quantum computers. *Journal of Chemical Theory and Computation, 20*(9), 3359-3378. DOI: 10.1021/acs.jctc.4c00067

Reviews

- Huang, B., Kong, L., Wang, C., Ju, F., Zhang, Q., Zhu, J., … & Bu, D. (2023). Protein structure prediction: challenges, advances, and the shift of research paradigms. *Genomics, Proteomics & Bioinformatics, 21*(5), 913-925. DOI: 10.1016/j.gpb.2022.11.014
- Bertoline, L. M., Lima, A. N., Krieger, J. E., & Teixeira, S. K. (2023). Before and after AlphaFold2: An overview of protein structure prediction. *Frontiers in bioinformatics, 3*, 1120370. DOI: 10.3389/fbinf.2023.1120370

Computer-aided drug design

Primer: Are molecular dynamics simulations overhyped in drug discovery, or do they provide indispensable insights?

Molecular dynamics (MD) simulations allow researchers to observe the behavior of molecules over time, offering detailed insights into the dynamic nature of protein-ligand interactions. This technique is often used after initial docking studies to refine and validate the predicted interactions. However, MD simulations are computationally intensive and require significant expertise to interpret.

MD simulations are typically performed after initial docking studies in the drug development pipeline to validate and refine the predicted protein-ligand interactions. The question arises: Should researchers invest in computationally expensive and time-consuming MD simulations or proceed directly to wet-lab experiments, which might provide more definitive answers? This decision point is critical, as it impacts the drug development process's efficiency, accuracy, and cost.

Possible discussion points:

- **Accuracy and Precision:** Debate the accuracy of MD simulations in predicting real-world molecular interactions compared to static docking models.
- **Computational Resources:** Consider the computational costs and accessibility of MD simulations for different research institutions.
- **Predictive Value:** Evaluate how MD simulations can refine docking results and their impact on predicting binding affinities and interaction stability.
- **Experimental Validation:** Discuss whether the insights gained from MD simulations justify the delay and resources compared to proceeding directly to wet lab experiments after docking.
- **Case Studies:** Examine specific case studies in which MD simulations have either provided critical insights or been unnecessary in the drug design process.
- **Future Prospects:** Discuss potential advancements in MD technology and their implications for future drug design, considering both the benefits and limitations.

MD simulations could be indispensable because they can provide detailed dynamic insights and refine docking predictions, enhancing the reliability of subsequent wet lab experiments. Conversely, others might highlight the practical challenges, such as the computational expense and the potential delays in the drug development timeline, advocating for a more streamlined approach that moves directly from docking to experimental validation.

Here are some scientific articles to help get you started.

Primary

- Alibay, I., Magarkar, A., Seeliger, D., & Biggin, P. C. (2022). Evaluating the use of absolute binding free energy in the fragment optimisation process. *Communications Chemistry, 5*(1), 105. DOI: 10.1038/s42004-022-00721-4
- Eberhardt, J., Santos-Martins, D., Tillack, A. F., & Forli, S. (2021). AutoDock Vina 1.2.0: New docking methods, expanded force field, and python bindings. *Journal of chemical information and modeling, 61*(8), 3891-3898. DOI: 10.1021/acs.jcim.1c00203
- Wan, S., Sinclair, R. C., & Coveney, P. V. (2021). Uncertainty quantification in classical molecular dynamics. *Philosophical Transactions of the Royal Society A, 379*(2197), 20200082. DOI: 10.1098/rsta.2020.0082
- Sahakyan, H. (2021). Improving virtual screening results with MM/GBSA and MM/PBSA rescoring. *Journal of Computer-Aided Molecular Design, 35*(6), 731-736. DOI: 10.1007/s10822-021-00389-3

- Lee, T. S., Lin, Z., Allen, B. K., Lin, C., Radak, B. K., Tao, Y., … & York, D. M. (2020). Improved alchemical free energy calculations with optimized smoothstep softcore potentials. *Journal of chemical theory and computation, 16*(9), 5512-5525. DOI: 10.1021/acs.jctc.0c00237

Opinion

- Song, L. F., & Merz Jr, K. M. (2020). Evolution of alchemical free energy methods in drug discovery. *Journal of Chemical Information and Modeling, 60*(11), 5308-5318. DOI: 10.1021/acs.jcim.0c00547

Reviews

- Sabe, V. T., Ntombela, T., Jhamba, L. A., Maguire, G. E., Govender, T., Naicker, T., & Kruger, H. G. (2021). Current trends in computer aided drug design and a highlight of drugs discovered via computational techniques: A review. *European Journal of Medicinal Chemistry, 224*, 113705. DOI: 10.1016/j.ejmech.2021.113705
- Bassani, D., & Moro, S. (2023). Past, present, and future perspectives on computer-aided drug design methodologies. *Molecules, 28*(9), 3906. DOI: 10.3390/molecules28093906
- Yang, C., Chen, E. A., & Zhang, Y. (2022). Protein–ligand docking in the machine-learning era. *Molecules, 27*(14), 4568. DOI: 10.3390/molecules27144568
- Dhakal, A., McKay, C., Tanner, J. J., & Cheng, J. (2022). Artificial intelligence in the prediction of protein–ligand interactions: recent advances and future directions. *Briefings in Bioinformatics, 23*(1), bbab476. DOI: 10.1093/bib/bbab476
- Sadybekov, A. V., & Katritch, V. (2023). Computational approaches streamlining drug discovery. *Nature, 616*(7958), 673-685. DOI: 10.1038/s41586-023-05905-z

Analysis assignment

Write the “Analysis” section of your perspective article, articulating and defending your unique viewpoint on your chosen topic.

Instructions

1. **Length:** Approximately 1 to 1.5 pages.
2. **Content requirements:**
   - Clearly articulate your perspective or argument
   - Provide evidence and reasoning to support your viewpoint
   - Address potential counterarguments or limitations
   - Discuss the implications and significance of your perspective
3. **Writing style:**
   - Use clear, precise language appropriate for an academic audience
   - Maintain an objective tone while presenting your argument
   - Use proper citations in APA format

Rubric

Your analysis section will be evaluated based on the following criteria.

1. **Clarity of perspective (30 points)**
   - 30-25: Perspective is exceptionally clear, specific, and relevant to computational biology
   - 24-19: Perspective is clear and relevant, with minor areas for improvement in specificity
   - 18-13: Perspective is somewhat clear but may lack specificity or relevance
   - 12-0: Perspective is unclear, overly broad, or not relevant to computational biology
2. **Evidence and reasoning (25 points)**
   - 25-21: Exceptional use of evidence and reasoning, with a strong connection to the perspective
   - 20-16: Good use of evidence and reasoning, with clear connections to the perspective
   - 15-11: Adequate evidence and reasoning, but connections to the perspective may be weak
   - 10-0: Insufficient or irrelevant evidence, poor reasoning
3. **Counterarguments (20 points)**
   - 20-17: Thorough and thoughtful address of potential counterarguments or limitations
   - 16-13: Good consideration of counterarguments, with minor areas for improvement
   - 12-9: Some counterarguments addressed, but treatment may be superficial
   - 8-0: Counterarguments ignored or inadequately addressed
4. **Implications and significance (15 points)**
   - 15-13: Insightful and comprehensive discussion of implications and significance
   - 12-10: Good discussion of implications and significance, with some depth
   - 9-7: Basic discussion of implications and significance, lacking depth
   - 6-0: Little or no discussion of implications and significance
5. **Organization and flow (15 points)**
   - 15-13: Exceptional organization with seamless flow between ideas

- 12-10: Good organization and flow, with minor issues
- 9-7: Adequate organization, but flow between ideas may be choppy
- 6-0: Poor organization and flow, ideas are difficult to follow

6. **Writing quality (10 points)**
   - 10-9: Excellent writing quality with clear, concise prose and appropriate academic tone
   - 8-7: Good writing quality with minor errors or awkward phrasing
   - 6-5: Adequate writing quality, but with noticeable errors or inconsistencies in tone
   - 4-0: Poor writing quality with numerous errors or inappropriate tone
7. **Use of sources (5 points)**
   - 5: Excellent integration of relevant, peer-reviewed sources with proper citations
   - 4: Good use of sources with minor issues in integration or citation
   - 3: Adequate use of sources, but may rely too heavily on a single source or have citation errors
   - 2-0: Poor use of sources, lack of peer-reviewed sources, or major citation errors

Tips for Success

Remember, the goal is not just to state your viewpoint, but to persuade your readers of its validity and importance through careful argumentation and evidence.

- Start by clearly defining your perspective before diving into supporting evidence
- Use topic sentences to guide your reader through your argument
- Ensure each paragraph contributes directly to supporting your main perspective
- When addressing counterarguments, be fair in your representation of opposing views
- Use concrete examples or case studies to illustrate your points
- Review and revise your work, paying attention to the logical flow of your argument

Pre-class assignment

Please read the selected paper below and answer the following questions in your own words. I am only looking for a few sentences per question. Write your answers in your favorite Word processing software and submit them as a PDF to Canvas.

Paper: Champion, C., Gall, R., Ries, B., Rieder, S. R., Barros, E. P., & Riniker, S. (2023). Accelerating Alchemical Free Energy Prediction Using a Multistate Method: Application to Multiple Kinases. *Journal of Chemical Information and Modeling, 63*(22), 7133-7147. DOI: 10.1021/acs.jcim.3c01469

Do not get too bogged down by the methodology. Your instructor will go over the following topics in the respective lecture.

1. Molecular dynamics (MD) simulations:
    - Computer simulation of physical movements of atoms and molecules
    - Applications in computational biology and drug discovery
2. Force fields:
    - Set of parameters used to calculate the potential energy of a system of atoms
    - Examples mentioned in the paper: GAFF and OpenFF
3. Free energy calculations:
    - Importance in drug discovery (predicting binding affinities)
    - Brief overview of traditional methods like Free Energy Perturbation (FEP) and Thermodynamic Integration (TI)
4. Alchemical transformations:
    - Concept of “morphing” one molecule into another in silico
    - Why it’s useful for calculating relative binding free energies
5. Sampling in MD simulations:
    - Importance of exploring conformational space
    - Challenges with traditional methods (e.g., getting “stuck” in local energy minima)
6. Enhanced sampling techniques:
    - General concept of improving exploration of conformational space
    - Brief mention of replica exchange as an example

Q01

What is the main goal or purpose of this study?

Q02

The authors applied RE-EDS to four different kinase systems. List these kinases and briefly describe the types of ligand modifications studied for each.

Q03

What two small molecule force fields were used in this study? Why do you think the authors chose to use multiple force fields?

Q04

Briefly explain the concept of “hybrid topology” used in this paper. How does it differ from “dual topology”?

Q05

What metric did the authors use to assess the accuracy of their binding free energy calculations? What value is considered the threshold for “chemical accuracy”?

Q06

For which kinase system did RE-EDS perform best? For which did it perform worst? What factors may have contributed to these differences?

Q07

What were some of the key limitations or challenges identified for the RE-EDS method?

Q08

Based on the results, what are the potential applications or benefits of using RE-EDS for drug discovery?

Q09

What questions do you have about the paper?

Appendix A. Student responses to teaching evaluations

This contains a comprehensive, unedited collection of my teaching evaluations as a postdoctoral associate in the Department of Biological Sciences at the University of Pittsburgh. For a more concise summaries, please see above. Note that responses that contained "N/A" or equivalent were removed for brevity.

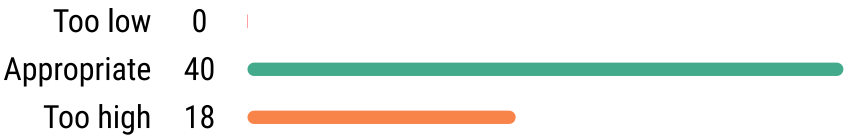
BIOSC 1540

You can find a description of this course above.

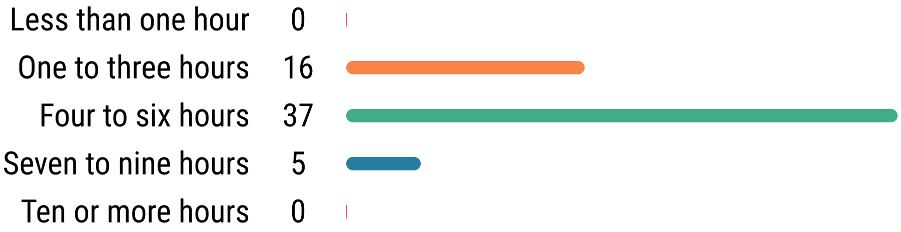
Fall 2024

I received evaluations from 58 out of 72 enrolled students, representing an 80.6% response rate.

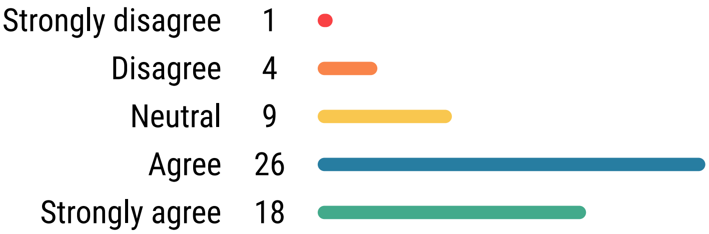
The standards the instructor set for me were



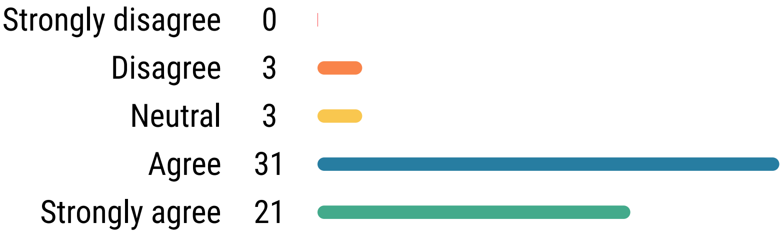
How many hours per week did you usually spend working on this course outside of classroom time?



The instructor created an atmosphere that kept me engaged in course content



The instructor was prepared for class



The instructor treated students with respect

Strongly disagree	0
Disagree	1
Neutral	2
Agree	18
Strongly agree	37

The instructor was available to me (in-person, electronically, or both)

Strongly disagree	0
Disagree	1
Neutral	8
Agree	19
Strongly agree	37
N/A	4

The instructor evaluated my work fairly

Strongly disagree	1
Disagree	5
Neutral	12
Agree	19
Strongly agree	19
N/A	2

The instructor provided feedback that was helpful to me

Strongly disagree	2
Disagree	2
Neutral	9
Agree	20
Strongly agree	22
N/A	3

I learned a lot from this course

Strongly disagree	0
Disagree	2
Neutral	6
Agree	21
Strongly agree	29

The instructor creates an inclusive learning environment for all students

Strongly disagree	0
Disagree	2
Neutral	4
Agree	18
Strongly agree	34

The course was intellectually challenging

Strongly disagree	0
Disagree	0
Neutral	3
Agree	27
Strongly agree	27

I could ask a question or make a comment if I wanted to

Strongly disagree	1
Disagree	0
Neutral	3
Agree	25
Strongly agree	28

This course helped to develop my ability to solve real problems in this field

Strongly disagree	1
Disagree	4
Neutral	5
Agree	28
Strongly agree	19

This course helped to broaden my interests

Strongly disagree	0
Disagree	5
Neutral	7
Agree	22
Strongly agree	23

Not requiring Python has positively impacted my learning experience in this course

Strongly disagree	7
Disagree	13
Neutral	17
Agree	7
Strongly agree	11

Incorporating required programming assignments would enhance my understanding of course material

Strongly disagree	1
Disagree	5
Neutral	14
Agree	22
Strongly agree	14

I feel more confident about pursuing a career in computational biology after taking this course (if desired)

Strongly disagree	3
Disagree	6
Neutral	12
Agree	24
Strongly agree	11

The skills learned in this course are relevant to my career goals

Strongly disagree	0
Disagree	2
Neutral	11
Agree	20
Strongly agree	23

This course should be split into computational biology major and non-major sections

Strongly disagree	0
Disagree	4
Neutral	14
Agree	19
Strongly agree	19

What did you like best about how the course was taught?

- The lecture slides, the method the material was taught, and the assignments.
- I liked the content that we learned in this class. It was interesting and engaging. I also liked that we were able to have hands on practice with different tools.
- I liked how the coding aspects were separate from the class content.
- The problem sets were a good way to gauge our understanding of the past two or more lectures and really delve into our basic understanding of the content.
- I liked that we learned about current bioinformatics softwares.
- You're a really nice instructor and provided entertaining lectures! I also enjoyed the layout of the grading and the bonus for assignments turned in on time, and definitely appreciated the cheat sheets for exams. The articles about each topic was a really good idea too.
- I liked the structure of the lecture slides and how he had a lot of materials posted to the course website.
- I liked that all lecture notes were available online.
- I liked how knowledgeable Dr. Maldonado was. He not only taught us well but would drop knowledge on us that inspired me to want to really learn more about the course.
- I liked that presentations had the little follow online link and a lot of topics were interesting and thoroughly discussed (or boring but extremely necessary in this major, so I had to suffer and learn).
- I liked how things were taught in class, I thought a lot of the more complicated concepts were explained well.
- Multiple aspects on Computational Biology, which is helpful for us to learn the full area of CompBio.
- I liked how the homework questions were at the end of the semester, because I feel they were tailored to help learn the content.
- I liked how Alex structured the lecture slides into subtopics and really made sure we knew the info before moving on. It is clear that Alex knows a lot and also cares a lot about his students. He teaches with enthusiasm and helps us outside of class and office hours if needed.
- I like how the weekly assignments encourage us to learn and memorize what we have learnt on time.
- Open source / accessible; different medium than canvas; openness to feedback; responsive to emails and enough office hours; no false conceptual frameworks; challenging questions.
- I thought that the material was interesting, and I liked how student feedback was taken into account and adjustments were made when needed. The grading distribution seems okay. The professor is nice and he seems like he actually cares about the students.
- The expectations were clear and the coursework was consistent with the material taught. I also liked the testing format.
- Computational structural biology.
- Instructor created very thoughtful slides and graphics that helped make concepts easier to understand. Instructor also created engaging lectures and provided a lot of additional resources for understanding subject matter.

- I thought that Dr. Maldonado was really good at keeping things down–to–earth, and did a good job of making connections between more difficult concepts and real–world examples/well–known examples that helped me understand things more.
- I thought the lectures did a good job with deep dives into specific computational biology "ideas", such as with BWT or classical approximations for MD sims. I thought it gave a very good idea for how they worked, while maintaining the right depth considering the pre–reqs for the course. I also enjoyed how the course covered so many varied topics that combined a large number of other classes I have taken (or will take) together. I just found it fun when I could talk about stat mech with friends in modern physics because of a comp bio course I took.
- I liked how the learning objectives were outlined for every lecture, which helped guide my studying.
- I liked the lecture style and how we could follow along with the slides on our own computers. I also liked how questions could be asked over TopHat anonymously.
- I liked the slides and when they were available I especially liked the other sources he would include in the section about the lecture we just did.
- I appreciated the way he tried to break down different difficult concepts. I do think there was a better way to kind of tie in the computational aspect of things, or at least the use of different packages and databases, that wasn't directly related to programming.
- The slides were will done so that all the information was on it in way that wasn't overwhelming. Also the lecture was done in a way that was both informative and not–boring.
- I liked the bonus for turning stuff in on time; I also liked the website with the information. It was much easier to navigate and have lots of information on compared to canvas. I liked the grading scale as well. It made it so that one bad exam wouldn't bomb your grade. Grading projects, tests, and assignments are basically the same weight allowed students to diversify their skills.
- I liked that this course was challenging and made us work for the grades. Dr. Maldonado is also very helpful and encouraging to students and incorporates new developments in the field into the course. I also appreciate how there were multiple opportunities to increase the grade, as this course is very difficult for someone first learning about these topics.
- The slides that you provided and your website were set up really nicely. I appreciated all of the resources you provided. The HW though took much time, really helped me understand the concepts better.
- Professor made the course so that everything we learned would apply to how we would use information in workplace.
- Alex's class is incredibly engaging, and he has a knack for simplifying complex concepts. Attending his office hours is highly recommended. The class introduces both bioinformatics and computational biology criteria, providing me with a comprehensive understanding of each. This knowledge helps me identify which criteria resonate with me the most and aligns with my future goals. The class introduces many novel concepts that I believe are very useful for my future work. It solidifies my passion for computational biology, and I had a lot of fun.
- The slides are helpful, but that is it.

