JRaviLab docs

For members of the JRaviLab: past, present, and future!

Janani Ravi & members of JRaviLab

2025-12-18

Table of contents

About JRaviLab docs

This is a short e-book compiled to guide, onboard, and offboard JRaviLab members at the University of Colorado Anschutz Medical Campus (previously at Michigan State University).

Acknowledgments

We appreciate and build upon awesome onboarding resources from Arjun Krishnan, Casey Greene, Fan Zhang, Annika Barber, Christoph Rau, Jeff Leek, eLife, and several others.

If you'd like to edit/update/contribute to this e-book (*built with Quarto*), please email me or submit a pull request to the GitHub repository directly.

How to reach us

Webpage | PI | Email | Mastodon | Bluesky | Twitter (inactive)

1 Onboarding

Welcome to JRaviLab! We are excited that you are here — as a student, postdoc, or visiting researcher! We have a number of resources available in this short e-book and on Slack. Please take the time to check these out. As you read them, if something is unclear, please feel free to reach out to the group anytime.

Once you are done with the 'familiarizing' and getting to know the first steps, please send me (JR) a message on Slack. Please bring your questions, thoughts, and feedback to the meeting.

Good luck & welcome aboard!

2 Mission, Expectations, Conduct, Support

- Mission, expectations, code of conduct
- Health, wellness, safety
- Diversity, accessibility

We believe in investing in and nurturing scientists as much as the science. We pride ourselves on providing a safe and inclusive space for people across intersectionalities. We are committed to mentoring (postdocs, students, programmers), education and outreach, and personalized professional development. We are passionate about increasing diversity and retention in STEM and quantitative sciences. If you share our passion, please reach out to us!

3 Role-Specific Expectations

Details on expectations per-role can be found here in the Roles and expectations section, but if you have any questions, feel free to talk to any of us!

4 Lab Proceedings

- Working remotely | CU Remote Work Agreement
- Communication within the group | Slack | In-person/Zoom co-work
- Research update presentations | Meetings
- Giving & receiving constructive feedback

5 Slack

Since our lab runs on Slack, let's first get you on there. All our conversations & every last bit of science get done here! :) So, please install this on your desktops (phones) and turn on notifications during working hours, whenever that is!

5.1 Getting started

When you join, please introduce yourself in `#general`. Tell us a little about yourself here — where you are from, what your interests are, science-wise or otherwise, why you are interested in working with us, and in what capacity you'll be joining us.

I've never used Slack – where do I even start? Maybe you can start with the Slack cheatsheet and Keyboard shortcuts.

5.2 What next?

Next, you can join the different channels to participate in various kinds of conversations with the group — look at the channel description, check out the pinned messages of the channel, and dive right in! Here are a few examples.

Finally, you will be invited to specific project channels (based on the primary and secondary projects you will be working on). Based on chats with your colleagues, if you'd like to contribute to/give feedback to other projects, feel free to join those channels too.

5.3 Meet the group

You may have briefly spoken to a few group members prior to joining. I urge you to take the first few weeks to introduce yourself to the group members and chat with them (via Slack or over coffee/tea) to find out what they are up to and learn a bit about the lab.

6 GitHub

If you haven't already, please create a professional GitHub account (e.g., `jananiravi`) with your full name. Once you pass that along to us, we will add you to the JRaviLab GitHub organization. Also, a neat memorable username will give you the opportunity to host your own webpage (e.g., `jravilab.github.io` or `jananiravi.github.io`). Our group repo is here.

7 Social media

If you have officially joined us as a grad or postdoc, or undergrad for 3/6+ months, you will automatically get added to our group webpage. If not, please check a few sample pages and furnish me with those details – I can add you there right away. Also, we use Twitter for professional networking & announcements (connecting with the broader scientific/R/technical communities). If you have a Twitter handle, share it with me (DM), or follow #auto-twitter for updates. You can check out/follow a few accounts maintained by us: JRaviLab | my handle | RLadies-Aurora | RLadies-EastLansing | Women+ Data Science | AsiaR | ISMB EvolComp-Gen COSI. Thanks to recent *interesting* changes at Twitter, people have been migrating to Mastodon (e.g., genomic.social, fosstodon, JRaviLab) and Bluesky | my handle!