- Even though a lot of the concepts seem to be a lot more complicated, Dr. Maldonado did a great job of explaining the content and going into more detail or using analogies to help explain some of these. He was also very accessible, and office hours were extremely helpful. I also like how there was background reading for the lectures just to help understand the content better.
- The lectures were interesting and engaging, but I would have liked them beforehand to prepare for the lecture.
- I liked the "flipped" classroom approach when that was a thing. I find it extremely helpful to prime to my brain before anything and it was especially helpful for complex topics this course introduces. I also think the grading system is pretty lenient. I liked all the theory it introduces as well, I feel like I learned a lot about the field and who the big players are in it, alongside what they're doing and what they're using to accomplish their advancements. I thought the review guides for the exam were also extremely helpful and effectively guided my preparation. I also appreciated that demonstration of tools during lecture. I'm also excited to do the final project, it seems extremely interesting and educational.
- Very organized.
- I liked the simulation part of the course.
- Office hour available.
- I like how the hw was based on concepts in class.
- I liked how the homework helped me understand the course material even if it wasn't so clear after lecture.
- What I liked best about this course is it allowed me to understand computational biology and what it truly is as a field since it was something I was interested in. Coming into the course I knew it would be challenging and I can say that the instructor explained the concepts in a way that made sense. I could comprehend coming from a field of limited knowledge of algorithms and other computer science-based aspects of the course.
- I didn't like much about how the course was taught by this professor. It was hard to retain concepts, and there were no supplemental material to turn to if you were having trouble. This was a problem since the slides were so bare-bones.
- The material was interesting and Dr. Maldonado clearly had a deep understanding of the material.
- I liked that there were opportunities to utilize computational tools.
- I like that the lectures built off of one another, and that the PDFs were made available immediately following the lecture. There was always time to ask questions, and the content was straight forward.
- i liked the exam format and homeworks
- The way Alex teaches this course is fun and engaging. For the first time in all four years of my Biology career at Pitt I felt excited to come to class and learn new material. A good teacher really does make all the difference.
- I enjoyed the lecture portion of the course and found Dr. Maldonado to be very informative.
- Taught based on what we felt we needed more or less help with.

- I honestly enjoyed how the tests were formatted. By changing the points to be aligned with question difficulty, it made it less pressure while still making sure we understood the content.
- Dr. Maldonado masterfully made complex concepts easy to digest. His personable and approachable attitude made coming to class a delight rather than a slog, and combined with a lecturing style employing active student participation, visualizations, and comparisons to concepts we understand outside of class (like comparing micro state–based entropy to a coordinate system with points), made understanding difficult material much easier.

  Dr. Maldonado consistently exposed us to the bigger picture, discussing how class concepts were used in labs around Pitt (also encouraging students to get into research!) and where the field may go in the future. This kept me engaged with the content, making it feel far more tangible, and by discussing real–would applications, improved my understanding of class concepts.

  Additionally, Dr. Maldonado gave outside–of–class opportunities for students wanting to practice more coding–related skills relevant to computational biology — this increased accessibility and engagement across the various student demographics in the class (majors who may he experienced in programming, and non–majors wanting to learn about a new field).

  He gave a comprehensive view of computational biology, leaving me feeling genuinely prepared to pursue computational biology research and excel in future major courses.
- I liked how friendly and personable the professor was with the students. He was very approachable during class and office hours and tried his best to make the course manageable despite the heavy amount of lecture content.

If you were teaching this course, what would you do differently?

- I would make the first portion of the assignment available in the first half of the semester.
- I wish the content was split differently so that exams had less content. If there was a quiz–based approach I think it would be easier to do better in the course. Also, I wish we did more Python for the people who are taking this class for their major.
- I would not do anything differently.
- I feel like the way the information is presented straight from lecture without prior reading content makes it difficult to understand in one sitting during lecture and I'm not able to fully focus when so much information that I've never seen before is getting thrown at me.
- I would include some more R and Python that would count as extra credit and would have fewer questions on the homework assignments since they were hard to finish and keep up with.
- I understand why you would keep the PDFs of the slides unavailable until after the lecture, but I found it tough to take notes effectively this way. Usually if I have a PDF of the slideshow already with me I only need to write down what the instructor says (which for this class, is a lot of important info) on top of the existing slides, but here I ended up juggling between trying to write down info from the slides, taking screenshots before I missed them (since I couldn't go back on the slides myself), and trying to remember what was being spoken. Of course this is just personal preference,

but I feel like it would be much easier and better for retaining information from lecture if we were provided the slide PDFs before/during lecture.

The exams could be a bit rough and I also felt like some of the homework was hard to answer without substantially googling or looking up the answers and sometimes it seemed to go much more in depth than the lectures. However I also understand that there's a lot of high level content to try to pack into this class and that's probably difficult.

The final project though was way beyond the scope of what we were taught in class and there were little to no resources about it online or posted on the website. Part 1 was perfectly fine, but Part 2 and 3 were incredibly tough due to lack of knowledge on the programs and files we used and with not enough guidance (there was a lot of guidance for Part 1 but suddenly very little for Part 2 and 3 which were the more complex parts), and that the questions asked (especially on Part 2 and 3) also involved way more depth and information than what we had actually learned in this course. I found myself clueless on many questions and struggling a lot on the final project!

I think overall I just wish the expectations of the course wasn't set so high especially since this course is so broad in topics, and I wish we had more guidance on assignments/projects/exams.

- I would provide more practice problems with answer keys that are similar to the exams because the exams were pretty difficult and it was hard to know what to study for it.
- Maybe I would introduce some discussion board assignments? Since some questions have more than one answer, it may help to hear every students' opinions.
- Some things that were really hard to understand were brushed over sometimes, but we should've just asked more questions because he always asked for input and if we needed to go over something again.
- I was really sad that we did no programming in relation to compbio – additional assignments are good, but with no grades/extra credit it's hard to find time to spend several hours doing them, I always ended up prioritizing rest/learning something else, new and more fun. Also, some lectures were rushed bcs we spent 40 minutes discussing the easiest thing possible and then had 5 minutes for each slide with ton of difficult information. A lot of HW questions were not based on readings/slides, but somehow we were expected to know the answers.
- I would better specify what we need to know for homeworks/exams during the notes because certain topics we talked about in class but didn't need to know too much for the homeworks/exams and vice versa.
- I would put less detail and focus on the most basic concepts. Also, I would make my slides a clearer structure (Eg. with a mind map).
- Each test had so much material on it, it was hard to comprehend and apply everything for so much material. I would've rather had more tests that covered less material each that the big midterms we did have.
- I would make the exams less multiple choice based and more open ended based as some of the concepts are better tested with written responses than picking the best option.

- I would make adjustments to exams so the answer choices were not too tricky. It'd be better if they are mostly straightforward instead of having hard reading sentences.
- I already gave everything I would have done differently as direct feedback over email, which was integrated into the lectures already. It just included some ideas I had to reduce cognitive load by contextualizing material.
- Post lecture slides and relevant content a little earlier than during/after lecture. I would also include practice problems since the test question wording gave me a really difficult time despite knowing the concepts. More frequent TopHat questions would be beneficial. I would try to have more outside resources for some of the concepts; oftentimes I did not know where to look or even know what to search for some of the concepts taught in class. I felt the homework was graded a little bit too harshly but idk.
- I would go over the material before class to make sure there were no mistakes.
- Include coding, i learn better with practical methods, in this case coding. I understand that people choose this class as an elective but when they chose a class that has "computational" in its name, they should expect to use computers, without coding i thought it was a lot of content dumping
- I would structure the class somewhat differently. Instead of the traditional question–and–answer homework assignments, I would assign projects similar to the final project but smaller in scope. In place of the two midterm tests, I would implement biweekly quizzes that focus on the subject matter introduced in the preceding smaller projects. It might also be fun if the final project built upon these smaller projects, and allowed students some creativity, such as choosing from a selection of topics for the final project—for example, antibiotic design versus cancer drug discovery.

  For the computational biology portion of the class, I would include projects that incorporate physics–based problem solving. For instance, a homework project could involve calculating the Gibbs free energy for a protein–ligand binding interaction. Some of the course material might be easier to understand if students could engage with it in a more hands–on way.
- I would include programming officially; there were several moments where I would've wanted to be able to implement code to try out my understanding, and although there were extra programming questions given, I didn't always have time to work on them due to the other work I had for the class.
- I think that there was a bit too much breadth at times, where we would cover a topic for one slide, then be expected to be masters on it. I think it would be better to spend more time on a couple topics and be expected to fully understand them, or at least for the topics that are covered more quickly, be tested less heavily on them.
- I would post the lecture slides ahead of class. Because the lecture slides were never posted before class, I found myself focusing more on copying the information from the slide than what was being said in the lecture. Also, I would be much more clear in what I expect from students in the graded assignments. The questions in our assignments were often extremely vague in terms of the professor's expectations for our responses, and I felt they were graded unfairly.

- I would probably do a little more focus on basic concepts that are important to each unit to make sure there is a fundamental understanding. I felt like we were a little rushed sometimes.
- Have articles for every topic some of them didn't have any articles to explain the topics and the slides rarely were hard to interpret or get full information from.
- I would record lectures and post them online. Even though I was present in class all the time, certain concepts were harder to interpret just by looking back at slides alone; having the recorded lectures to refer to and refresh my understanding of concepts would have been far more helpful for me. I think with such a lecture and content–heavy lecture, it was also hard to stay engaged 100% of the time, so being able to have the video lecture available as a buffer would have been very helpful.

  ALSO WOULD HAVE TAUGHT PYTHON!!!!! I absolutely understand the reason why you chose to take it out this semester, but now that you are thinking about bringing it back next semester, I can't help but feel a little jipped in terms of what I could have learned throughout this course. I think that certain concepts would have been more understandable if they were taught through the more technical lens (i.e. actually using these databases instead of just talking about them).

  I think that homework assignments should have been more project–related, using things like PyMOL and SWISS–MODEL, so that the first time we would be exposed to these things would not just be in the project. It felt a little weird to be using these things for the first time in this project even though it should have been a reasonable culmination of what we had been learning. There was just a real lack of use of these different databases and platforms, outside of just programming, which I think was a major missed opportunity and learning experience. I think it would have been more beneficial to integrate it into our assignments, and it would have been more helpful to our understandings as well. Short answer assignments just did not really resonate with me; it never felt like there was a clear rubric, and the level of depth that we were supposed to go into was never really clear to me.
- Slow down the content rate, towards the end, it was hard to understand the materials. Also dial back the questions amount or deepness on the homework. With multiple homeworks and plenty of tests, I had to choose between getting a homework done on time or understanding the materials.
- I thought that the slides were vague sometimes. I appreciated the cool pictures, but sometimes it was easier to understand when Alex drew a simpler version on the white board.
- I feel like the assignments took so long due to the open nature of them. I feel to shorten them, maybe the assignments can be half multiple choice and half short answer. Students who get any multiple choice question wrong should have an option to request a regrade with the correct explanation of why their choice was wrong. If this is not an option, maybe the questions could be more focused and guided so students do not worry as much about what the "right" answer is. Also, because the lecture slides do not contain a lot of text for some questions, they are difficult to answer without doing more research.

  I also feel that the exams also had confusing wording sometimes and there were so many questions, especially for the first exam, which had to be answered before the time limit. This is a challenging course, so students should be challenged, but some of the questions felt tricky to answer due to the options and wordings of the options.

Lastly, for lectures, I think often the lectures goes too in depth too quickly. I think the lectures can more effectively get across to students by starting with a super simple explanation anyone would be able to understand, along with the explicit goal of said experimentation or methodology. After that, it would be fine to go more in depth and explain how everything is connected. The way lectures are now, the students have to figure out a lot of the connections and learning goals by themselves, which could slow down learning in an already difficult field. For a different field, this might be more effective, but I believe the uncertainty in some of the lectures should not be left up to students in a difficult course like computational biology where a breadth of knowledge is needed.

- As a comp bio major, I think teaching a little bit more python would be nice, but I totally understand the reason for why you did not. I think the way you have it set up right now is a fair balance.
- Include python.
- I believe it would be beneficial to include video recording and taking attendance, and counting them as 5% of the total grade. While Alex is entertaining and an expert at simplifying concepts, the concept is new to us, so we still need time to process it at home. Therefore, I think watching the recording would be incredibly helpful in aiding our comprehension of the content. In fact, watching the recording would be more beneficial than reviewing the notes we take during class.
- I would engage students by providing more course material for practice and learning. The course is taught purely by looking at slides and the instructor does not put content online after class to review and help to understand. I would make sure students know what we are learning in class and have the proper prerequisites for it. This class material is relatively new in a changing field and very hard to understand, and the instructor does not help with it.
- I think that python should definitely be included in this course for later semesters since the concepts are super important, but I wouldn’t really know how to start with the code otherwise. I know there are a lot of concepts to get through, but I would have liked if maybe some less important material was taken out or taught less as I did feel like sometimes the course was extremely fast–paced, especially for something that is mostly new to me.
- I would like more python throughout the year
- I’d give assignments back out more quickly, I didn’t have a chance to really improve my results and answers since I didn’t exactly know how I was doing, it was just a bit of a black box. I would also provide more content to prep on before lectures throughout the term because of how helpful they were to me. I would also want to implement Python into the course and to give the BIOSC classes the responsibility of applying Python for students since the CS classes teach Java, at that point it would just be a better of making sure students get practice in Python since the logic carries over between the programming languages besides a couple small changes here and there that could be knocked out as a hw assignment or during a weekend. I think I’d also provide a slight cushion to grades with something such as 5% attendance grade since you needed to be present anyway. I think more interactive questions through tophat would also be fun to introduce to get people thinking

and answering questions and seeing what people can and can't answer as it would be helpful to teach and guide self study.

- Give more practice tests
- I will improve my Powerpoint to make them organized. Also I would make exam prompt properly to cover the real content in the exam. It is important to set up reference book or website to fully illustrate the knowledge covered in lec.
- We had a final project and a lot was real world based and I wish that he gave us more information how to do that. I would also rather have more smaller tests rather than 2 huge tests.
- I would implement more lectures about the platforms that we needed for the final project throughout the semester because it was hard to start the final project without any background on how to use PyMol and MolModa. Additionally, I felt that the exams were very particular and the multiple choice options were so similar that I would often miss a lot of points based on a couple incorrect words in each option which I don't think is the best way to test our understanding of the material.
- Something that I would do differently is instead of taking out Python completely from the course be a sub–part of each lecture since attending these optional Python lectures has given me insight into how to create an expression graph or heat map by coding rather than using something like excel which I think is going to be very beneficial since it gives more diverse skills that I can use in my future career when I'm working in the industry.
- I would offer more opportunities to get points that boosted our grade. Everything in the class was graded for accuracy, including homework. Homework was also far too extensive and graded far too harshly, especially for the time we were given to complete it. It would take me about 30 minutes to an hour to complete one question and homework would be about 15–20 problems long, with each question having about 3 parts.
- The slides were kind of bare and difficult to interpret. I would include more in depth resources that have clear explanations rather than vague slides.
- The questions in homework often felt like they were not able to be answered using the lecture slides and from notes taken in class. Finding information to answer the questions was time consuming. I think questions should be answerable using just information from class.
- I would probably spend more time on the matrices and string productions, as I personally found this confusing. It also seemed like this was new to others in the class as well, so I think it would be worth it to go more in depth with this content.
- try to make a little more engaging
- I know the class is fairly new so its still a work in progress but just make sure there's enough time for all the material to be presented in one lecture as well as taking time for questions into account. I also really liked when we did topcoat questions as practice, so maybe more of those.
- Maybe incorporate some basic coding back into the course.
- Include more python.
- At some points it felt we needed a background in certain subjects. I have a background in biology but not computer science and only a moderate background in physics. This left me to miss some

of the background that Alex expected us to already know. I also would change the format of the site. I would put everything on canvas as that is where all other classes are located and it makes it easier for me to access.

- Bring Python back into the curriculum. And change the homework from being free–response to programming exercises. I learned how to write out algorithms very well, but I don't know how to actually encode them, and I fear in the real–world a boss is not going to ask me to draw out Needleman–Wunsch…
- Genuinely, nothing. This has been one of the most enjoyable classes of my time at Pitt, and while difficult at times, kept me consistently engaged, putting in effort, and enjoying the course material.
- I would definitely make coding a larger part of this course, since it is expected of Computational Biology majors in their careers very often. It would be great if the coding exercises for Python that we had were extra credit. I also think a large portion of the course involves memorization of information and would have liked more problem solving, specifically with some kinds of math. Another idea – since the lectures are very heavy on content, it seemed like the professor struggled covering it in the class time and I often felt like the slides were rushed through and hard to follow. I think what I would do differently is give students some reading and exercises to do before class. Lastly, some smaller quizzes to test and keep up with material, knowing where students stand.

What aspects of the computational biology degree/program could be improved to prepare students for their careers?

- A greater investment/expansion into the program, as more students are choosing the major. Career opportunities/advice, or mentoring.
- More major specific classes and real world applications
- I feel as though the current degree/program sets students up to be a computer scientist with a strong understanding of biological sciences, not necessary to be a computational biologist. The degree could also benefit from possible sub–fields of study(e.g. Genomics, Simulation and Modeling, etc) that allow students to take more classes that align better with their interests, instead of taking just one extra class.
- I feel like more technical skills early one will allow us to develop these critical skills early and be able to work on projects quickly.
- I think that there should be less emphasis on java and even organic chemistry. As someone that is a computational biology major that is not going to med school or applying to be a software developer, these courses take up time that I could spend learning Python or learning relevant bioinformatic research.
- More specific classes would be great.
- Make the computational biology class the same across all sections because my friend who took it a couple semesters ago said it was way different than the course I am taking right now.
- Perhaps having a better understanding of the industry component would help a lot better, as it stands for now this course is really pushing people to get higher education for a lot of job opportunities.

- A lot of compbio labs
- I think more practical applications could be incorporated into the courses
- More class diving into specific area of CompBio.
- More research, internship, and shadowing opportunities.
- There could be more resources as well as more classes.
- Incorporating cs and biology skills can broaden students' career options.
- I would like some more information and guidance on what I could do with the degree after graduation, as well as available industry connections. I also wish that the computational biology courses were offered more often and at more than one time, since having some courses offered so rarely (like BIOSC1544/1542 only being offered in alternating Spring semesters) makes it very difficult to plan my schedules. More resources for help (tutoring, etc.) with computational biology concept understanding would be really helpful.
- There aren't many major–specific internship opportunities or hands–on classes available through the Pitt programs. A lab course might be useful.
- Class for majors and non–majors is a good idea.
- More specialized electives would be helpful. It might also be helpful if BIOSC1540 included a lab component.
- There are no high–level Python courses outside of CS0011, which is very rudimentary. I am a junior, and I won't be taking a programming–included computational biology course until next semester. The biology–based courses don't touch on the computational side at all, and the computing–based courses don't touch on the biological aspects, as they're focused on the software engineering industry standards. Additionally, it often feels that the advising staff don't fully understand what the program looks like after school, and they seem less confident in giving advice.
- More computational biology classes, rather then a lot of comp sci and bio classes. Also having more options for classes in the major may be nice allowing for specific concentrations since the field is so broad. For instance, a more software design heavy option where you go into JS / software, or a more bio heavier focus, where you take classes like systems bio / genomics. The ideal would be if these classes could be comp bio specific, instead of being just the general versions too.
- I like how the biology department at Pitt sends out emails with updates and information, and I think it would be nice if there was something similar for Computational Biology since it is a more niche major. I feel like there's not a lot of information being shared.
- a recitation
- I think that there should be two "optional pathways" within the computational biology degree. Like I think there should be a more data science, bioinformatics pathway and a structural biology pathway. i.e. the bioinformatics pathway would include suggested courses like Data Science with Python, Data analysis (INFSCI 0510), computational genomics, things related to the programming–heavy side of things. the structural biology pathway would include things like biophysics, that chemistry computational drug discovery course. It just feels like right now I am not really prepared for either of these fields based on the way this major is structured. Like I only know things about regression and machine learning concepts because I am a stats minor, it has nothing to do wiht

actually being a computational biology major even though that is a necessary understanding and skill for this field. Like even if i wanted to learn more about structural biology, physics is a really important part of that but its not even an actual major requirement. It feels like all the prerequisites that I have taken to get to this point are really not even that relevant to this major and I waste a lot of money and time taking classes and learning concepts that I will never use again.

The optional pathway idea is just based off of how the information sciences minor in the SCI department is structured but I feel like it is really great for tackling and getting a better idea of skills that you need within this field. even planning wise I would have been a lot more prepared if the major was laid out for me in that viewpoint.

- Having a research capstone class, whether in simulation or genomics. This allows for people in the Comp Bio program to understand what they're getting into.
- more python courses and more comp bio specific classes; definitely feels a little bit like I am getting 2 separate degrees in biology and computer sciences as there are not a lot of classes offered the combine the two fields.
- I do not think there is much outside of the classwork present to prepare for a career in industry. I know research is widely available, but in general, I feel that other major programs have much more resources to get internships and such that will directly help me get a job after graduation.
- Provide more detailed information about the day–to–day life of working in this field, and focus on teaching marketable skills rather than just imparting basic knowledge.
- There should be more courses specifically catered to computational biology majors, and there should be less prerequisites to declare this major. The five requirements are hard to fit in quickly with all other requirements we have to do, especially if we want to pursue more academic plans.
- I wish the program relied more on coding related to python. It is nice to have the coding classes where you learn the logic, but with most of them being taught in java, I don't feel that it would be quite as beneficial as something taught using python. If this is also a degree that is meant to prepare students for a career in comp bio, I feel that it should be stressed as an undergrad program that is really only for people who are thinking of going into comp bio and maybe not catered as much toward others who might be interested in the major but want a different career path unless they want to take comp bio as an elective without all of the coding concepts. It would also maybe be cool to have a way for comp bio majors to do research in comp bio labs and get credit for it to go towards the comp bio degree.
- Actual application of the concepts we learn
- I think more application of actual computing in the major would be a good start as the CS classes feel like they want us to be data scientists and not computational biologists. Machine learning should be part of the program, and so should physics, and something even more advanced like cell biology. I think the degree requirements could be tougher and should be tougher.
- To have more resources to seek for students to prepare their careers

- The introduction class for coding is “Introduction to python”. Somehow the intermediate class turns to JAVA. Why not set the primary class just aiming at JAVA since all the high level courses of coding are based on JAVA?
- The real life experience.
- I think understanding that this program is always going to need to be changed in some form every year since the field is constantly growing and new techniques and algorithms are being created that makes others no longer the industry standard
- Allow more opportunities for research, internships, and careers. There are little to no tables for comp bio at the career/internship fairs.
- Research opportunities for undergraduate computational biology students. More python classes because the java classes we have to take aren’t career–relevant. Also we only have 2–3 actual computational biology classes.
- Because this varies for majors and non–majors, I would emphasize which skills would be particularly useful for bio people and comp people, and comp bio people. This was already touched upon at the beginning of the semester, but I think this could be further developed a bit more.
- Providing everything computation al bio entails, specifying what it is/makes it different from bioinformatics or systemic bio. Also career opportunities and applications
- I am not a comp bio major, but this class made me consider it so I feel like it prepared me well.
- Incorporating more math courses could be helpful to the undergraduate major.
- More comp bio courses and don’t have them be only in one semester or alternate each year (I understand the major is small and new but still)
- I am not a computational biology major
- Make BIOSC 1542 and 1544 available both semesters or both in the same year. Add electives that incorporate computational aspects to biology more.
- Since the biology is easier to learn later on, I think the emphasis should be more on coding when it comes to the curriculum. More guidance towards internships and research opportunities always helps. I also think that the computational biology class definitely should be more marketed towards majors, rather than just an elective. A good foundation with coding and algorithms is super important.

What additional resources or support would enhance your learning experience if you wanted to pursue computational biology as a career?

- I’m not sure, I think that the courses do a good job of introducing the field.
- Having a textbook, or being able to have access to to the main course concepts in a physical and centralized form would be helpful. However, I recognize this is a relatively new field and those might not be available.
- I would like to see more career, research, and internship opportunities.
- There are so many topics in the field and they all require really high–level knowledge to understand, and you tried your best to simplify things but at times I still found myself kind of lost around many of these topics and/or only having an extremely surface–level understanding of it, especially due

to lack of resources online besides scientific papers which can be hard to understand. So I think your “omics breadcrumbs” website is a great idea and I would have loved to see all of the topics in it.

- More research opportunities provided through the school.
- For this major in particular, I think both research and bioinformatic software (Biosc and Cs 1640) should be necessary for students, I think both are rather necessary if we are to really pursue a career in this field. I think if both courses were required, albeit being a harder major, it would prepare students for what comes after. Machine Learning in this context also applies, if there is an in depth ML course about how it is being applied in C–Bio careers.
- More of a focus on programming because the biology half is focused on more.
- Enough research opportunities and industrial connections.
- More python and coding classes tailored to computational biology majors.
- More connections and career guidance.
- connections to research, internships, and coding based computational biology classes, more opportunities to apply computational biology concepts
- Study and practice material outside of class would’ve helped
- Perhaps having a professional internship system, somewhat similar to the engineering schools coop program.
- I think if there was more Python and more openness from research faculty it would make it easier. I think that, even if it was as simple as research faculty going to a class and giving a talk about what they do for 5 minutes, it would make the major feel more accessible. The lack of Python, especially as it was cut from the base Computational Biology course, makes it harder for a student not in research already to understand the expectations, and they can’t realistically get that knowledge until late sophomore year at the earliest.
- Maybe more information about what it means to join industry, what requirements you need, and general career paths.
- Some kind of event for computational biology majors to make more of a sense of community, and just more resources being put out overall.
- More research internship opportunities etc.
- CAREER RESOURCES!!!!!!!!!!!! Literally the very first time ever that anyone had ever mentioned anything related to computational biology and its careers was your class and the very first lecture. Talking to advisors is literally useless because it feels like they don’t actually know anything. They are just like listening to me and nodding their head….. like give me somethingggg. I think that more opportunities to take Python programming–based classes would be really, really great as well. I think that is something I am really missing, and as someone that didn’t come to college with a programming background, I feel that it is something I am pretty insecure about in this field.
- Career guidance and easy opportunities to do research with faculty at Pitt.
- Specific advisors that have greater connections to comp sci in addition to biology; more resources available for career opportunities.

- Internship help would be nice. I have applied to so many internships, but have been widely unsuccessful.
- A club or a organization that meets weekly and teaches coding related to the field of computational biology
- There should be catered counselors or people with experience in what computational biology can be as a career. As a student it is hard to understand what I plan to do in the future especially because of the limited resources, but I find biology and computer science as fascinating subjects and want to use them both in my career.
- At least through the Department of Biological Sciences, I think it is kind of hard to lump in comp bio with the other majors in that department. Although the biological sciences advisors are wonderful people, I don't think they are very knowledgeable when it comes to such a niche career. It would be nice to have advisors who have backgrounds in comp bio to help guide students through the major. I know some of the other departments in Dietrich do have some of their professors be students' advisors, so I think that could be a possible suggestion.
- More practice questions or more exercises in class
- More in class time learning how to use resources like PyMol
- More networking opportunities to find jobs in the field that aren't just for research.
- Research, Grad school information and preparation, Shadowing, information on possible careers in both academia and industry.
- I really liked the introduction to Kaggle, as this was helpful for someone who has no prior python experience.
- info events
- Encouraging students, especially biology students, to take this class earlier in their career may spark interest in pursuing a comp bio career.
- It would be helpful if there were more information promoting the major itself because I had not known about it until a couple months into my freshman year, hearing about it from a source outside Pitt.
- more courses
- I am not a computational biology major
- Teaching me how to actually run the programs we learned about in Python and not just web–based applications.
- Help building industry connections, as in fairs for computational biology or bioinformatics or something close to that. I also wish there were more clubs or organizations that were geared towards computational biology majors who are interested in PhDs and Master's, not just pre–meds.

Based on your experience in this course and understanding that I am applying for a teaching faculty position in computational biology, please provide feedback about my effectiveness as an instructor and my potential contribution to the department's computational biology teaching mission. Your feedback will be considered as part of my application evaluation.

- Alex provides detailed but not convoluted explanations of topics that may be difficult or confusing. Lecture material is presented in a way that is easy to understand and to refer to since there is not a lot of outside, readily available information about some topics. He has demonstrated a comprehensive understanding of the field as a whole and has constantly updated us with recent findings and publications, which is essential due to the rapidly evolving nature of computational biology. He can answer questions effectively and provide career advice to anyone who may ask.
- Good lecturer and provides lots of resources but could slow down lectures and simplify them a bit more because it was harder to grasp some concepts. providing lecture slides before class so people know what to expect would be good.
- As an instructor I felt one of the biggest strengths you had was keeping the content engaging. Some of the concepts covered with pretty advanced and difficult, however, you kept lectures enjoyable which helped me stay engaged when I would usually zone out. Outside of the classroom you work with the computational health science club, and while the club is still new I feel like you are helping guide it towards something bigger that will help all computational biology students.
- The content taught is good and relevant for a proper introduction to the field. However, sometimes, it is difficult to follow the content that you're teaching in lecture at such a fast pace, or sometimes its disorganized, which I understand since its your second time teaching the class. Once the class becomes more structured and developed it should be fine.
- I think Alex did a good job of explaining computational biology concepts in a way that was easy to understand in lecture.
- I hope you get this position!!! I think you're a great lecturer that can contribute a lot. I really like your "omics breadcrumbs" website and think that it's a great idea. However, I think sometimes your expectations of us can be a bit high, and there's very few resources for this class, so I do find myself struggling (especially on the final project).
- Alex is a good teacher and seems to be really passionate about the computational biology field and teaching others about it.
- I think Dr. M is a great professor. He provides many opportunities for students to seek help if necessary, and show off their understanding to him if not. I found him to be a rather approachable person, and he also makes sure to provide the most optimal answers. He is also very passionate about what he teaches, and that makes it a lot easier to follow the class.
- Dr. Maldonado is a fair, smart, and overall amazing professor. He is one of the most knowledgable teachers I have ever had and inspired me and will definetiley inspire those to come in future classes to want to explore the field of computational biology.
- I thought you taught very well and explained complicated concepts in a way that makes sense.

- Prof is knowledgeable enough as an intro CompBio teacher. He've tried his best to introduce the full spectrum of CompBio. The only thing I want to sugget a change is the richness of content (which I think is a bit too much for me as a sophomore).
- You explain and are very knowledgeable about the concepts you teach. In addition your work in the field provides good insight into real applications of learned concepts.
- You are a very effective instructor and would excel if given the chance to add more to the computational biology curriculum in other classes.
- Alex is approachable outside the class, willing to help every student in the class. In terms of teaching an lecture, I think it would be better if he can explain things slowly and more clear. Exams are okay but the multiple choices are hard to read and too tricky.
- First and most importantly, this course is open–source and accessible to anybody on the web. It is refreshing to see that for the first time, especially for a course that teaches the fundamentals of drug design, a common good for humanity.

  Speaking further to Dr. M's integrity, he spends an incredible amount of effort on every little aspect of the course, going every extra mile there is to cater to the smoothest possible student experience. I have genuinely never seen anybody work so tirelessly on a project. It is built from the ground up in a medium independent from canvas, which, needless to say, works much better. I've even told him that he's clearly going above and beyond what's required, and in response, he proposed a plan to restructure the entire curriculum, integrating Python lessons into every lecture—a change I fully support and believe will make the experience even better.

  But that is not to say Dr. M is unreceptive to feedback. In fact, he is the only professor to my knowledge who asks for direct feedback on lectures. He recognizes that growth doesn't come from getting everything right the first time, but from continuously refining and improving.

  In lecture, we did not learn any false conceptual frameworks. Instead, we were made to understand how complex things actually work, just presented in a clear, digestible way. This method, in my opinion, is far more effective than the traditional approach of learning things in overly complicated (and often flawed) ways, only to have to unlearn and relearn them later.

  Needless to say, there were no problems with office hours or emailing for help, which I certainly needed much of. At the risk of sounding arrogant, this has been the only course that has challenged me academically. With just one semester left until graduation, this is my lowest–performing course, but that doesn't mean I didn't learn a tremendous amount or thoroughly enjoy it. Quite the opposite —I'm grateful for the experience, and I highly recommend the course to anyone who wants to learn how bioinformatics and drug design really works.
- I think I did learn a good amount in this class (despite me having a difficult time grasping most of the concepts), and I appreciate your consideration for other students, making adjustments to class structure or the way that some of the information is presented in the slides. The use of a class website was very helpful in keeping the information organized. I think it would be beneficial

to have the lecture slides looked over in more detail and prepared and posted earlier, and provide more practice for the exams and how to approach problems.

- Your class expectations were very clear and unit organization was effective. Your use of visual elements aided in understanding and your talking was clear and well paced. It was clear you knew the information well, were confident in your knowledge, and could answer questions with an in–depth response. The notes left on homework assignments were useful in improving for future assignments. I feel that you could improve upon your planning as some slides contained errors that you realized during class.
- Good at lectures, really prepared for every lecture and managed to make very content heavy lectures fun, kind of a hard grader in assignments but exams were very fair. I think you should provide more study material and practice material, it was really challanging to study for your class outside of lecture.
- Excellent instructor, extremely knowledgeable and approachable. Really goes out of his way to promote student interest in the computational biology major, and encourages student success.
- I think that it was obvious you cared, and that the effort put into the class definitely had a positive effect. I feel much more confident about my interest in the subject now, and I think that this class was an overwhelmingly positive experience for me. You seemed prepared for lectures, and down–to–earth and approachable. While some of the material was harder to understand, I think the way you presented it made it easier to pick up.
- I thought that the teaching style was mostly pretty good, though often the amount of content felt overwhelming (though I generally got it in the end, studying on my own). I also found student hours to be incredibly helpful, as getting a second, different and slower explanation of certain concepts was incredibly helpful.
- I think that you were a very good instructor and put visible effort into helping us understand the content. I think there were some points at which you assumed people had a lot of understanding of topics that were pretty confusing. Also, some of the homework questions wanted depth in the answers that was not gone over in class, and it was hard to find information on the internet. Overall though, I think you were a good professor and have a lot of potential to improve even more considering you are fairly new and the computational biology major is also new.
- First off, thank you Professor Maldonado for your instruction this semester. I think your teaching was good I liked the idea of top hat questions through the lecture, I liked your analogies a lot most of them were very good. I think the switch you did by the end making the assignments have a word limit and the criteria being a little more clear was good but maybe do that from the beginning.
- I think 100% as someone that actually has a background in computational biology, you are an addition that we absolutely need in the computational biology department. It feels like there is such a low level of knowledge of computational biology within the bio department, and you are the only person that has actually given us guidance relevant for computational biology majors. I think that you are really doing your best to teach us a lot, and I think that I really learned a lot in this course. You exposed us to a lot of important relevant concepts, and compared to what I heard it was before, you did a lottttt. You address a lot of things that no other course had really taught

before and I think you were a really great resource to us and preparing us for success. Like I havent never used anything I have learned in organic chemistry in what I am doing right now, but I KEEP applying topics I have learned in this course to what I am doing everyday in research.

- You're content delivery was great You interact well with the students. However, I feel like towards the end, the content delivery rate was increased and I started struggling to understand what was being taught.
- Very understanding professor in situations that were outside of my control; engaging and funny in class; willing to adapt the class to the student's opinions and in their best interest; very knowledgable about the field. I really appreciated the career advice that he gave throughout the semester. It is very helpful to get that information from someone who understands both the industry and academic worlds of comp bio. Overall really great professor. I wish I could take more classes with him!
- I think that you would be a great fit and you are giving students an insight and understanding into the computational biology world. There are some things that could be changed about how the course is structured and taught, as I previously mentioned, however.
- I think your slides were very engaging and understandable. You taught pretty well, and was always open to answering any questions. I do think you went slower in some parts that you could have gone faster, and faster in some parts where I think it would have been better if you went more in-depth. Overall, I thought that some of your best teachings were in the exam review lectures before each exam. I really felt that youn explained stuff really well during those times.
- As a comp bio major student, from my perspective Alex done an amazing job as an instructor. He has opened doors for aspiring computational biologists. He introduces fundamental concepts and equips us for future courses like simulation and modeling. His clear and effective introduction to the major equips us with resources to enhance our self-preparation for future careers.
- I believe that there should be a better understanding of the students in this class and how to cater the best way to teach and convey the concepts of computational biology. With the pace and content of this class, it was very hard to grasp what was happening especially with only foundational knowledge in biology and chemistry. With the no python requirement, it felt like there was heavy reliance on the concepts that require so much more knowledge than just foundational science.
- I believe that Dr. Maldonado is extremely passionate about the material he is teaching, and it translates to how he teaches the course. It is really nice to also learn from someone who has a lot of knowledge in comp bio and does research in this field. Dr. Maldonado was very accessible, and his office hours were great for better understanding concepts. The material is hard as I expected it to be, but I did walk out of this class with a lot of new knowledge about comp bio as a field as well as career.
- Alex is an amazing professor. He makes sure you understand the concepts by simplifying them and building off of that knowledge.
- It was obvious that our professor is extremely competent through his lectures and explanations. There was never a question he couldn't answer, and he always had recommendations for

your success if you asked. He’s an extremely ideal professor who accommodates everyone (Programming+)

- I think that your effectiveness as an instructor is good. I liked how the structural biology section was structured and I liked the content we learned. The teaching style was good one imporvement is that the rubric of how homework assignments should be done to be more explicit and more detailed.
- Professor is a nice person and is kindly available to students.
- This class was genuinely so boring. I fell asleep almost every lecture I can’t lie, I had to chew gum to stay awake. Dr. Maldonado would get himself on tangents frequently, and I felt like he talked down to us instead of trying to talk to us as an educator. He had this vibe about him that came off as “I know everything about computational biology and therefore none of you know anything, and I am always right”… when he is not a computational biologist by trade. It was very condescending, and the way he spoke to us did not make me want to engage with course material. Additionally, he would gloss over some topics, and then would ask us questions about them on exams. It did not feel fair, especially when some of the nitty–gritty details he would test us on were not in his lecture slides. Also, he speaks very fast, and I often had a hard time taking down notes as he was lecturing without falling behind.
- I would say it was decent. It is clear that you have a lot of experience, I just wish you slowed it down so people with not that experience could follow.
- I think how the topics were explained allowed me, who only took CS0011, to gain a much deeper knowledge of some of the tools I’ve used in the past. Though I did feel like a bunch of these more algorithm–based concepts went over my head I felt like I could go and ask a question and receive a very insightful comment and gain an understanding.
- I feel as if the professor was not able to convey the material properly. It was hard to understand lectures, and the slides were of no help. There was little to no supplemental material to turn to.
- Professor had deep understanding of the course material. Lectures were decent, a bit heavy. Assignments were relevant but harshly graded considering the length restrictions and limited resources. The exams were ok, but the wording for the questions and all the multiple choice questions was unnecessarily complex and confusing, so it was difficult to tell if your answer was correct even if you knew the course material. Also didn’t give clear example of format for open–ended responses (had multiple different formats in the slides), then penalized when it was formated one way that was in the slides vs. another way that he didn’t specify was required. The project is very good and useful application that I am enjoying so far. I also liked the python classes at the end of the semester.
- I think Dr. Maldonado was a really excellent instructor. The content was delivered clearly, and made accessible in multiple formats immediately after lecture. The homework and exams were directly related to the content presented, and he always made room and time for questions. Additionally, he was always available in person and online outside of class and was extremely apporachable. In an academic environment at this level, it is refreshing to have an instructor who does not look down at you, but rather wants to build up his students to prepare them for post–graduation. Dr. Maldonado is additionally very knowledgable about the work and research opportunities within the

greater Pittsburgh area. He has made this educational experience very enriching and supportive. After this class, I feel very prepared for the next steps in my research career, and have gained interest in computational biology as a potential career path.