If you would like to co-maintain any of these accounts or our webpage, please DM me. I'm always looking for volunteers! :)

8 Lab culture

8.1 Mission

We love all things related to computational biology, biomedical data science, and microbial genotypes, phenotypes, and diseases, but we value and believe in investing in and nurturing scientists as much as the science. We pride ourselves on providing a safe and inclusive space for people across intersectionalities. We are committed to mentoring (postdocs, students, programmers), education and outreach, and personalized professional development. We are passionate about finding new, better ways to increase diversity and retention in STEM and quantitative sciences. If you share our passion, please reach out to us!

8.1.1 Mentoring Philosophy | full_post

My experiences as a mentee have varied widely: fantastic professors in college inspired me to start a research career; later, a mentor overlooked my whole dissertation work resulting in prolonged authorship discussion. In my postdoctoral lab, I was the sole computational researcher collaborating with a dozen experimental biologists. I seized this invaluable opportunity to restart my academic career from scratch in a different field (i.e., studying microbial diseases), to learn, understand, and speak the language of both experimental and computational biologists, and to bring these worlds together. Collectively, these experiences have shaped my mentoring philosophy.

- *Opening doors and leveling the playing field* (i.e., not gate-keeping)
- *Lives bigger than work*
- *Collaborative mentoring* (i.e., not top-down advising)
- *Mentoring is not imprinting*

8.1.2 Inclusive lab climate | full_post

I have mentored 30+ diverse undergraduate and graduate students. Several joined my lab as first-time coders and have co-authored computational papers; five have contributed to submitted manuscripts (see Bibliography). All these students have had positive research experiences, presented at (inter-)national conferences, and won awards/support from national programs, including NIH-funded T15 and R25-BRUSH, NSF-funded REU-ACRES, SROP, and NSURP.

Several undergraduate trainees have graduated and started advanced studies at prestigious graduate and medical schools or jobs at pharmaceutical/IT companies. Using lessons learned from these experiences, I will focus on the following key areas to ensure an equitable and inclusive lab climate.

- *Recruiting practices* - e.g., working with the Offices of Inclusion/Diversity, R/Py-Ladies, W+DS, and online groups for minorities in STEM (e.g., Black in Computational Biology, LatinR, AfricaR, AsiaR, ArabR, RainbowR).
- *Safe space for full participation* - for people from all intersectionalities, zero tolerance to discriminatory or exclusionary behavior, appropriate pronoun usage, sustain a culture rooted in mutual respect.
- *Empowering mentees to forge their path* - e.g., co-developing semester plans, conducting professional development workshops within the lab, creating networking opportunities through local/national/international meetings, and tailoring mentoring strategies to each mentee's career aspirations.

8.2 Roles and expectations

The following sections were drafted by folks at each of these levels and edited collectively by the entire group for cohesion.

8.2.1 PI

My role as a PI would be to ensure your growth and success as a trainee, and the success of the project. Towards this, I will

- **Plan and design**
  - **Sounding board**. You will have the freedom to work on your best ideas broadly aligned with the lab's vision, and you will always find in me an audience to discuss and brainstorm.
  - I will help outline and plan your project(s) and design appropriate analyses.
  - I will check in with you once a semester as you plan your time, and more regularly to ensure you're on track – based on the professional, scientific, and personal goals you've set for yourself. In addition to the project/group meetings, check in with me often (bi)weekly (or as needed) to make sure you are staying on track towards your primary research and professional goals and to help me help you as best as I can.

- **Honing your skills −> career development.** I will provide resources to develop your research program and technical skills as you plan the next stage of your career (e.g., PRA −> grad school, PhD student −> postdoc/industry, postdoc −> PI/research lead).