- effective overall beneficial office hours slides were always posted and homework was manageable
- Alex is truly the best teacher I have ever had at Pitt. He is so easily accessible for office hours, over email, and after class for general questions. He is also energetic and very approachable which makes his class that much more enjoyable. With this being said though, I never felt like I could skip his class because the material was easy, but I never minded attending class because it didn't feel like a chore.
- I believe Dr. Maldonando did a good job at explaining difficult topics in simpler terms. Since this is an introductory course and actual computational biology is incredibly complex, it is vital for the professor of this class to be able to look at, to interpret, and to explain this content to less knowledgable people, which was done well in this course.
- you teach well and make things clear just go over your slides in advance
- I believe Alex really cares about his students. His teaching can be rocky at some points but it is clear it is from lack of experience and he works hard to learn. He accepts feedback from students and clearly wants everyone to understand the material and succeed.
- The professor is very flexible and accommodating with students. He answers every question with thorough explanations if you ask him during lecture or office hours, giving examples and explaining things on the board too. He is also very approachable which makes it easier to ask questions. I think he did struggle sometimes covering large amounts of content during lecture, but he genuinely wants students to succeed and just needs more experience.

What aspects of the computational biology major have been most challenging, and how could they be improved?

- I think the beginning of the course had some challenging material, but I think focusing on the conseptual parts was good.
- the actual coding that we will be doing would be good to know and practice
- I felt like some of the science concepts were a little challenging. I had a rougher time in Biology 2, so I probably would have benefited by waiting till I had taken some more biology classes like genetics before taking this class. Possibly suggesting a higher level science class be taken before this class might have been helpful, but not everyone would really benefit from that.
- I feel like a lack technical skills and hope the more technical and practical skills an be taught early on rather than later in the undergraduate years.
- The field is just so broad and so deep at the same time, with so many different topics that all require high–level knowledge to understand. Also, doing the practical applications is pretty challenging and high level as well. I'm not sure how this can be improved aside from more specialized courses and providing more resources (please!!). At times I find myself both clueless and helpless when a question or topic isn't something we directly covered in class because I struggle to find information about it online aside from a few papers that only complicate it more and there's no resources

provided for us in this course aside from the slides (which can often be vague and/or not cover the question), and it can be very frustrating.

- The computer science courses are sometimes pretty difficult, maybe only required some basic level courses would be easier.
- I found exams to be rather heavy in content; there are too many topics and this makes studying for them rather hard. I would recommend increasing the number of midterms, and perhaps having 50min lecs instead of 75 min ones. A recitation section which is strictly for preparing for midterms/finals would also be nice, opposed to a recitation that is for the project.
- I think all of the first unit, but thats because I didn't think this was going to be a full on computational biology course. It was and I was wrong. This class felt like it taught me the mojority of what I needed to succeed in this field and i was not prepared at the beginning. I feel like others definetely felt the same thing so clarifying that at the beginning would be very helpful, and not to scare students away, just to prepare them that every lecture needs to have full concentration.
- I think it goes too in depth with certain topics without having actual practice with tools/applications to help with understanding
- It takes time to dive deeper into concepts by ourselves. I would like some more guidance on self–taught progress.
- I think the coding parts. I also struggle with application of material I learned.
- Personally, it is the coding part. Since the job industry require us to do more python an r, we should have focus on that instead of bunch of java stuffs.
- Understanding the concepts of computational biology. I am really not sure but I think if i took more classes as a prerequisite to taking this course I would understand it better; a lot of the algorithms and math–based concepts were confusing to me. Maybe if the course required some coding I would understand it more.
- Understanding how to use and read the current databases needed to do most situations and construct models. Spending some extra time teaching about the use of these resources could improve this.
- Bioinformatics, probably including coding could have helped
- Many of the co–reqs for the major, in particular the chemistry/biology co–reqs, are heavily geared towards the MCAT and are brutal! Some of the CS co–requirements are tediously dry. Having additional major specific courses would be a welcome respite for comp bio major students.
- It feels like the people in charge of the program don't know what they actually want the program to be. As I've mentioned before in this survey, the lack of department support for increasing programming inclusion makes it feel almost like the department wants us to fail. Also, some of the required courses are important, but from how those instructors speak, it's obvious that they never think about this program or any similar ones (I think clearly of how in one of my courses, which had a large pre–med section, my instructor had us memorize something solely (in her own words) because it would also be on the MCAT. I am not planning on taking the MCAT.) I feel like because the program is so small, it's not well–known, and it feels like the people controlling it don't want it

to be well–known or advertised. I only found out about the major because a friend of mine is also majoring in Computational Biology.

- I cant think of anything specifically difficult, besides hard classes which is to be expected.
- I think the most challenging part is making the major feel more cohesive because the coding classes are a lot of Java, and it seems like computational biology classes are going to use a lot of Python. Additionally, the pre–requisites for this class were basic, but I felt like some of the topics dealt with a few things going on in classes I'm taking right now, like Organic Chemistry.
- The exams are very challenging to me at least the multiple choice answers some of them will be obviously wrong but sometimes 2 answers will be almost impossible to discern that is my opinion however and maybe due to not preparing enough however I believe I prepared enough and knew the answer generally but sometimes not to the level that the question required.
- Taking courses that are actually relevant to this major. I personally think the entire major needs to be relooked at and revamped so we are actually learning more relevant concepts.
- Simulations. Not sure because this is my first Comp Bio focused aspect.
- Finding outside of classwork extracurriculars like internships.
- I think the mathematical parts are challenging, I can't understand the equations introduced with knowledge in cal 1. I feel like the department need to recognize the importance of mathematic to this major and includes more mathematics courses related to this major.
- Almost everything. Teaching needs to be at a manageable pace.
- It is definitely a lot to cover all of the science core courses, CS core course, and computational biology core courses. I have found having to do science and CS classes has been rather difficult in the same semester, but you can't really put either of them off because to do the computational biology major, it is very helpful to have some classes in science or coding under your belt already. Pretty much all of the classes are important, so I'm not exactly sure how they can be spread out to make the class load a little bit lighter.
- Certain concepts were challenging, but having the lecture slides beforehand would help me better understand and learn what we do in class
- As of right now, a lack of practical application until taking a second course in CB could be something that's improved upon. I also don't know if this course was more challenging because the semester was technically accelerated.
- The genomics part it could be improved with more detailed explanations
- I think is coding. It is better to set up coding class especially for ComBio students.
- The lack of experience for real world
- I don't know since I'm not a comp bio major
- Combining the comp and the bio. Could be improved by offering classes that combined the two.
- limited resources
- In my opinion, the matrices and the computational techniques we learned for sequence storing were challenging. I think they could be made more accessible by extending the time we took to learn them, and perhaps work through more in class examples.
- the implementation of coding and physics into all the different models and simulations

- The most challenging part of this class has been the applied project aspect, but like I said earlier, Alex is very accessible and will help you as needed.
- The toughest portion of the major has been balancing a combination of natural science and computer science in my coursework, but it is possible with just a little extra effort.
- more practical stuff and practice
- I am not a computational biology major
- The fact that I only take four classes that actually center around CS + Bio together, and otherwise it's a CS minor + a Bio minor + supplemental math and chemistry… I worry I am not qualified to actually pursue this as a career.
- This is more personal, but having to take organic chemistry and computational biology in the same semester has been extremely stressful and challenging. The coursework is definitely challenging when there is a need to take natural science and more coding related courses. I have heard that organic chemistry and chemistry in general is really not used that much in the field of computational biology. I think that like how engineering majors have a separate chemistry course designed for them, as well as different tracks depending on whether or not they want to do something related to chemistry, it would help if there were chemistry/organic chemistry courses designed for CompBio majors that are more suited to their careers and not just "weed–out" courses to challenge or test them. Something that is like computational chemistry would be a lot more engaging and I would feel motivated to work for this, since I know I will be using this in the future.

What are your career goals after graduation?

- Research in pharmaceutical sciences
- Medicine.
- grad school + job
- After graduation I either want to attend graduate school to study genetic counseling or become a software developer.
- I'm hoping to purse an MD only or MD PhD in some area relating to computational biology for the PhD portion.
- Grad school
- I am a computational biology major and I was originally thinking of heading into the computational biology field, but now I realized that I would rather not do anything biology–related if I can manage it, and that I would also only like to achieve a bachelor's, if possible (so no Masters/PhD program unless I am out of options), though I will stick to this major still. (This is not your fault/the fault of this class, just a changing of my personal goals)
- I want to attend dental school
- Masters, PhD if possible, then I am escaping to industry (I wish to buy a house before I am 30).
- I don't really know. I plan on getting my masters but in what I am not sure yet. I switched my major from comp bio to marketing but I loved this course and want to do something like it in the future so I think I'm going to get an internship, graduate, then keep working until I decide what i want to

master. Currently definetley something science related. Medical or pharma sales is my ideal job right now however, if I can I would also love to be a computational biologist.

- A programming job, hopefully with medical applications such as protein analysis or genetic sequencing.
- Master Degree
- I am on the pre–med track and want to continue with that
- PhD in computational biology or AI in medicine.
- Dentistry
- I am really not sure about whether or not I want to go to grad school—-I would like to get a tech related job. If I do end up doing grad school, I would want to work in the biotech industry.
- research
- I would like to pursue graduate school.
- I am currently planning on entering academia, with an end–goal of teaching at the university level.
- Graduate school, and go into some kind of research either academic or in industry.
- PhD in biochemistry, then industry/academia
- I want to get a Master's degree or PhD.
- We'll see I am currently comp bio major pre med one of those will be my career in my opinion but I'd like both skills so that I may choose at a later date
- INDUSTRY!
- Become a research PhD focusing on Bioinformatics
- Most likely to go for med school and possibly be a physician.
- Going into the pharmaceutical/biotech industry.
- Pursue a master's degree in computer biology, possibly even a Ph.D.
- Masters in computational biology or data science
- I would like to go to physician assistant school after graduation.
- Bioinformatics, SWE, grad school, etc
- Grad school
- To get into medschool and become a doctor.
- Applying for master or PHD degree.
- Med school
- To work in industry before going to med school
- I would like to get a job in the industry or go into graduate school.
- grad school, probably in bioinformatics
- I am pursuing my PhD after undergrad.
- Apply for jobs/enter the work force.
- My main career goal is to attend medical school, but I do find computational biology very interesting and wish to learn more about it.
- PhD or MD
- Working in a biology lab
- PhD

- PhD and then work in either industry or research

What are the essential qualities of a successful instructor for computational biology courses?

- Someone who has a deep understanding of the material as well as the field and how is is changing.
- A successful instructor for computational biology needs to be knowledgable and be able to effectively communicate. This class is not one where a instructor can just read of the slides verbatim and expect the students to understand everything when exam time comes, the instructor has to understand the content so that they can convey that information in an effective manner.
- Explaining things in a way that the concept is easy to understand without too much fluff/complexity that isn't needed to explain the topic at hand well.
- I think that they should be able to simplify high–level topics concisely and also be sympathetic to the fact that many undergraduates may not fully understand many of these ideas or these programs.
- Respect, understanding, empathy, kindness.
- Being passionate and knowledgeable about what they teach, being approachable, I think they should also be human (optional), someone who does their best to understand their students and succeeds on education them should be a qualified instructor for these courses.
- someone who is very knowledgable
- I think balancing the computational and biological aspects of the major
- Explain concepts concretely and set appropriate richness of content.
- I think they need to know very integrating knowledge about the content, have in–field experience, and be able to simplify the concepts to people who have never been introduced to them before.
- Willing to help students without any background of cs or bio so they can feel comfortable in learning.
- Besides experience in computation biology itself, I think its the same as for any other course:, clearness in expectations, fairness, availability for help outside of class, flexibility / receptiveness to feedback.
- considers student opinions, a good amount of availability, able to simplify complex topics
- The ability to answer questions about fieldwork and the use of class material in a realistic context.
- Enthusiasm, knowledge, promotes problem solving ,respectful attitude towards students.
- Approachability, knowledge and the ability to share it well, motivation, and a compassionate attitude.
- Someone with a strong understanding of general computational biology ideas (modeling, and genomics), and a strong ability to explain everything assuming that students have little to no background in subjects other then basic bio.
- Being able to connect with students, having the ability to explain complicated topics at various levels, deep knowledge of computational biology
- Knowledge of course, ability to speak, these are more general however for instructors of anything. I would say a genuine enthusiasm for the subject which Professor Maldonado has as he would frequently review things himself and research throughout the semester.

- The ability to explain difficult concepts in a digestible way. Experience wihtin the computational biology field i think is a MUCH – that is how we can really benefit in terms of career and academic path insights.
- An interactive/not–boring delivery of the content. Otherwise people will struggle to understand what was being taught, and since there are no textbooks, they flat out what understand the topics.
- The ability to break down complex topics so anyone can understand them. Also the ability to connect different topics to each and why we are learning something and how it relates to the previous topic. The reason is due to the sheer amount of complex topics required for a one–semester course.
- Focus on getting you the knowledge needed to actually apply your education and get an actual job.
- To simplify the concept and make it engaging, while simultaneously imparting programming skills, Alex has excelled in all these criteria.
- Knowing what students have learned and how to go further to teach more but at a manageable pace
- Someone who is very accessible, is knowledgeable about the content and has a background in comp bio, and someone who is able to reduce more complicated topics into something undergrads with less background could understand.
- I think someone who is currently active in the field and still learning and up to date about the field is necessary
- Nice person
- Good explainier, good coder, expert in the field.
- Have strong biology and coding ability
- Helpful, ability to reach out with him, knowledgeable
- I think someone willing to put in the work and make a big effort for their student's success.
- Wanting to help their students get better
- Essential qualities include patience, and ability to explain complex concepts in digestible words.
- understanding of content as well as what students can and cannot understand right away
- Knowledgable, accessible, approachable, confident
- The instructor must be well–versed in the topic but also be able to effectively communicate what they know to beginners.
- Can teach well and adjusts to the students needs
- Being able to explain concepts to students of all different background – both computer science and biology
- Someone who is engaging, willing to push students, and has ample practice for real–world applications of their course material.
- Approachable, engaging, flexible, makes the class more conceptual and problem–solving based rather than heavy memorization.

Spring 2024

I received evaluations from 36 out of 69 enrolled students, representing an 52.2% response rate.

The standards the instructor set for me were

Too low	0
Appropriate	21
Too high	14

How many hours per week did you usually spend working on this course outside of classroom time?

Less than one hour	1
One to three hours	12
Four to six hours	13
Seven to nine hours	6
Ten or more hours	2

The instructor created an atmosphere that kept me engaged in course content

Strongly disagree	1
Disagree	4
Neutral	8
Agree	16
Strongly agree	7

The instructor was prepared for class

Strongly disagree	1
Disagree	1
Neutral	12
Agree	10
Strongly agree	12

The instructor treated students with respect

Strongly disagree	0
Disagree	0
Neutral	2
Agree	11
Strongly agree	23

The instructor was available to me (in-person, electronically, or both)

Strongly disagree	0
Disagree	1
Neutral	1
Agree	10
Strongly agree	23
N/A	1

The instructor evaluated my work fairly

Strongly disagree	0
Disagree	0
Neutral	2
Agree	19
Strongly agree	14
N/A	1

The instructor provided feedback that was helpful to me

Strongly disagree	0
Disagree	2
Neutral	10
Agree	12
Strongly agree	11
N/A	1

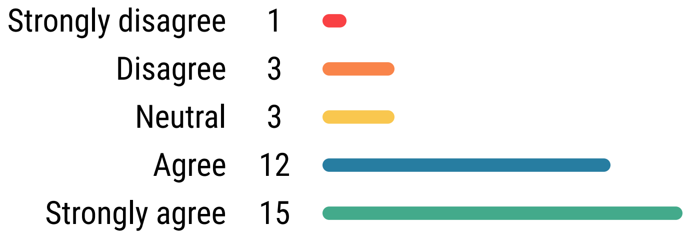
I learned a lot from this course

Strongly disagree	1
Disagree	1
Neutral	8
Agree	11
Strongly agree	13

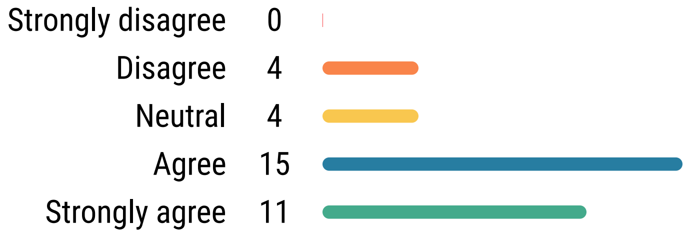
The instructor creates an inclusive learning environment for all students

Strongly disagree	0
Disagree	1
Neutral	1
Agree	15
Strongly agree	17

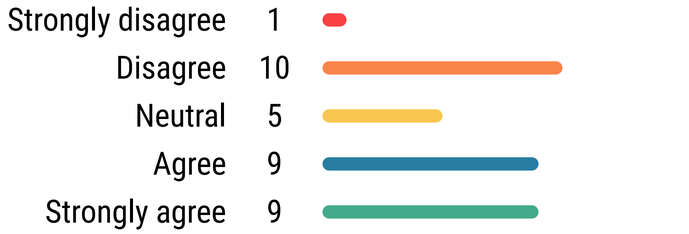
The project(s) provided a meaningful learning experience



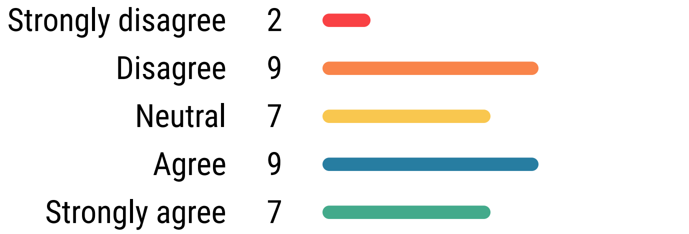
The project(s) provided a meaningful learning experience



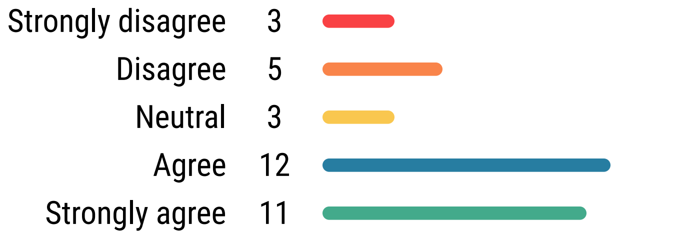
I would rather have no exams and harder/open-ended projects of real-world problems



I would rather have fewer topics and more depth (i.e., half transcriptomics and half cadd) instead of more topics at a surface level (i.e., genomics, transcriptomics, cadd, simulations)



If this class had CS 0011 (Introduction to Python) as a prerequisite, would you still have wanted to take this course if it meant an additional class?



This course should be split into computational biology major and non-major sections

Strongly disagree	1
Disagree	2
Neutral	2
Agree	9
Strongly agree	20

How would you rate the relevance of the course content to your personal and professional goals? (Strongly agree = Strong relevance)

Strongly disagree	3
Disagree	4
Neutral	8
Agree	12
Strongly agree	7

What did you like best about how the course was taught?

- Learning about the newest technology so we are up to date
- How modern it was. Computational Biology is a field that is always changing and Alex really leaned into that and did well in making the topics relevant and applicable to what would happen in the real world.
- I enjoyed the last two checkpoints. I think that if you do keep teaching this course, incorporate more work like that.
- I liked how engaging Alex was. It was evident that this course is what he is passionate about and it showed in the lessons. I also liked how he was very open minded when it came to feedback and adjusted assignments based on the needs of the students.
- I liked the breadth of information and how the material actually directly applies to a lot of current research.
- It was project heavy, which was nice because it had a real world application to it.
- I liked how the assignments and checkpoints enhanced the learning from the classroom.
- taught interesting material given in a productive way. made an atmosphere that was comfortable.
- The adaptability and flexibility of assignments and course material. He had a good understanding quickly what we were capable of and what we were not and how to adjust to it all. However, he also gave options to expand on learning for those who wanted a little extra.
- The quizzes being open note
- More weight on checkpoints and assignments
- Even though the professor was not always prepared to teach the class or if the lectures were lackluster, such as the salmon day, he was available for office hours M– F, excluding Thursday, which made it much more tolerable and enjoyable. Note: The professor was only given two weeks to prepare for the class and make up a syllabus, so I don't really blame him on this.

- There were many opportunities to get things right.
- The outside resources which were linked in the syllabus.
- I like how flexible he was in changing lesson plans if he saw we were struggling
- I enjoyed the wide variety of comp bio topics that Alex introduced us to because it is evident he is passionate about the course content and wants to share it with us. I also learned a lot from completing the checkpoints and was proud of the product I ended up submitting because of the amount of effort I had put in. I know a lot of students were critical of the way the course was taught because they didn't feel equipped to search on their own for documentation on scikit-learn etc. I think a lot of this had to do with the fact that many students had never done coding of any sort before. Alex did a great job adapting to this while still maintaining aspects of comp bio for us to be able to learn.
- Learning about how we can use computers to create and discover drugs or do science generally was amazing.
- how it was lots of projects and not tests
- I love how the course was altered to help all of us. It is something that I greatly appreciated and made me enjoy the class more than I already did.
- I liked being able to have hands-on experience with the docking simulations and Python. The reason I took this class was because I wanted to have some experience with Python and how it works in a lab, or bio kind of setting. I guess I got some of that, even though I struggled heavily with how to "do" Python in the first few classes (I have experience with C++; they're really different…). I also liked how you outlined the assignment criteria very clearly, and responded relatively quickly to emails when I had questions.
- I liked the level of python that we used I think it was a good understanding of the system without having to actually know how to code.
- I liked the structure of how there weren't any exams. Especially having other hard classes this semester, it made me feel better.
- I like the wy the professor taught the class, he tried to be kind and polite to students and answer everyone question.
- I liked the structure of the class being lectures, homeworks, projects, and quizzes
- I enjoyed the projects because they helped reinforce what was learned in class
- Step by step help with understanding concepts of python for the ones unfamiliar with it
- I liked the flexibility of this course and how you adhered to the students. I think one dropped quiz takes a lot of stress off the students and the checkpoints were very fair and not overly difficult.
- the concept of having a portfolio was initially appealing. Prof took feedback and incorporated it into class. Using tophat to anonymously ask questions was really helpful. Discussions during class was useful and engaging. Open note quizzes with plenty of time to complete them. Time to work on checkpoints during class was VERY helpful
- The slides used a lot of real examples, including going through programs in class to better understand their function.

If you were teaching this course, what would you do differently?

- If wish it was more hands on actually learning how to do the programs we were describing instead of just learning how they worked from a chem/bio standpoint
- I would set expectations a bit lower. I felt that on quizzes or projects that I would spend 60+ hours on, and I don't think that the projects needed to be that stressful. Alex and the teaching team was very helpful, but I felt that on my own, I had to do a ton of self teaching outside of the classroom. Which is usually okay, but the transcriptomics checkpoint legitimately took me over 100 hours to complete.
- I would offer a textbook (but I do know that a comprehensive computational biology textbook is hard to come by).
- Give students answers to the practice problems before exams and provide study guides. Be more clear about what would be on exams and teach more efficiently.
- I think I would structure it entirely differently, I know this was the first year teaching it but I couldnt find helpful resources to learn outside of class. The lectures themselves were not very informative either, and nothing could be learned from going over them again outside of class.
- I think this happened only because this is a new course, but the expectations for what we as students should be able to do varied widely throughout the length of the class. I would make ideas of expectations clearer to students and try to stick to them as much as possible.
- I wish the syllabus did not change every couple of weeks and the rules for exams were changing every time. I also wish it was more programming based.
- I would have more clear writing on the lecture slides, so its easier to understanding when looking back after lectures and studying.
- more explanations on what will be on exams. more study materials. more coding in a computational biology class!
- We've already talked about this but it honestly should be 2 courses––one python and one not. They both have their benefits but it is hard for a bunch of bio majors to just go into the class knowing python at the level that may be needed for the best code. Other than that I would have a few more assignments only 5 felt too little and practice problems for quizzes which I know is not a lot but after this class, it should be kind of possible. I would also just make reading mandatory as they really help even if its just like you have to answer 5 questions on top hat for an assignment that is in the reading just to clear up some confusion come lecture. If we can't do 2 courses just make the checkpoints have a python and excel/sheets option. (I would still go over some python it is useful) And for CADD keep it MolModa.
- I would give more homework assignments to ensure everyone is prepared for quizzes
- Less confusion with python at beginning of semester
- I would change the prerequisites. The class prereqs should also include CS 0011 and STAT 1000 to get a basic understanding of the course. If that's not allowed, I would attempt to either separate the classes into two versions or give a prerequisite quiz on the topics as a test to see if you're ready for the course. Also, within the textbook, over the summer, I think it would help students greatly if you put sample problems on important topics(the videos you made in Unit 1 were invaluable

to my understanding of the quiz). For example, on a greedy algorithm section, embed your video lesson from unit one on the page and put practice problems below. I believe this would seriously increase student understanding. I think what made this class so much harder is after unit 1, there was significantly less written content, mostly just blank sections with the header “TO DO.” Overall, you were an excellent instructor within time constraints and showed me how much I truly enjoyed this subject.

- Organization and expectations for students.
- I would use Canvas. Utilize more powerpoints and be more organized when it comes to day to day activities and goal for upcoming exams and assignments.
- I would go into more detail about fewer topics and make sure that we have a solid background of the material. I would also maybe do more hands–on activities so we can see what we’re supposed to do for assignments.
- I think unfortunately some students were still discouraged because up front the content can be intimidating. There is a lot to learn and people who haven’t done coding of any kind before can have more trouble ignoring the things that they don’t need to understand and focusing in on what they do need because it’s hard to know what to look for and what to ignore (for example all the different available parameters in the sklearn documentation for the classifiers that makes it hard to understand what you actually need to use in order for it to work). I think a lot of students aren’t used to searching online for support with coding because they’ve been taught not to rely on online resources that are not endorsed by the professor. I don’t know how plausible this is, but I think that in the beginning of the semester emphasizing that online support is a necessary aspect of this course and that it is normal to have questions and need support with coding and to get errors and how to resolve these errors on your own will be helpful. However, I think a lot of the issue which is difficult to prevent is that there isn’t a coding prereq for the course.
- This course was brand new to me, and I had never seen any of the content in this course before. At the beginning of the course, I would go slower into Python because not everyone knows what that is. In the beginning, I would teach and go deep into the material to ensure every single student grasped the course concept.
- maybe make more consistent assignments so we keep up with content better
- Considering comp bio is a new field with no basis of a textbook for an intro class I would not do anything differently. I can tell that Alex put a lot of time and effort to all of his materials and assignments so they are clear and make sense.
- I think maybe assigning us Python courses from the platforms you suggested would’ve helped with making us learn Python, especially since I tried one of the intro ones and it didn’t seem too time–consuming. I understand why you had us practice via your assignment questions, but I think the Python courses from the platforms were better at contextualizing and teaching us the content. Assignment–wise, I think criteria were outlined pretty well, barring the times that 1. the criteria wasn’t listed, and 2. the documentation for the assignments weren’t there. The second one is particularly directed to the CADD unit, with little to no documentation on the website. Yes, you gave us textbook material, but it doesn’t give us the exact phrasing / information you might be looking

from us, and it made the CADD checkpoint, particularly the docking segment, hard to do. Because I wasn't exactly sure what you were looking for, and I felt like I just repeated myself for five questions, but in different ways.

- I understand that this was one of your first times teaching this class, and I think you did a relatively good job!! But sometimes things just felt so disorganized that I just felt like I was floating through class. Also, sometimes the quiz questions didn't really feel like they reflected lecture material; I think I was pulling from my genetics knowledge for the first quiz, when it came to sequencing methods and things like that.
- I would utilize class time differently, I think that class time was wasted on topics that were not needed in that moment, the timeline was skewed for what assignments were upcoming. I think that the material covered in class did not properly reflect what we were tested on. It might as well have been an asynchronous course with how much out of lecture teaching there was. Which yes, in most courses students need to research and learn on their own outside of class, but in this course all the learning and teaching was done outside of class. The information given in lecture was practically just a guiding point to know what topics to look for. Literally equivalent to a study guide, maybe less than. There was also too much information to sort through and completely understand how everything fits together.
- Understandably, it's hard to know what to teach when a course is so new, I just wish instead of him trying to explain for the students who are taking this as an elective, he focused on helping those who are in the major.
- I think if i were teaching this class, I will make two classes, some people were very familiar with phyton and other computational sciences and there were people like me, that everything took extra longer. if this class was two sections one for people familiar with this and one for people who had zero familiarity, it will be better. i don't think this is a mistake of the professor but of just being the first time this class was taught and no one was familiar or could research how the class would work.
- I think if i were teaching this class, I will make two classes, some people were very familiar with phyton and other computational sciences and there were people like me, that everything took extra longer. if this class was two sections one for people familiar with this and one for people who had zero familiarity, it will be better. i don't think this is a mistake of the professor but of just being the first time this class was taught and no one was familiar or could research how the class would work. iMAGINE,
- I wish the lectures had more content on the slideshow so I could refer to something from lecture if I missed it in my notes. I would also increase groupwork as this class had a large learning curve, especially because so much of the content revolved around python without us actually being able to fully use python. I wish there were smaller and easier projects that would allow us to use minimal python as it would allow us to be more immersed in the field. I would also focus on some content rather than attempt to push it all out because it made the content very confusing since I had no background knowledge on the subject.

- Record lectures. The content was very dense and it was hard to write everything down/retain information
- Strongly advise people to learn programming language on the side so that work isn't overwhelming
- Some people may not agree with me on this, but I liked the freedom on the second checkpoint. Instead of specific outlines, you allowed us to be creative with what we learned and draw our own conclusions. It was still enough outline to make sure we did everything needed, but still gave us a lot of freedom. The resources you have on the website were more than enough to help us through that checkpoint. I would give more assignments like that if I were teaching.
- I would forgo a class website and instead have all resources laid out on the same page on canvas. Website is stressful to navigate and not beneficial. I would record lecture and include more material on lecture slides. Lecture slides with predominantly images/screenshots are not useful. More text is needed. I would use Tophat more effectively and purposefully for class participation. I would include more lectures and guidance dedicated for how to complete checkpoints including how to code and what the code actually represents, what your expectations are etc. I would include more practice material, support in general. such as TA lead review sessions before quizzes and more drop in hours for checkpoints.
- I believe some type of way to review for the exams is necessary, either through multiple choice questions with an answer key, or homework assignments similar to A4 and A5, which are more theory–based questions throughout the class.

What are the essential qualities of a successful instructor for computational biology courses?

- Meet the students at where they are
- Availability, Knowledge, and Efficacy. All of which Alex had.
- You have to know how to code, but also have a good content foundation in the field of general biology.
- Is able to explain things on a simpler level to students.
- Someone willing to help out their students as well as help them find opportunities within the job field and related job fields
- Willing to explain detailed concepts. Willing to explain basic concepts/not assuming students have background knowledge in the subject matter. Understanding students may not be using computational biology in their career. Being available for questions. Consistent with expectations. (you already do a lot of this!)
- Understand the students in the class and maybe emphasize that the class is programming heavy.
- Open mindset when teaching students, as this is a newer field we are diving into without much of background due to a large focus in Java and Biology courses that are not necessarily very relevant to the Python, R and specific topics of computational biology not covered in the major curriculum.
- knowledgeable on the many topics that encapsulate the course. being able to accurately answer questions.
- Flexibility, Time (Sry), Enough knowledge to dumb it down for some. A great knowledge of food. An understanding of students' issues.

- To slow down and explain
- Strong knowledge of what materials students can use to learn since there is no one concrete textbook
- The professor does not have a problem with this.
- knowing limitations
- The ability to explain abstract concepts in an elementary way.
- Someone who is flexible and able to break concepts down to understand them easily
- Adaptability, patience, and the ability to determine the difference between when students truly feel incapable of understanding course content and when students aren’t trying to understand something because it requires a different kind of learning than the learning they are used to. A lot of students I think were unable to get over the hurdle of taking a class that felt so different from one they took before and thus weren’t able to see that once they accept that it is a different style of class it isn’t necessarily that difficult.
- Don’t expect students to know so much of the material in the beginning. I came to class with no knowledge of this course.
- knowledge, passion, communication
- Willing to work with students and able to explain these complex concepts in a way that can be understood by people new to the material
- Able to explain coding and other biological concepts succinctly and clearly, able to clearly outline expectations for assignments and tests.
- Someone who is in the field/ has worked in the field for a good amount of time.
- good communication (Prof.Alex has good communication). accesibility (being available to students questions). flexibility
- Being able to fully breakdown code and computational biology concepts on an easy to understand level.
- efficient in teaching information, simplifies information enough for students to understand
- Patience. It may seem overwhelming non–compbio students, but to the ones you helped, being patient allowed us to appreciate the real word computational aspect of python
- Understanding with students not being familiar with coding. Coding takes more than a couple classes to learn, so it helps if there is an understanding that we are not experts.
- A prof who has the ability to think like a student and someone with no prior exposure to the topic and can empathize with students who are learning the material for the first time. This involves limiting the use of Jargon, explaining acronyms, using visuals/drawings to explain concepts, explaining things in multiple ways for students to understand. A prof who can systematically present information in a clear step by step manner, including why things are done such as lines of code or sequencing technologies
- Ability to summarize objectives/ reasons for a process. Open and available communication, flexible class layout (able to spend more time on a subject students are not understanding).

How likely are you to recommend this course or my teaching to other students, and why?

- 6/10 You tried your best but a lot of the work took an unnecessary amount of time to reiterate the same information over and over again
- I don't know about how likely I'd be to recommend this course to non Comp–Bio majors as it is very technical. But I'd recommend Alex's teaching to anyone in STEM as he's been one of my favorites thus far at Pitt.
- I would highly recommend this course because you are actually applying biological concepts to work, not just listening to information you've already heard before (and that's the end of it).
- I thought your teaching was good, but it did not really help me prepare for exams and you did not provide enough resources to study for exams, which is what I struggled with the most.
- Very likely for other computational biology majors as you provided much help for those in the major.
- not likely at all, as a non comp bio major it was a really unapproachable class.
- I would recommend your course to others with the idea in mind that it was your first time teaching it, and that future iterations of the course will probably go more smoothly/be more consistent.
- VeryI would be very likely to recommend you as a teacher to other students. I think you're grading and syllabus were more than fair and very helpful to students with other difficult classes. likely, but with tweaks in the teaching style and syllabus.
- Highly likely; You created an environment where we could ask questions and not be afraid to ask for help or change checkpoints if we were struggling with aspects covered.
- very likely! very knowledgeable and welcoming to discuss class and future plans
- I would definitely recommend it to comp sci majors with an interest in biological problems. As for bio majors, I would still recommend it but with a caveat that it will be a slow start and it is different from anything you have done so far. I think this way you get more students that come in interested to learn and a good distribution between majors is set
- I would 100% recommend, you can tell Alex cares about his students and their learning.
- I would recommend to other students who are interested in comp bio specifically because it's a good learning experience about the field
- In the current state: 6/10. But I'm sure over the summer when the textbook is completed, and the class is more blue–printed out, it will be an invaluable course for computational biology majors. On the "I would rather have fewer topics and more depth (i.e., half transcriptomics and half cadd) instead of more topics at a surface level (i.e., genomics, transcriptomics, cadd, simulations)" question, I don't think you should remove content as this is the first main computational biology class and not an advanced elective and students must get a general introduction to major areas in the field before advancing in one particular field. Maybe look into making a CADD–specific HL elective class.
- Slightly likely because it was a difficult course with a lot of coding and machine learning. These were things not stated in the prerequisites. However, if the goal were more clarified I would probably recommend this course because there are many useful skills which can be taken from this course.
- As this course was this year, I would not recommend it. However, I know that there will be a lot of changes made and I think it could be a very engaging class if it was a little bit more organized.

I would recommend your teaching since it was very flexible and obvious you care about your students.

- I would be wary of recommending the course to anyone who I don't think adapts to change very well or isn't familiar with the challenges that come with coding initially. However, I would recommend it to anyone interested in comp bio because I think the course does a great job of introducing what it is like.
- I would recommend your teaching because, as a student, you are a very fun professor. You make the environment of the class friendly to everyone, which will make students want to learn from you. I would not recommend this class to anyone if they don't need it. This course did not catch my attention, and it was never of my interest. It shouldn't be required for the Biology major.
- very likely because you are young and understanding of students, so i have no worries you will adjust your courses to accommodate for mishaps.
- Very, Alex has been my absolute favorite professor at Pitt and will be pushing everyone I know to consider taking this class specifically with him. I only took this class because he was teaching it rather than the other comp bio professor.
- I recommended this course to a few friends because I liked how this class felt more application–based rather than just memorizing things.
- Very likely.
- I would recommend the course as it is, if you have familiar with phyton. OR give you advice to prepare with some phyton before taking this class at least the basics so you could be able to get all those extra credits.
- Semi likely, I felt like a lot of the class required me to have previous knowledge on the content and I was often left grappling with what I was supposed to learn or what was to be expected on the assessments, and the open resources were often overwhelming and confusing as well.
- Yeah, probably would. I think there are some kinks that need ironed out in terms of course structure, which i'm sure Alex will agree with, but once that is done I think this course will be incredibly useful for those interested.
- Not unless they are interested in comp bio
- Still very good recommendation to anyone willing to diverse their computer major with biological aspects. I would still warn those with limited programming experience since its pretty much learning a new language + academic rigor of classes
- I would be very likely to recommend you as a teacher to other students. I think you're grading and syllabus were more than fair and very helpful to students with other difficult classes.
- Likely, your teaching style is challenging but if you are willing to put in the time and visit outside of class for issues you are having you can develop a deep understanding for the course.