- **Scientific communication.** I will provide ample opportunities within and outside the group to develop your reading, writing (grants and papers), and presenting skills, including engaging and networking at (inter)national meetings.
- **Professional networking.** I will connect you with potential collaborators and mentors for scientific and professional growth and reciprocity. Team science is one of several skills you will learn during your tenure with us – you will learn to communicate effectively not just with other computational colleagues but with the admin, grants and scientific writing teams, and experimental and clinical collaborators.
- **Safe space.** I will continually work towards creating a diverse and inclusive safe space for all intersectionalities in the lab to ensure everyone thrives and grows scientifically, professionally, and personally while respecting and supporting their colleagues.
- **Education and outreach.** I will enable and strongly encourage you to pay forward by creating inclusive communities through scientific and technical education and outreach reach opportunities within and beyond CU (e.g., R-Ladies, AsiaR, WiSTEM, SACNAS).

8.2.2 Graduate students

- **Learning to be a well-rounded scientist**
  - Strive to be creative, independent, and collaborative (as a team scientist).
  - Be willing to learn new things and be open-minded to new ideas/directions. Be thoughtful and think deeply about your rationale and hypothesis. Be critical and rigorous in your methods and evaluation. It is perfectly OK to find your passion and love (and dream about) certain topics/fields, but remain open enough to like/appreciate other topics (and avoid maintaining a curtailing *hate/dislike* list). This will ensure you read, listen to, and consume ideas constructively, ultimately honing your scientific growth.

- **Reading**
  - Have a clear and deep understanding of concepts (computational and biological) related to your research project.
  - Keep up with the literature and have a sound reading system (update your project/group members and PI of your most recent exciting reads via Slack/Journal clubs). Post updates regularly to the `#papers-articles` channel based on your self-assigned journals here.

- **Coding**
  - Write clean and well-documented code, and commit to GitHub regularly.
  - Do code review or request review(s) from other lab members at a proper frequency e.g., once a month.
  - Your work must be reproducible; this applies to the methodology, codebase, software, and data you work on. (e.g., Ten Simple Rules for Reproducible Computational Research)

- **Writing**

  – Keep applying for internal/external fellowships and funding opportunities (it would be excellent practice for scientific writing and critical thinking; it recognizes your ideas and trajectory; it would also be amazing to receive funding to pursue new independent research projects that even slightly divergent from the PI's projects).
  – Contribute proactively to grant proposal writing with the PI by proposing new ideas and generating preliminary results.

- **Scientific communication**

  – Formal/semi-formal scientific writing, e.g., abstracts, manuscripts, grant/fellowship applications, blog posts.
  – Give oral presentations in different settings, e.g., group meetings, posters, talks, research updates, and journal club meetings.
  – Keep well-organized meeting notes. In group meetings, take turns as the scribe.
  – Look out for opportunities that PI/others often post in the lab Slack, the RIO/CU Bioinfo Slack workspaces, or email. These opportunities to look out for are but are not limited to, funding, poster events, presentations, conferences, seminars, and courses.
  – Submit abstracts to several conferences; ideally, all regional meetings that the whole group would attend (e.g., Rocky Mountain Branch of ASM, Rocky Bioinformatics, Frontrange Microbiome, qCMB at CSU), at least one national meeting (2+ years/post-comps grads are encouraged to submit talk abstracts and more junior (pre-comps/rotation) students can submit posters if they have in-progress or shorter stories to communicate. All students are encouraged to apply for travel and conference fellowships from CU, national societies, and the meetings themselves.