BIOSC 1630

You can find a description of this course above.

Fall 2024

I received evaluations from 19 out of 21 enrolled students, representing an 90.5% response rate.

The standards the instructor set for me were

Too low	0
Appropriate	16
Too high	3

How many hours per week did you usually spend working on this course outside of classroom time?

Less than one hour	0
One to three hours	4
Four to six hours	12
Seven to nine hours	2
Ten or more hours	1

The instructor created an atmosphere that kept me engaged in course content

Strongly disagree	0
Disagree	2
Neutral	2
Agree	5
Strongly agree	10

The instructor was prepared for class

Strongly disagree	0
Disagree	1
Neutral	1
Agree	6
Strongly agree	11

The instructor treated students with respect

Strongly disagree	0
Disagree	1
Neutral	0
Agree	6
Strongly agree	12

The instructor was available to me (in-person, electronically, or both)

Strongly disagree	0
Disagree	2
Neutral	1
Agree	5
Strongly agree	10
N/A	1

The instructor evaluated my work fairly

Strongly disagree	0
Disagree	1
Neutral	2
Agree	9
Strongly agree	6
N/A	1

The instructor provided feedback that was helpful to me

Strongly disagree	0
Disagree	0
Neutral	2
Agree	5
Strongly agree	11
N/A	1

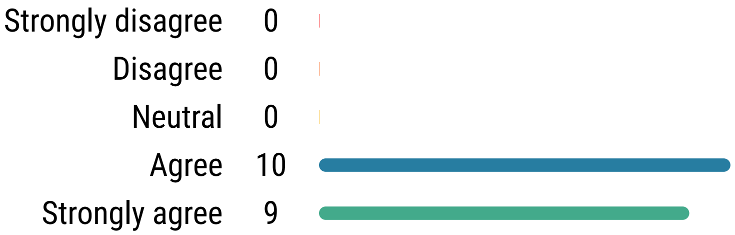
I learned a lot from this course

Strongly disagree	0
Disagree	0
Neutral	2
Agree	6
Strongly agree	10

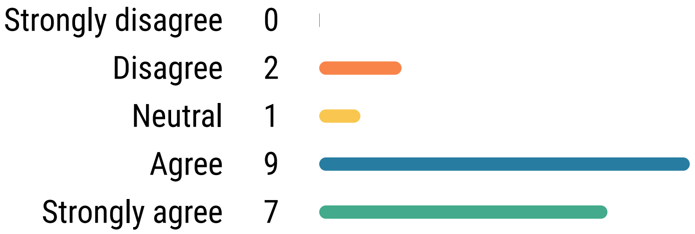
The instructor creates an inclusive learning environment for all students

Strongly disagree	0
Disagree	1
Neutral	0
Agree	3
Strongly agree	15

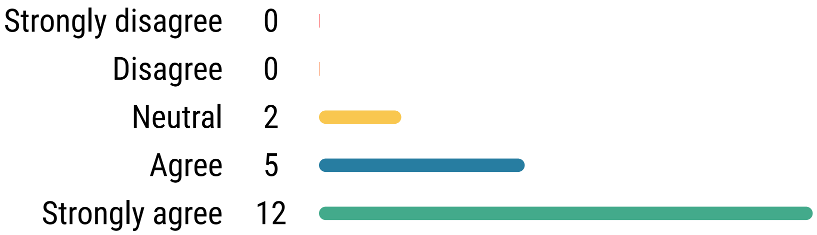
The course was intellectually challenging



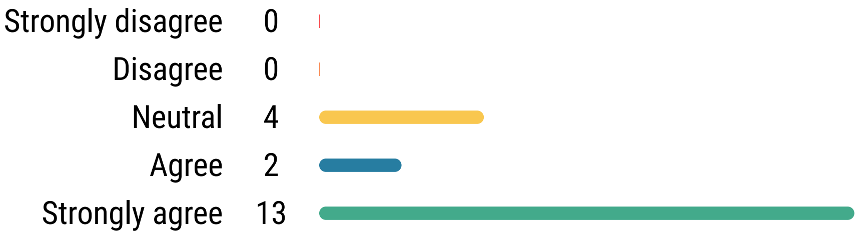
The course stimulated my interest



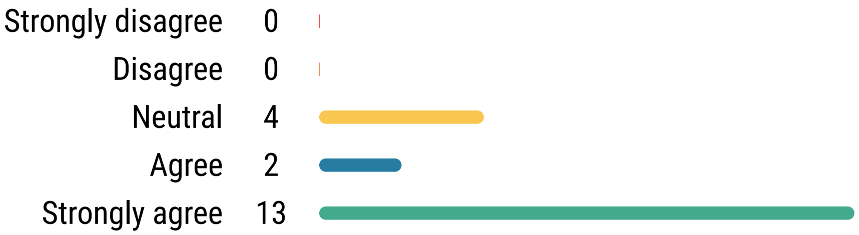
I could ask a question or make a comment if I wanted to



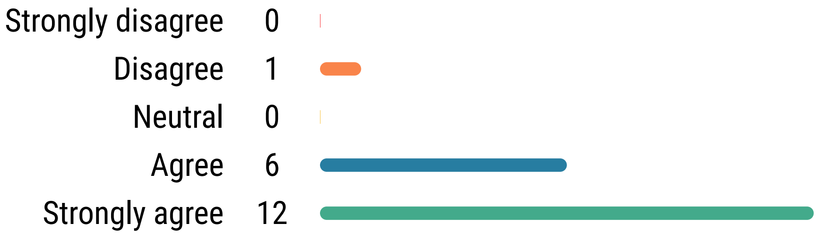
I felt comfortable expressing my perspectives related to the class materials



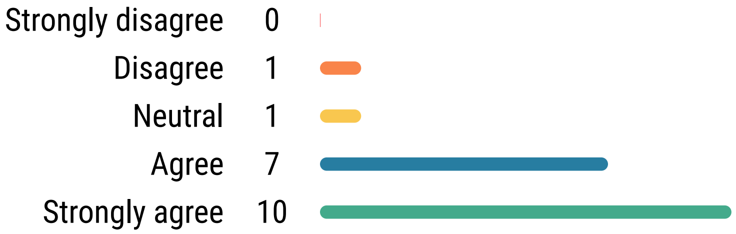
The instructor maintained an environment where students felt supported asking questions and seeking assistance



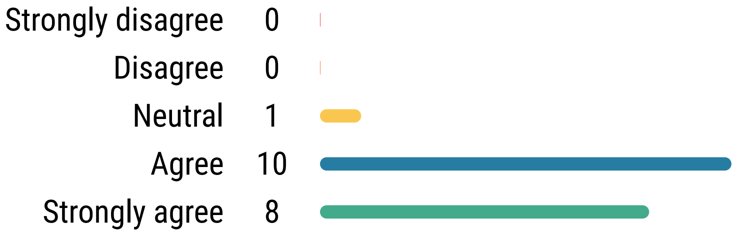
This course helped me learn to identify main points and central issues in this field



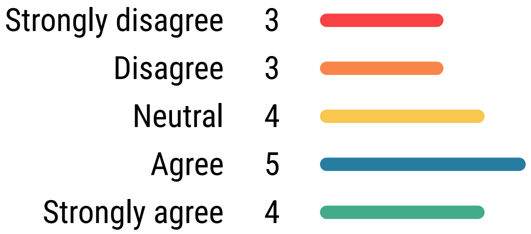
This course enabled me to read research in this field with understanding



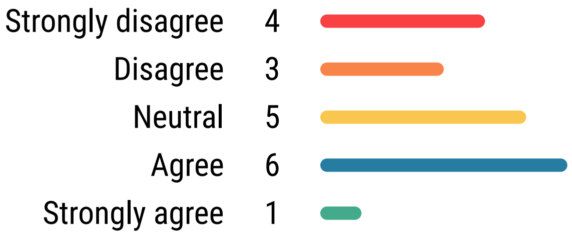
This course helped me to improve my critical thinking skills



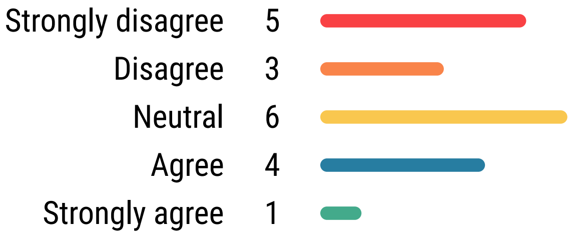
The computational biology major curriculum effectively integrates biological concepts with computational methods



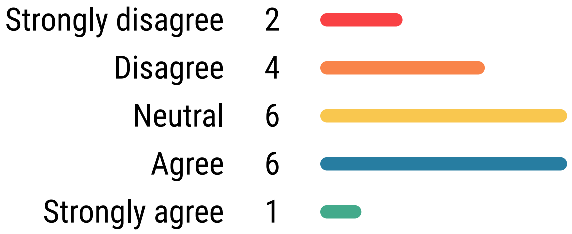
Courses for the computational biology major include an appropriate amount of practical applications



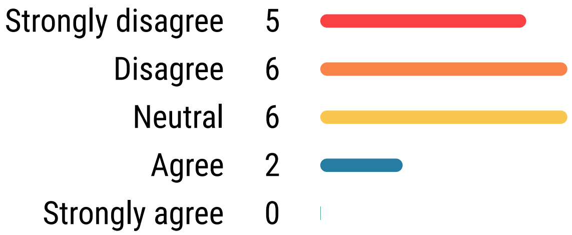
The career planning guidance available for computational biology students at Pitt meets my needs



Research opportunities in computational biology at Pitt are readily available to students



Industry connections are readily available to computational biology students



What did you like best about how the course was taught?

- I like that the course centers around writing our own perspective paper, and the activities that revolve around reading and presenting the contents of published papers.
- I enjoyed the presentations, the fact that we were given adequate time to prepare, and the tips we were given afterwards on how to improve.
- Alex is one of the best professors I've had at Pitt. As a student, I felt that he truly wanted me to succeed and gain skills I could apply in my future, regardless of where my path takes me. I enjoy how the structure of the class encouraged collaboration and curiosity through the group presentations on Comp Bio papers, without placing emphasis on needing to know and understand everything. I never felt afraid to ask questions, and Alex would work with me until I felt I actually understood the concepts being discussed in the papers. Furthermore, he provided great resources and instructions for how we could structure our perspective papers, as well as detailed feedback on where we could improve for every individual part of the paper. The structure of the class ensured that we had ample time to put our papers together, and that we wouldn't wait until last minute to try and write such a large piece. I'm not someone who usually has the time to revise my work because I wait till last minute to do it; by having deadlines throughout the semester, I was forced to prioritize my writing and then had time to fix it.

- I liked Alex’s flexibility and knowledge in many fields to help explore different papers and fields within comp bio.
- The course gave me a good understanding of the computational biology field as it currently stands, especially with focus on the subtopics I want to pursue. It helped me get better at reading scientific articles and improve my own scientific writing through the perspective paper assignment. Group work for paper presentations is well structured so that students have a good understanding of the paper despite them sometimes being very complex. Expectations for assignments are always very clear, and Alex has been willing to meet students where we are if there is a mismatch in what he expects and what we can deliver at the moment. Alex’s feedback is always very thoughtful and thorough, and he engages with students often to discuss what is/isn’t working about how he is teaching the class. He has been very flexible and has prioritized student feedback in order to meet both his teaching goals and provide what we are looking for from the class. Alex has cultivated a class environment which is comfortable and encourages discussion and collaboration. This allows students to freely express if we have concerns about any assignment or if we wish to take the class is a specific direction (for example, we are allowed to discuss deadlines and decide what kinds of papers we would like to learn more about). Alex has also made efforts to bridge the gap between the minimal coursework on computational biology methods most of us had prior to this course (earlier sections of 1540 did not prepare us well for the CADD topics we’ve been discussing in this course, and many of us have not yet taken our higher–level computational biology course). This has been largely by giving pre–lectures on a paper’s topic so that we have a foundational understanding before reading the article and encouraging pre–lecture assignments to skim the paper.
- He provided detailed and thoughtful feedback on assignments, which was very helpful. That said, the feedback was often given two days before or even after the next assignment’s deadline. It might be even more beneficial for students if feedback were provided about a week earlier, as this would allow ample time to thoughtfully reflect on the comments and make meaningful improvements to the next assignment.
- I like that our final paper was divided into multiple smaller assignments in order to encourage progress throughout the semester.
- I liked the regular feedback being given after each assignment. This helped me improve in each assignment.
- It taught me a lot of things that I thought were directly applicable to my continued education / career. I was never formally taught how to read academic papers or write them, and the skills I learned through Alex’s instruction has helped me tremendously outside of class in my own research experiences.
- I liked the level of input that students got to provide to class topics. For example, getting a say in topics we wanted to see in the papers we read, etc. I also liked the level of in–depth feedback we received on our writing assignments.
- I enjoyed the lectures, the presentations made me nervous.
- I liked that the writing assignments were distributed over the entire semester, and we received feedback. It made it less stressful to complete the the final draft submission.

- I throughly enjoy the setup of the course, with reading a different paper some weeks and presenting other weeks. I think it kept the course interesting and I also enjoyed all the open discussions we had.
- I liked that we worked on our perspective paper throughout the semester, so that we could work on parts of it before putting together the final submission. I also enjoyed working in small groups to understand comp bio literature. I think this was helpful to my understanding because it is always more demanding to try and find a way to explain something to others than to simply be tested on it. I feel like it also allowed for more collaboration and overall better retention. I also greatly appreciate Alex's decision that if we are happy with our draft grade he will use it for our final. I think it makes a lot of sense.
- I liked that the course was set up with checkpoints during the entire semester for our perspective paper we needed completed by the end of the term. It allowed for ample feedback and guidance that was thorough and very helpful. More specifically, I have never written a scientific paper before. Additionally, I have read scientific journal articles on my own time, but have never had the opportunity to deeply read and understand these articles in other classes. Hence, having the checkpoint assignments for each part of our paper, as well as the professor's thorough and clear feedback and guidance, greatly enriched my understanding and writing. Furthermore, having activities in class where we had to present and also read articles given to us that day in lecture was difficult, but is a necessary skill to learn. The way the course was taught gave me resources for the future that are so valuable.
- I really appreciated the way Dr. Maldonado would give us time to go through a research paper on our own, and then in groups focus on specific portions of it that we would present to the class. I felt that this was really helpful because it allowed us to individually dive deep on portions of the article, but still understand the whole thing since the presentations became very approachable and understandable since we had a very similar knowledge level of the content.
- Class time was structured fairly and well. I liked the topics taught a lot and found it extremely relevant to the field. This is the first time at pitt I felt like this was true for computational biology.
- The course structure was very good, I personally found a lot of value in how we were required to read, understand, and evaluate the strengths and weaknesses of papers during class. For me, this was very good practice in critically engaging with scientific literature, and I feel that my ability in this domain has greatly improved. The guidance given on searching for, engaging with, and presenting scientific literature was invaluable, and I feel that I am better prepared for my future career.

If you were teaching this course, what would you do differently?

- I would give some more time, maybe one or two extra weeks, to students to find the topic of their perspective paper, in the hopes of having each student narrow down to a specific perspective. I feel that once students have a thorough understanding of their specific theme and a logical flow of how to go about arguing this perspective, the writing portion will be much easier and less time will need to be given.

- I would say the time until we received grades was very frustrating. Although we had peer review feedback it just never compared to the feedback you gave us. I do appreciate that you would give us extra time when the feedback was late. I also didn't understand the grading since sometimes you would grade some parts of my paper with no context from my previous parts. So then you would dock points at moments where you needed context from other parts of my paper.
- I think Alex was still finding his rhythm at the beginning of the semester and that lead to us not having enough time to really put in our best work for the first couple in class assignments. However, he was really understanding and receptive when we brought up that we didn't have enough time, and was able to somewhat restructure the class to give us the time we needed. I think for future semesters, he could spend more time in the beginning of the semester teaching skill like how to give a presentation, create slides, and write a perspective paper (maybe have the class read a perspective on something related so they can understand the format).
- nothing!
- I would not make major changes to the way Alex has taught this class –– I have appreciated the changes he's made and that he's been very intentional with his decisions in teaching this course. Based on feedback this semester, I believe he will be able to design a better syllabus and schedule that meets students' needs in the future.
- I would encourage students to explore a broad range of topics. While he did not explicitly prohibit selecting topics outside his specialty in molecular dynamics simulations and chemistry, he did not seem to actively support such exploration. While I respect his expertise in molecular simulations, I believe it is particularly important in an undergraduate seminar to foster an environment where students feel empowered to pursue diverse interests within computational biology.
- Depsite the fact that the smaller assignments helped encourage progress throughout the semester, it also led to me being significantly less knowledgable about the topic I was writing about for the earlier assignments as compared to those that came later. This meant I had to change the sources I found/used multiple times, as well as adjust my introduction, etc. It may help save students some time to try enforcing a stronger understanding of the paper's topic at the beginning, rather than letting it continually grow while working on later assignments.
- I would explore more concepts in computational biology
- I thought Alex did a great job.
- I think I would be more consistent with assignments and grading. There were several pre–class assignments in the second half of the semester that were on the class website but never updated or addressed in class. Also, grades and feedback could sometimes take a long time to be returned, which is critical when sections of a paper that build on each other and are being graded rather harshly. Getting feedback from one section on Sunday at 2pm, when the next section is due midnight the same night was hectic and disorganized.
- I wish we could switch the groups around me, I sometimes felt left out.
- Since most of the students in this course did not get an adequate overview of comp bio from taking BIOSC 1540, I wish we had spent more time learning about different computational biology techniques, and then reading papers that showcased how these techniques are put into practice.

I did not like how we split up into groups to present different portions of a paper because I didn't understand the paper's contents after the presentations.

- I would most likely have more lectures going through the methods described in the papers we read.
- I understand that this is difficult given the volume of students and fast pace of the semester, but I feel if we had started writing the paper sooner and gotten feedback on the individual portions we had written, it would not be as difficult to make all of those edits last–minute before the draft was due, meaning that more of us could have potentially submitted drafts that we felt addressed Alex's concerns. I happened to have enough time that Sunday to address Alex's analysis feedback that was published to me that day, so I did my best to address almost all of his comments, but I do know others in the class struggled with having little time to address the feedback. Alex did mitigate this by reopening the draft for more days, but I think some of the students that were stressed about not meeting the deadline may have been more relieved to know that information sooner.
- I would not change how the course was taught.
- I would just go over more perspective papers earlier in the semester.
- Focus more on how to write, this was my first real scientific writing course, and I already knew pretty well how to present slides and papers, but have never written a review before.

What aspects of the computational biology degree/program could be improved to prepare students for their careers?

- I think there should be concentrations for the major. Because I am pre–med I enjoyed this major because it gave me the flexibility to pursue something I was interested in while allowing me to take my prerequisite courses for med school. However, I do believe if someone wanted to go straight into the field, or even go to grad school some of the courses offered wouldn't benefit that much. To excel in this career we should be required to take linear algebra, intro to machine learning, etc… In general more higher level CS classes that teach us python instead of java.
- I did not take Intro to Comp Bio with Alex and I don't think I really understood what the field of Comp Bio is until taking this seminar. He did a great job introducing us to the major themes of Comp Bio by focusing on specific papers from the field. It's somewhat unfortunate that this class is usually taken by seniors, because I feel like many students entering the Comp Bio major haven't been exposed to what the field actually looks like and what they would be doing upon entering the field. I choose this major because I like coding and I'm interested in med school but wanted a backup path in case that doesn't pan out. To me, Comp Bio seemed like it should provide some job security. However, I'm now unsure if I would actually enjoy doing the types of things that the Comp Bio field entails. It would have been nice to be given a better understanding of Comp Bio from Pitt before getting to my senior year; however, this could also be due to the Intro to Comp Bio class that I took with Brouwer, which had nothing to do with the actual field of Comp Bio.

  Also, I have no idea how the CS classes I'm taking actually relate to the major. They feel completely separated from what I'm learning through bio or comp bio classes and have no real practical applications for me. I'm taking BIOSCI 1544 next semester alongside BIOSCI 1640 so I'm hoping those classes will finally teach me something that I could use in the field, but once again I feel like this is too late in my academic career to be the first time I'm exposed to real life applications of Comp Bio.

- More focus on AI and data science, more classes with real–world projects and applications, more exploratory or helpful classes/teachers that cover different subjects and relate directly to computational bio. i.e. reading and understanding academic papers, and designing program
- I would recommend for the computational biology program to be further developed. The field is vast and the program is only growing, but there are currently few resources for students to gain a broad understanding of the the different subtopics. One suggestion is that it would be good for the 1540 curriculum to be more structured and better prepare students for these higher–level computational biology courses. The 1540 section I took had greater focus on data science and global human genomics, with almost nothing about CADD. As a result, I ended up learning most of the techniques used in this field through my research or in seminar. The program should either be refined to focus on a select few critical applications in computational biology or provide an overview of the various subtopics so students are aware of the different paths/what might be best for them. This way, computational biology students required to take this seminar can spend more time honing our scientific reading/writing skills and diving into more specific aspects of the field rather than having to spend time learning the basics.
- I deeply appreciate the Pitt Computational Biology program's strong foundation in biological sciences and chemistry. To further strengthen the program and enhance its future potential, I believe incorporating greater emphasis on physical sciences, such as mathematics, physics, and machine learning, could be highly beneficial. For example, incorporating a dedicated machine learning course would better prepare students for careers in this rapidly evolving field. Additionally, adding courses in physics and mathematics, such as PHYS 0174, 0175, linear algebra and differential equations, as part of the curriculum would provide a more comprehensive foundation. Since these courses are currently offered only as electives and not mandatory, I believe that only individuals who are already actively pursuing related fields are likely to recognize the importance of these prerequisites. Making such courses a more integral part of the program would ensure that all students develop the necessary foundational knowledge to succeed in computational biology and related fields. Integrating these topics into the program would not only align with current industry demands but also position students to excel in computational biology careers.
- Providing more opportunities/examples of ways students can use their degree progress to get involved with research and other extracurriculars while still an undergraduate.
- There can be more curriculum relating to the industry and its applications.
- I wish we learned more about machine learning and computational tools implemented into biology. I felt like the computer science core and the biology core were very separate, and too few classes focused on their intersection. We had computational biology and computational genomics, but even the application in those courses could have been further explored. I think the cirriculum needs constantly revamped to prepare students to enter such a dynamic field
- I think most of the program needs to be workshopped for computational biology. Upper level courses in Java are largely unhelpful in the field. Not many courses focus on R or Python, which

are necessary. Many programs require courses like linear algebra and machine learning, which are not required currently.

- Help them get more experiences, more hands–on learning
- The whole major feels like it's an afterthought, there aren't consistent professors teaching the courses, and we do not go into depth for a lot of concepts. I feel like I have a general understanding of the field, but no practical experience.
- I think extra elective classes need to be added to the curriculum that are highly recommended such as machine learning and linear algebra. There also needs to be less cs courses and more actual computational biology courses since the cs courses such as taking two semesters of data structures seems unnecessary. Discrete structures also seems like a highly unnecessary course since I have not used the material anywhere else.
- The only classes I have learned computational biology from at Pitt have been BIOSC 1540 and BIOSC 1630, both taught by Dr. Alex Maldonado. Other than that, none of my computer science classes integrate any sort of data analysis or biological applications, and the kind of coding we do is very different from what Alex has told us computational biologists do. As such, we are not prepared to enter the computational biology workforce.
- We are required to take core biology classes and core computer science courses, however there are very few courses, like this seminar, that are able to tie together both fields. More specifically, more machine learning based courses and less java focused and more python–based courses would be much more useful for our future careers. I think more computational biology specific courses where we can apply our understandings from other courses, would be very beneficial. Additionally, some of the courses feel like they do not flow as well together. For instance, the first computer science course I was required to take was Intro to Computing for Scientists. This was a python based course, which gave me the foundations I needed in python since I had never taken a CS course before. The next course after this was Intermediate Programming, which was entirely java–based and we were given no foundations in java, which made the course and transition to java much more difficult. Overall, I think more computational biology specific course would be beneficial.
- I think there are several improvements that could help. To start, I think the major progression sets you up to have either a really strong biology background or a strong computational background, but doesn't effectively integrate the two until much later in the major. Also, there is only one upper level class that is required to be taken, which further makes it feel somewhat like a dual degree between biology and computer science. Also, the introductory course not having a coding prerequisite hurts computational biology students, as it doesn't teach you how to apply a lot of the fundamental concepts that we learn about.
- You seriously need to listen to people with experience like Dr. Maldonado. Especially since the school of medicine has such an amazing Comp Bio phd program. It is actually ridiculous how terrible the undergrad major is right now. I learned anything and everything practical from the school of medicine by doing research. Please stop screwing people who want to get into this field but do not know enough originally to know if the program is good. Listen to Dr. Maldonado. He is genuinely the only person/ class where I learned something useful. There needs to be major

change here. Without my research opportunities in the school of medicine (which are few and far between in terms of opportunities, I got really lucky), I would have basically just been useless in the job market. Fix it please.
- The education in python programming feels very disjointed. The order of courses required for CompBio feels off, and each course feels as if python programming is tacked–on or isn't taught comprehensively. For my research, python is essential, and I have a lot of confusion as to why the major is structured in this manner with respect to python.

What additional resources or support would enhance your learning experience if you wanted to pursue computational biology as a career?

- I would say requiring students to do research for credit. You need field experience to really gain an understanding of what the field is, as it is very new and still growing. It would be great to have hackathons or events geared towards comp bio students.
- I would have liked coding classes to be more major specific, or have options for coding assignments to relate to comp bio applications.
- Computational biology does not seem well supported in most places, so honestly any additional resources such as people to talk to in the field, and earlier/better teachings of what jobs computational biology is good for. There are many options with the degree, but I wasn't sure about most of them until toward the end of my undergraduate career
- I would integrate more project–based work into the curriculum so students can gain first–hand experience through experiential learning. I think it could also be good to do more partner programming with the Dept of CSB, graduate–level programs we offer at Pitt, or invite presenters from industry to familiarize undergrads with the various paths they could pursue with computational biology. For example, I know D E Shaw did a talk at CMU some weeks ago –– it would be nice to have these sorts of events available to undergrads as well so we can learn more about the field in academia and industry.
- More applications to real industry use of computational biology methods in courses.
- I would want to use more software relating to the field.
- earlier talks about graduate school preparation. within the first or second year to give students time to find research / mentor opportunities. Talks about what you can do with your degree
- More resources outside the classroom would be helpful. An increase in research opportunities, relevant workshops, and industry connections in things like career fairs would be a good place to start, as well as an increase in course offerings.
- More integrative classes where students can see the practical relevance of computational biology techniques. I can create data visualizations and do rudimentary stuff with unix, but I still don't have an idea of what I can do with the major. Additionally, we take a lot of intensive computer science courses, but I have not used the knowledge I've gained from them in a biologically relevant way. I would like to see more integration of the mandatory CS courses with an application to biological research.

- I would like more mentors who are knowledgeable about the field of computational biology such as Dr. Maldonado to be readily available. I also would like more computational biology courses and less cs courses that I feel do not help me.
- I think that more coding prerequisites for comp bio classes would be good so that the comp bio classes themselves can use coding and aptly prepare students to apply computational methods to topics in biology. Additionally, many students are unaware of the preparation they would need to pursue a comp bio career, like the research they would need to undertake, how to prepare for interviews, the extracurriculars they should be doing like preparing a portfolio or doing their own projects, etc. I am a pre–med student and at Pitt what we need to do to apply to med school is very well outlined for us so everyone knows exactly the things they should be doing as an undergrad and that has prepared me very well, but I don't think I would be prepared to enter industry in comp bio. I do now understand that most undergrads need a PhD before entering industry in comp bio, so that should also be advertised to students.
- As I previously stated, I think more computational biology specific courses that incorporate practical applications would be beneficial to bridge the gap between the core courses we are required to take.
- Writing a paper has been eye opening, but I would've preferred something more related to what you would find yourself doing in a comp bio job.
- More outside resources available to students within the major, but also a refined curriculum and major progression.
- More research opportunities, more practical classes, more comp bio focused python classes, batch programming classes, etc. The department is basically just Maldonado and Durrant, which is insane because this field is quite literally exploding right now. Get it together and follow the trends. Hire more people, use the school of medicine to help you set up the classes and prerequisites.
- More resources on pursuing graduate schools for computational biology and related fields.

Based on your experience in this course and understanding that I am applying for a teaching faculty position in computational biology, please provide feedback about my effectiveness as an instructor and my potential contribution to the department's computational biology teaching mission. Your feedback will be considered as part of my application evaluation.

- I like the way you used PowerPoint slides to give a brief overview of the topic, but didn't base the entire class around it, giving time for discussion, both individual and group activities, and questions, which I think enhanced my learning as a student.
- I believe you have done a great job teaching this course, as I felt safe to ask questions and I was stimulated intellectually every time I came into class. At times waiting for your grades would be frustrating, and sometimes feedback was confusing. But I would just ask to you to clarify and you have been fair with deadlines. I do believe that you would be great for the comp bio teaching mission as you have already given so much diverse experience to students in this class and the actual comp bio class that you teach. You are very knowledgeable in the field, and you are very

driven and passionate about the subject. You wold be great as teaching faculty, the bio department could really use someone like you.

- Dr. Maldonado has been the most helpful and supportive professor I've had at Pitt. Most of the courses I've taken feel disconnected from anything outside of a classroom environment and haven't actually prepared me for how I can use the information I've gained. This class was the exact opposite: everything we did made the concept of comp bio more concrete to me. I think that Alex would be a great addition to the computational biology department at Pitt due to his passion for teaching and his ability to conceptualize complicated topics for undergrad students.
- Alex is by far the best instructor I have had in my time here at Pitt. He is extremely knowledgeable about his field and has given me a clearer idea of what I've been studying than anyone else in the computational biology department. He was always open and responsive to questions, great at explaining, and it felt like he cared for us and our needs as a class. This man is too good for you. Please just give him the job.
- Expectations for assignments are always very clear, and Alex has been willing to meet students where we are if there is a mismatch in what he expects and what we can deliver at the moment. Alex's feedback is always very thoughtful and thorough, and he engages with students often to discuss what is/isn't working about how he is teaching the class. He has been very flexible and has prioritized student feedback in order to meet both his teaching goals and provide what we are looking for from the class. Alex has cultivated a class environment which is comfortable and encourages discussion and collaboration. This allows students to freely express if we have concerns about any assignment or if we wish to take the class is a specific direction (for example, we are allowed to discuss deadlines and decide what kinds of papers we would like to learn more about). Alex has also made efforts to bridge the gap between the minimal coursework on computational biology methods most of us had prior to this course (earlier sections of 1540 did not prepare us well for the CADD topics we've been discussing in this course, and many of us have not yet taken our higher–level computational biology course). This has been largely by giving pre–lectures on a paper's topic so that we have a foundational understanding before reading the article and encouraging pre–lecture assignments to skim the paper.
- I sincerely appreciate the effort and dedication you put into this course. Your feedback was thorough, and I gained valuable insights into academic writing. I found your teaching approach effective, particularly your method of selecting key papers, breaking them into components, and encouraging us to work in groups. This structure deepened our understanding of each stage of the research process and helped us connect these elements to the broader context and incorporate them into our writing assignments.

  However, I wanted to share a constructive observation regarding professionalism. As computational biology is a small cohort program, I noticed that your interactions with a few students and UTAs occasionally appeared more relaxed or familiar compared to others. While your jokes and efforts to build rapport were appreciated and reflected your genuine care for the cohort, ensuring that all students feel equally engaged and supported is important for fostering a fully inclusive and professional learning environment.

Additionally, I observed instances where specific faculty members in the biological sciences department were mentioned in class. While I understand these comments were made with genuine concern for the program and its students, such mentions could raise questions about professional boundaries. As an educator, facilitator, and future career advisor, I believe maintaining a focus on education and fostering a neutral, supportive atmosphere is essential. This professionalism complements the emphasis on research, teaching efficacy, and competence, ensuring students are supported in both their academic and professional growth.

I hope this feedback is helpful as you continue to refine your teaching practices and pursue a teaching faculty position in computational biology. Your passion for the subject and dedication to student success are evident, and I believe your contributions could greatly benefit the department.

- It is clear that Dr. Maldonado is very passionate and knowledgeable about the field of computational biology, and I think this makes him a great fit for the department. However, I do believe his standards for his students' understanding and capability of writing scientific papers are simply too high. The classes before his do not prepare us for the graduate level of writing he seems to expect. I encourage him to consider this in his future teaching goals.
- I found you to be a very effective instructor in this course. I believe that the computational biology program here should be expanded with more staff and elective courses, more people knowledgeable in the field like you should be present in the department.
- I think you did a great job teaching this class. It is obvious that you know what you are talking about, and your experience within computational biology was evident with your explanations of tough concepts. I feel like you really pushed our class to learn about the current tools in the field, like machine learning and molecular modeling, and you were readily available to answer the tough questions that came with these complicated concepts. I think we need more classes that take a contemporary approach to computational biology, with less repeated lectures and more investigation into the current state of the field.
- Alex was excellent at providing useful feedback on assignments, especially focusing on areas that have been completely overlooked by the rest of the computational biology program, like honing scientific writing and presenting skills (essential real–world skills). However, he could be disorganized and the class could benefit from more structure.
- I feel like you are a good instructor that doesn't judge their students. I felt like no question was a dumb question and I felt safe to talk to you about any concerns.
- Alex would be a good contribution to the faculty for computational biology because he has the practical expertise that other professors lack. This was his second time teaching BIOSC 1630 so there is room for improvement as an instructor in terms of returning assignments in a timely manner and assessing what students have been taught. Overall though, I think he would be beneficial as a faculty member.

- Dr. Maldonado really does want to help his students and he is very approachable. He is knowledgable about the computational biology field and I only felt like I was learning about this major after speaking to him and taking his class. In class discussions are comprised of relevant topics to current research and industry and I have found myself talking about the topics Dr. Maldonado discusses outside of class or even seeing them in other classes. I think he is an excellent addition to this department and would assist many computational biology majors in the future with their careers and future goals if he is offered a teaching faculty position.
- I think that Alex is very knowledgeable about the field of comp bio in a way that none of my other professors at Pitt are. Despite the challenges of teaching novel concepts within a relatively new major, he is very adaptable and understanding. He takes student feedback, but also understands where he must stand firm in order to maintain the content of the course. Even though the workload of the course was not extreme, I felt that every activity and every assignment I turned in taught me something.
- I believe that you are a knowledgeable, dedicated, and engaging professor. I gained so much knowledge from you and this course that I would not have had without all your guidance. I not only learned key skills such as reading, writing, and presenting scientific research, but I also learned so much about the field as a whole. I decided that I wanted to pursue a PhD in Computational Biology after you shared your own experiences and insight in the field with the class. You also created environment where students felt safe to ask questions not only about the course, but also about the computational biology and research field and our own career ambitions. You also made sure to listen to your students and heed feedback that not many professors I have encountered do. You also simplified and explain complex topics, which greatly enriched my understanding. Having a professor like you gave me immeasurable understanding and knowledge in this field that I was new to.
- You were not easy to reach over email.
- I think you did an extremely good job of being available to students and answering questions. You also did so in a way that made you extremely approachable, which made it much easier for me to learn. One of the standout qualities was the thoughtful and detailed feedback you provided for all our assignments. Your feedback not only helped me improve my work, but also helped direct my research, which deepened by understanding of the material. Whether through examples, walkthroughs, or addressing questions patiently, you ensured that everyone in the class could follow and understand the material. You also (in my opinion accurately) anticipated areas where students might struggle and proactively clarified those points, making the learning process much smoother and more engaging. The way you were able to break down complicated topics was truly impressive. Also, you went above and beyond by discussing career paths, sharing insights about opportunities in the field, and providing guidance tailored to our individual goals. I truly felt that you were invested in our success, both inside and outside of the classroom. Your ability to communicate complex concepts clearly, your dedication to student success, and your enthusiasm for the field feels like the perfect contribution to the department's teaching mission, and I strongly

believe you would make a vital contribution to the academic and professional development of students in the program.

- Dr. Maldonado is fair, understanding, and knowledgable, and is fighting a hard fight to teach undergrad students useful techniques, especially since the prerequisites for the comp bio classes DO NOT AT ALL set you up for success in the later classes. He knows what needs to be done to make this program insanely competitive and important at Pitt.
- I found that professor Maldonado was unique in his specific feedback−−he gave deliberate and intentional feedback on class presentations and on our assignments, and for me his feedback was more in−depth than most professors I have had here at Pitt. This feedback was some of the most rigorous and specific feedback I have received at Pitt, and I feel that this greatly improved my skills, especially in effectively analyzing papers and in academic writing. He also was very deliberate about structuring this course, and I appreciated how he focused on developing our skills in effectively understanding and evaluating information in scientific articles, presenting that information, and also performing academic writing that synthesizes based on information found in papers. I haven't really had enough guidance on this front here at Pitt, and I feel that my skills in these areas have significantly improved as a result of this course. From these things, I get the impression that professor Maldonado cares greatly about improving the skills and abilities of his students, and has a good focus for what is important to develop in the field of computational biology. Furthermore, he spoke at length many times on the current trends in computational biology research, and given our discussions I feel that I have a more complete handle on the direction that the field is taking.

What aspects of the computational biology major have been most challenging, and how could they be improved?

- As stated above, I feel that some more time could be given to finding the specific perspective that will inform the term paper.
- I've enjoyed every aspect of my major, as it allowed flexibility for medical school. But if someone were not going into medical school and wanted to enter the industry from the major, they would not be competitive for jobs. This major requires you to go to grad school if healthcare or some other path is not your path.
- Since it is an underdeveloped field, it is tough to find opportunities in specfic areas and niches. I am still discovering some of the more practical research applications of my work, but given the many different amounts of subfields, I never had an opporunity to explore. It would be nice to have a more central "comp bio" department that is not split between DSAS and SCI, since half of the opportunities and things available to me are in a different school. I would be able to focus on my major more early on with a clearer vision.
- Coursework integrating biology and computer science foundations did not come until the very end of my time here. If I didn't join a lab, I likely would not have known which direction I wanted to take my computational biology work after graduating. I think this was primarily an issue with the way 1540 was taught, as there was minimal overlap between what we were learning in class and

what I was doing in my lab (computational structural biology projects). While I don't expect a full understanding of how the methods I use in lab work, I do wish I had been introduced to them before utilizing them in lab and gotten at least a general understanding of what they do and how to apply/ interpret their results.