- **Meeting and engagement**

  – Attend all required meetings specific to your project and group meetings. If you would like to gain a general understanding of the other projects, you are welcome to attend other project meetings.
  – Schedule regular (weekly/biweekly, as needed) one-on-one meetings here. Maintain a good frequency of one-on-one meetings, especially when you are co-mentored. Do your best to keep both PIs on the same page.
  – Interact professionally (and respectfully) with all your peers within and outside the group.
  – Actively (and voluntarily) engage in lab, department, and program activities.
  – If a personal, scientific, or professional issue/concern is keeping you from engaging actively and fully with the group, let the PI and the group know so we can make reasonable accommodations and allow you to heal/recover and better integrate.
  – If co-mentored, it is expected that the trainee respectfully engages with both group members and attends events/meetings from both (unless exempted due to reasonable requests).

- **Communication**

  - Check and respond appropriately to Slack and Outlook email and calendar regularly during working hours. Set reminders, as needed, for what is relevant/expected of you so your PI/colleagues do not have to remind you.
  - Respect everyone's time by showing up to meetings regularly and on time (and in person if from CU and if that's how everyone else is attending), and by not expecting prompts for responses and engagement.
  - Communicate regularly and clearly with the PI(s) on research, academic progress, and other professional development goals. Constructive feedback both ways is important for a strong long-term relationship.
  - Have good/proper interpersonal communication with everyone, and create an approachable environment.
  - Come prepared for all the meetings by noting your open questions, reading relevant material, planning your next steps, or the results you want to present.

- **Peer-mentoring**

  - Mentor junior/newer trainees by giving primers, helping them troubleshoot anything related to their projects, helping them contact the correct individuals when you can't help them, and answering their questions.

- **Project/time management**

  - Set aside 3-4 hours of productive slots on days without classes for (i) writing code, (ii) reading/writing papers, and (iii) generating results.
  - Inculcate good time management skills. Ensure you have an efficient work-life balance to nurture your physical and mental health.
  - [Side projects] Learn more about other projects in the lab and make meaningful contributions to them, especially those most relevant to your research program (co-attendees of project meetings).
  - **Deadlines.** Establish clear deadlines/academic plans, e.g., comprehensive exam, defense. Plan to graduate well within 5 years, especially if you are ready in terms of progress, manuscripts, conferences, and other milestones; finish your comprehensive exam by the start of your third year (end-of-second-year would be preferable and ideal to keep you on track).
  - Adhere to general work ethics laid out for the lab.

- **Semester planning.** In addition to detailed research goals, set SMART goals for professional and personal development. Below is a general list of what semester planning entails.

  - Academic planning: required/elective courses.
  - Take on leadership roles through mentoring or presenting.
  - Attend seminars (regularly/weekly; look beyond your compulsory grad program series alone) and conferences.

- – Set clear reading, writing, coding, and presenting goals.
- – Volunteer | education, outreach, JEDI.
- – Set recurring meetings with the PI(s) and attend project meetings.

8.2.3 Postdocs

As a postdoctoral researcher, you wear many hats besides those of more junior scientists in the lab. So, in addition to the expectations from a graduate-level researcher (above), these might be relevant to your senior level and experience.

- **Training opportunity.** You are here to learn new skills and expand your scientific boundaries. You are not expected to know everything, but you should be comfortable (and preferably excited!) throwing yourself into new problems and solving them.
- **Peer mentoring.** You are also here to help others learn. Postdocs have considerable experience in many aspects, typically involving academia, research, and more. That experience can be beneficial for scientists who are earlier in their careers. Contribute generously where you can, and make yourself available for others in the group. A rising tide lifts all boats.
- **Scientific independence.** Postdocs are strongly encouraged to pursue their independent research interests aligned with the group's research interests. For projects that succeed, they should have "future directions" they can take forward into their independent careers in the future.
- **Collaboration.** In addition, you may work on several existing projects across the lab – either leading them or helping each team navigate research obstacles. Postdocs have already had experience troubleshooting their way through their work in the past, and are expected to be able to leverage that experience by operating more independently. Some projects you work on might be new and higher risk, and some might be stalled and in need of a fresh perspective. Postdocs are also strongly encouraged to engage in a few active collaborations outside the group (from our ongoing list) — to test their methods on interesting biomedical applications, gain diverse perspectives, while balancing basic and translational research. The professional networking aspect is critical, too.
- **Grants and fellowships.** Grant/fellowship writing is a crucial aspect of a postdoc's job. It funds your position and research where possible and, most critically, gives you practice and experience seeking and applying for funding opportunities needed for launching into independent faculty positions.
- **Communication.** Communicating science is a fundamental and essential part of this role, be it through manuscripts, conference presentations, or workshops. Expect to keep busy presenting your work at regional and (inter)national platforms, and use the experience to get your name out there and network! Once you've found your footing, we encourage postdocs to always apply for talks (alongside posters).
- **Career development.** Finding your next step is the ultimate goal of your position. For all the roles you have, they should help focus your skillset, CV, and open future

opportunities for you to move onwards and upwards towards where you want to end up in life.

8.2.4 Research professionals (PRAs / postbac, post-MS trainees)

- In addition to the most applicable points from the `Graduate Student` section above (barring rare exceptions like comprehensive exams, etc.), the following are great pointers for research professionals.
- **Individual and team scientist**

  - Lead individual projects and contribute new ideas and research directions, while also supporting other lab members' projects collaboratively.
  - Assist with the design and development of major bioinformatics-related programming projects.
  - The supportive role can overlap with the project-related scientific tasks or other technical and software-development tasks like code review and any assistance with coding/environment/installation issues.

- **Technical contribution**

  - Perform scientifically rigorous data management and bioinformatic analyses.
  - Develop and disseminate a variety of tools designed to access relevant experimental and clinical data.
  - Develop and implement complex analysis pipelines, modular functional programming, and data visualization techniques (e.g., for multi-layered -omics datasets)
  - Creatively and effectively integrate data from multiple sources to accelerate discoveries; write custom scripts to access databases and analyze data.
  - Write custom web tools and R/Py packages for the group and the larger scientific/technical community.

- **Communication**

  - (In addition to the opportunities/modes highlighted above) Present research/technical updates periodically, host journal clubs, workshops/tutorials, and attend all group and project meetings.

- **Professional development and networking**

  - It is in the best and mutual interest of PRAs and the lab to hone and develop scientific, technical, and personnel/leadership skills that will benefit all involved parties, especially the PRAs' professional and career development.
  - Work with the PI to make short- and long-term research, professional, and career plans to stay on track towards your next steps (e.g., grad school applications).
  - Work with the PI, others in the group and department, and collaborators to form a strong professional network that you can later rely on.

8.2.5 Undergraduates (Short-term interns/Visiting scientists)

The `Graduate Student` section above should give you a fair idea about how to be trained as a well-rounded scientist. As a junior researcher, however, the following would be excellent starting points.

- **Be present.** Attend meetings and bring forth your best effort.
- **Ask for help.** We encourage you to be proactive, ask questions, and provide any updates or ideas to our Slack channels.
- **Strive for growth.** It will be challenging, but we want you to keep trying and persist – learning is the main goal, and being receptive and open to constructive feedback is a great way to improve.
- **Give yourself grace.** You've earned a spot in our community, and you belong here. Failure is inevitable with the work we're doing; your work doesn't define you or your abilities. We're all here for you!
- **Stay organized.** Keeping track of your priorities, tasks, and responsibilities will allow you to monitor your progress and make it easier for others to help you. Time management is essential for pacing yourself to achieve your goals.
- **Collaborate.** We encourage you to present your work and discuss it with others frequently. Learn about the projects your labmates are working on, communicate any troubles or issues before they boil over (work-related and not), and keep a positive mental attitude!
- **Own your research.** This is *your* research process, so be sure that you're putting forth your best work and that you continue to practice scientific reading and writing.
- **Take advantage of resources.** There are many avenues for learning in research. Here are a few to get you started:
    - JRaviLab: How to X
    - Slack channels
    - Stack Overflow
    - NIH NCBI
    - Google, Google Scholar
    - ChatGPT

8.3 Code of conduct

8.3.1 Overview

There is zero tolerance for harassment, discrimination, micro- or passive aggression in the JRaviLab and at the University of Colorado. These are forms of abuse, and they create a workplace environment that can leave people feeling unsafe and unwelcome. Members of the JRaviLab come from many different backgrounds, and this diversity is one of our greatest

strengths. If you are uncomfortable with or have any issue with any lab member (including the PI), please bring it up immediately for a speedy resolution and a healthy and transparent long-term relationship.