- I believe the computational biology major is adequately challenging. I wish the option of computational biology students completing the computer science minor was decided and communicated to students much earlier on. Due to the change being made during my junior year, I was not able to fit in the needed coursework during my remaining time at Pitt.
- The aspect most challenging was some of the general classes such as chemistry.
- the computer science side is much more challenging than the biology side, and I feel like there are more resources for biology help. Alex was always really kind about explaining concepts I didn't understand (when maybe I should have) and I think we need more professors who have worked in computational biology to bridge the gap in understanding
- The most challenging part has definitely been a lack of available resources and relevant course content in the program, as laid out previously.
- The presentations were hard for me, I felt like I didn't know enough.
- The classes do not seem thought out, and the lesson plans are extremely disorganized. This is what I've noticed from the three computational biology courses I have taken so far.
- The computer science aspects have been challenging as well as the individual concepts in the classes. I think they could be improved by keeping the current computational biology course as is and adding more comp bio courses that are based off of that course.
- I think many people entering the comp bio major at Pitt have no previous programming experience, and that can make it difficult, because coding is so different from any other skill we learn in college. That was the most difficult part for me. I think a better up–front understanding about how much programming this major requires, as well as resources for students to improve their programming skills, will be very helpful. Especially problem–solving and debugging help, so that students know how to solve problems they do not know the answers to.
- I think the most challenging part of the major is taking difficult courses in biology and computer science but having too few courses that are able to bridge the two subjects and give us practical applications to our future careers.
- A lot of the jargon can get complex, especially when reading a research article, but I feel that you did a very good job of breaking that own.
- The lack of resources, advice, and opportunities

What are your career goals after graduation?

- Medical School and maybe work in biotech afterward.
- I'm very undecided between med school or some other health career path.
- My main goal right now is to do a post–bac or fellowship program, and enroll in a PhD program further into the future. After that, I hope to continue research and potentially teach.
- Graduate programs in computational biology/bioinformatics/biotech, possibly into industry

- Graduate school
- Attending medical school to become a physician.
- I want to attend grad school
- I am going to medical school
- After graduation, I plan to start a job in industry and potentially get a masters degree.
- Masters
- Pursue a PhD in computational biology or bioinformatics and work in industry or an NGO.
- My career goals are to go into medicine. However, I hope to do computational neuroscience research hopefully or research related to my major in medical school.
- I plan to apply to medical school and become a physician, eventually splitting my time between clinical practice and research.
- I would like to pursue a PhD in computational biology and eventually work in cancer drug research and development.
- Grad school
- Comp Bio research in the school of medicine.
- Graduate school and research in academia

What are the essential qualities of a successful instructor for computational biology courses?

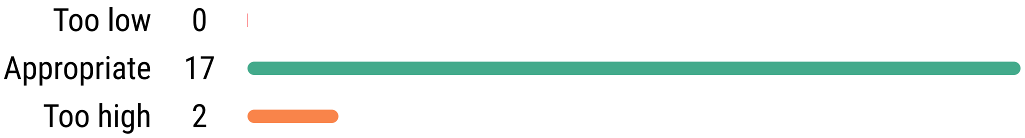
- Knowledgable, Kind, Patient
- Someone who has experience in computational research, can break down topics for student understanding, and is adaptable with the evolution of the field.
- Being open and responsive to questions, great at explaining, and it caring for the needs of a class.
- Strong understanding of the field, flexible, and approachable
- Being interested, up–to–date, and knowledgable in the field of computational biology.
- Being supportive, knowledgeable
- Understand and can explain computational biology. Ideally have research / career experience working in computational biology or applying computation to their biological research. Have a good grasp on where the field is now, and where they think it is heading
- Computational biology is a field that encompasses lots of things, and I think it is important for an instructor to be knowledgable in many of these. Many instructors are only knowledgable in the field of their specific research, which leaves students with an incomplete picture of the field and what opportunities are available to then after they graduate.
- Approachability
- Someone who has both experimental and computational research expertise, and is able to effectively integrate them and show students how both types of research come together through computational biology.
- An instructor must know current industry and what topics are relevant right now. They should help inform their students what skill sets they need to be successful for a career in computational biology instead of simply teaching a class. They also need to be approachable and willing to talk with their students and make recommendations based on the students' career path.

- Adaptable, understanding, knowledgeable about the field, willing to continuously make changes to curriculum because the field is constantly evolving. Cares about students' well–being and flexible with deadlines because the course material is difficult and flexible deadlines can allow for a better final product.
- There are numerous essential qualities of a successful instructor for computational biology courses. It is important to be able to simplify and explain complex topics in this field and give students real–world applications. Additionally, it is necessary to listen to students and create an environment where students feel comfortable seeking help. An instructor who also provides guidance in the field as a whole, with tools that we can apply in the future.
- Patience, persistence, compassion, enthusiasm
- Understanding of students knowledge, understanding of the current field, and understanding of how to merge these two to effectively teach students in a efficient way.
- In a fast–moving field like computational biology, I feel that professors who prioritize a forward–facing approach to teaching topics are essential.

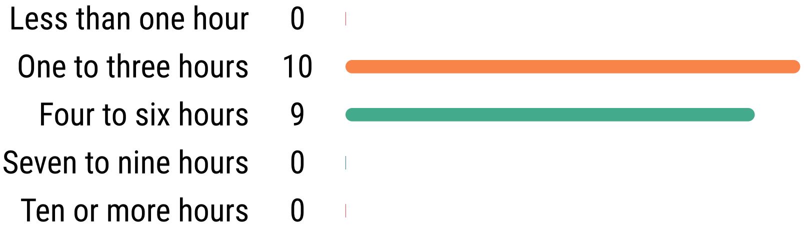
Fall 2023

I received evaluations from 19 out of 21 enrolled students, representing an 90.5% response rate.

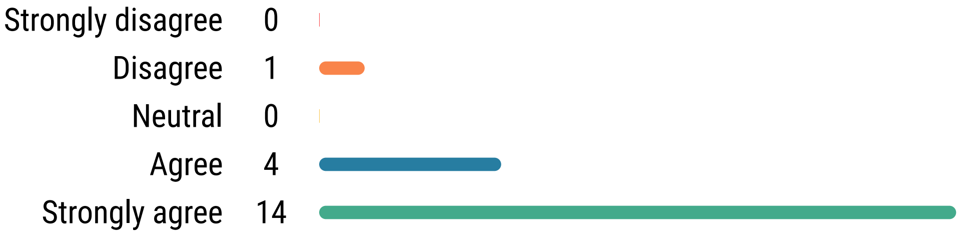
The standards the instructor set for me were



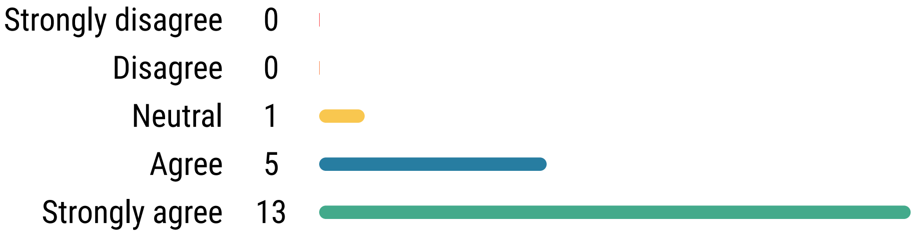
How many hours per week did you usually spend working on this course outside of classroom time?



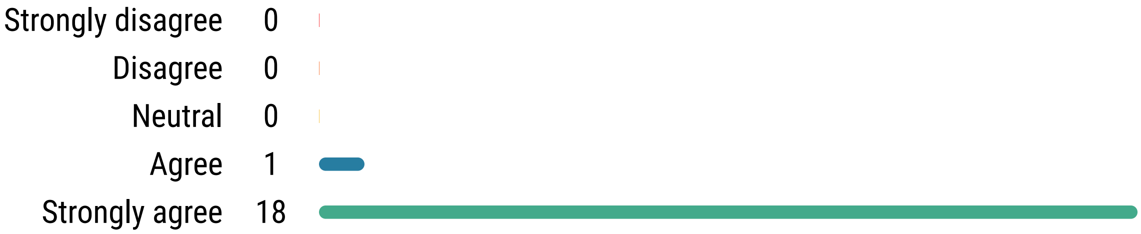
The instructor created an atmosphere that kept me engaged in course content



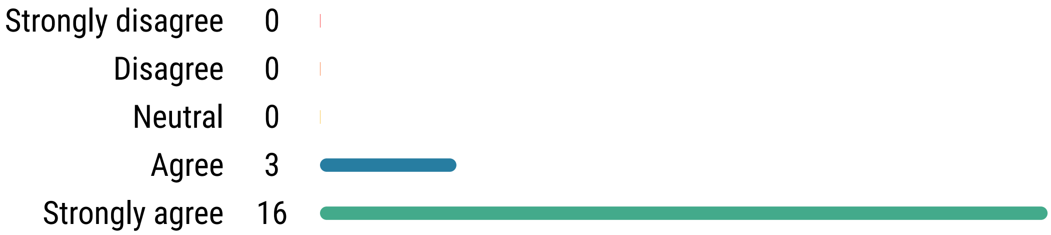
The instructor was prepared for class



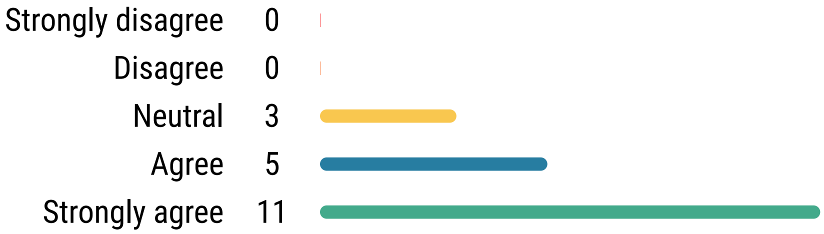
The instructor treated students with respect



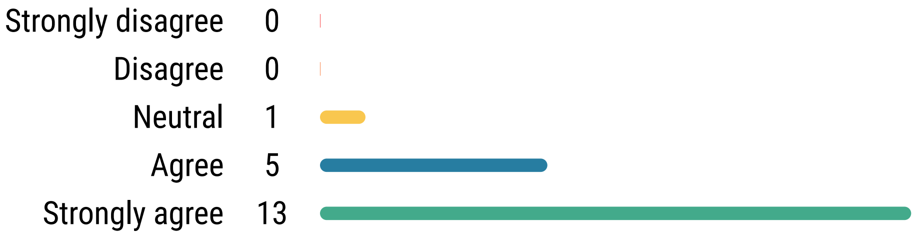
The instructor was available to me (in-person, electronically, or both)



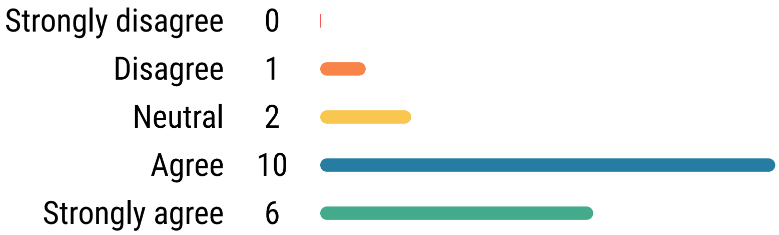
The instructor evaluated my work fairly



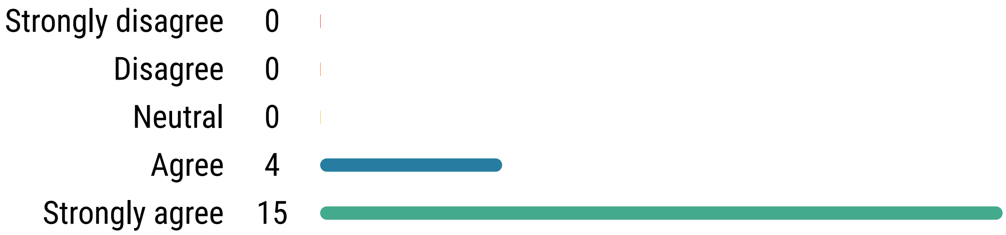
The instructor provided feedback that was helpful to me



I learned a lot from this course



The instructor creates an inclusive learning environment for all students



Classroom activities made a valuable contribution to my learning

Strongly disagree	0
Disagree	0
Neutral	3
Agree	11
Strongly agree	5

Class discussion contributed positively to learning in this course

Strongly disagree	0
Disagree	0
Neutral	1
Agree	8
Strongly agree	10

Assignments contributed to my learning experience in this course

Strongly disagree	0
Disagree	0
Neutral	3
Agree	8
Strongly agree	8

Grading policies were fair

Strongly disagree	0
Disagree	1
Neutral	2
Agree	4
Strongly agree	12

Useful feedback was provided

Strongly disagree	0
Disagree	0
Neutral	1
Agree	7
Strongly agree	11

Instructor conveyed knowledge of the subject matter

Strongly disagree	0
Disagree	0
Neutral	1
Agree	4
Strongly agree	14

The online course materials were easy to use

Strongly disagree	0
Disagree	1
Neutral	0
Agree	6
Strongly agree	12

The instructor required an appropriate amount of work for the credits received

Strongly disagree	0
Disagree	1
Neutral	1
Agree	12
Strongly agree	5

The instructor was well prepared for each day's class

Strongly disagree	0
Disagree	0
Neutral	0
Agree	8
Strongly agree	11

The presentations provided a meaningful learning experience

Strongly disagree	0
Disagree	0
Neutral	2
Agree	6
Strongly agree	11

This class made a valuable contribution to my professional development

Strongly disagree	0
Disagree	2
Neutral	1
Agree	4
Strongly agree	12

Group work contributed to my understanding of the course material

Strongly disagree	0
Disagree	0
Neutral	4
Agree	11
Strongly agree	4

This course helped me communicate clearly about this subject

Strongly disagree	0
Disagree	0
Neutral	1
Agree	10
Strongly agree	8

The instructor actively engaged students in the learning process

Strongly disagree	0
Disagree	0
Neutral	1
Agree	5
Strongly agree	13

What did you like best about how the course was taught?

- Alex adapted to the feedback and took action fairly early and quickly.
- I appreciated and valued how prepared the professor was for each and every class. Having everything set and laid out on a Github website made it clear and easy for me to find where and what we were doing beforehand. I thought especially in a writing class, it was extremely helpful to have all sorts of resources in an organized fashion.
- I liked how for the final paper we had multiple drafts to gain feedback
- I liked how the entire course was structured, from start to finish learning how to effectively read scientific papers. Then on to researching and picking our topics. To then write our final paper. While

it was an over two–hour class having a break every hour made it much easier to concentrate and learn while in the classroom as well.

- I loved the open and honestness of the classroom teaching style. There was a lot of information provided for every assignment, and we had an ample amount of help and resources.
- I liked that a lot of the activities were more conversational and that the entire class was in a conversational format.
- I liked how non rigid it was. There was certainly a lot of preset assignments however I really liked being able to give feedback that altered the content covered in the class.
- I liked the structure of this course a lot. In the beginning, we learned about how to read and critique research articles, which set us up for our presentations and final papers later on.

the paper was broken up by drafts and not just one big grade at the end

- I liked the fact that the course and instructor were adaptable and listened to what the students needed
- I liked the presentations and paper that we had to write, I feel that it has developed my skills in those areas
- I enjoyed the interactive coursework in class. The programs helped me gain more insight into the course itself.
- I liked the way the presentations were done. I felt liked I was able to comprehend and actually understand what was being thought in the class.
- I liked the smaller class size because it gave me opportunities to ask questions and really be involved in the discussions
- I like this class because it can urge me to read the research paper. And each part can be analyzed, which is very helpful in the future work or continue to study.
- I liked how the assignments were clearly outlined at the start of the course, and how the class itself is more–so spent doing group activities and supplemental.
- Alex was very open to suggestions and ran the class based on feedback given during the class.
- Generally, I liked how the course was laid out with the giant paper over the semester along with the presentation.

If you were teaching this course, what would you do differently?

- I would probably not include such a large volume of papers to read, but instead focus on one paper. This way the analysis process becomes much easier and it doesn’t feel like time was wasted reading multiple papers that were not particularly useful. Techniques can always be demonstrated with one paper.
- We did end up doing a variation of this, but I wish we could have a set in stone lecture on graduate school/professional career advice earlier in the semester. Most of the class was applying to graduate schools or jobs and it was honestly super helpful being able to hear from someone in the field who has gone through what we had.
- I would dedicate a whole class to working on paper/presentation so that we can ask questions. It was very hard for me to figure out how to get started

- I cannot think of anything I would change, Alex formatted the class amazingly well.
- I have no input! I though it was taught very well, the class was definitely long and hard to focus in, but the instructor did as much as possible to help.
- I did not enjoy all of the group activities at the start of the semester. I feel as though a broader discussion format or more tophat/canvas discussion posts would be more engaging for everyone.
- Maybe more smaller group discussions about papers.
- I think I would spend a little time teaching students where to find credible research papers and also talk about the many different fields that these research papers can cover. I know that this class is computational biology, but it would be interesting to talk about papers pertaining to computational neuroscience, chemistry, immunology, etc., since they have some degree of overlap.
- include some mini hw presentations to practice and get used to presenting.
- I do not think there is anything I would do differently for this class
- Maybe break it up into two different days
- If I were teaching this class, I would break the same papers into sections and move on to a new paper after one, rather than using a new study for each section of the research. Also, the grading for this lesson is a little different than my other writing lessons, where it is more based on right or wrong opinion. I may change the grading a little bit.
- I may focus more on how to present and how to write. I know that is difficult to do given its very subjective, but some more guidance on that front, given that it is the majority of the grade comes from those two things, would be helpful.
- The class is too long. If the goal is to teach us how to read and write about scientific literature, it could be done better with a shorter class period where we read something at home and talk about it in class. I have a hard time reading during class because there are too many distractions, so this approach would have been better for me.
- Alex's grading standards were tougher than those in other writing–intensive courses. Therefore, preparing for my presentation and writing my paper was more stressful than expected. In addition, I believe this course should primarily focus on evaluating students' understanding of their papers and observing their progress throughout the term.
- If you asked me what I learned from this class I could barely give you a list. I didn't understand the purpose of the class I wish that this was an elective rather than required because I think that students who want to go into academia should take this course, but others like myself who would want more of an industry focus where you could be working on actual projects or something that could give them experience in the field would've been better.

Are there any specific areas or topics in which you feel I excelled as an instructor or any areas where I could have been more effective?

- You gave great feedback whether it was for assignments in class or for our questions regarding post–graduation plans (grad school or work). Especially being so close in age to us I think is something you should use to your advantage and can make your future semesters great. It just makes it easier to talk to you as a person and professor since you were in our shoes not too long ago.

- I think the general inclusivity and mindfulness of all students was excellent. I feel like for such a small class, we may have benefited from required individual meetings at some point. I have had professors cancel a class and replace the class time with some one on one meetings around the midpoint in the semester.
- I believe you excelled at helping us understand the research papers better and how to effectively read scientific literature. I believe the only area that could have been more effective is explaining the purpose and how exactly to write the final paper, as I wasn't entirely sure what I was supposed to write until the second draft (I was also writing similarly to the unofficial microbiology papers).
- you were very good at engaging with the class. it made speaking in front of everyone easier.
- I think you did a very good job of giving feedback and being available for students
- The beginning of the class work.
- I really enjoyed how honest you were as an instructor and the relaxed environment you were able to provide. Going to class can be stressful but I did not feel that with this class because I felt comfortable participating.
- Communication!
- I think the small in class activities we would do were effective.

Is there anything else you'd like to share regarding my teaching style or my role as your instructor in this course?

- Keep your style of teaching. You were very approachable and kind which made it much easier to engage with you and participate in class and with my classmates.
- I really enjoyed how you incorporated your other interests into the class content. The graphic design stuff was cool to see. I though you were very approachable and a great knowledge base for students trying to figure out their post–undergrad plans.
- I liked how comfortable the environment was. It felt more interpersonal which facilitated more learning.
- You looked out for the students' best interests by extending deadlines and providing thoughtful feedback, this was a very nice thing to do for us.
- I really enjoyed your class
- I really liked your teaching style and the way you ran the class
- no I liked the teaching style
- The website was great and very helpful
- Thank you
- You were very amiable which made the class environment a lot more comfortable! Thanks for a great semester Alex!