While this is not exhaustive, examples of unacceptable behaviors include harassment, discrimination, or other unwanted behaviors, verbal or physical, that make others uncomfortable, both at and outside work. This applies to conduct in public or on social media, not just at the job. Protected categories include but are not limited to gender identity and sexuality, physical appearance, size, weight, and disability (of all types), race, ethnic group, age, and religion. These topics are covered in part by the required training modules you will take after employment at the University of Colorado.

You can review the policies from the Office of Equity's Discrimination and Harassment page here, and from the Anschutz Medical Campus' office of Diversity, Equity, Inclusion and Community Engagement here.

We recognize that because of our diverse backgrounds, our experiences with what conduct is considered appropriate may vary, and similarly, no one should feel afraid to be themselves out of fear of violating an unspoken rule. Instead, it is our collective responsibility to educate each other. If we see or hear conduct that is inappropriate, whether it is targeted at us or other individuals or groups, each of us must commit to calling that behavior out. A joke in poor taste alone won't get anyone terminated, but a pattern of misconduct despite being told it is inappropriate, arguing back or justification of misconduct when asked to avoid such behaviors, and severe misconduct are grounds for discipline.

If you are unsure what constitutes acceptable behavior and where the line is drawn, talk to us! These rules are important, but they are also a socially determined set of boundaries – they are not universally known and universally evident, and it is our collective responsibility to help each other understand what the boundaries are and how to steer safely clear of them. Similarly, if you feel like a comment or behavior might be inappropriate, you do not need to have a justification for feeling upset, you do not need to seek second opinions from others, and you do not need to observe the behavior repeatedly for it to be a problem – all of our individual boundaries, personalities, and histories are what define the rules, and your voice is important. You should always feel comfortable speaking up, especially if you are a lone voice!

If you observe or are the recipient of discrimination, sexual misconduct, or harassment, reports should be made to the Office of Equity. You can read about this process here. Depending on your classification, you may be a mandatory reporter of misconduct, meaning you must report any such misconduct you observe.

8.3.2 Professional Conduct

Details on responsible science, authorship, and the expectations of your role are found throughout this document. Generally speaking, as a member of the JRaviLab, you are a professional

researcher, and you can and should take pride in this. We are better for your membership! With this role comes certain responsibilities that go beyond the code of conduct overview mentioned above. Remember that your behaviors represent the lab, all of our research, and even Anschutz and the University of Colorado at large. Treat our fellow scientists in other labs and the general public with respect and understanding, try to avoid being drawn into public arguments on social media (nobody wins these), always be open to learning and re-learning, and do not let ego cloud judgment. We should never seek to use or diminish others merely to climb a career ladder.

As a group, our success and reputation, and the quality of our science and its positive impacts on society all depend on our lab having a sense of social cohesion and psychological safety. This goal is aspirational and does not materialize automatically, but with our collective efforts and daily upkeep, is very attainable.

8.4 Health and wellness

Health, wellness, safety

8.4.1 Inclusive and safe space

A part of this is also highlighted above in `Mission`, `Mentoring philosophy`, `Inclusive lab climate`, and `Code of Conduct`. In addition, the University of Colorado offers training in diversity, equity, and inclusion topics, including the Health Equity in Action Lab (HEAL). We encourage you to participate in these trainings, including the Foundations in Equity Certificate Program.

While not required by CU, these modules will introduce you to a great range of perspectives, explain topics like implicit bias or microaggressions, and how to recognize and interrupt them in ourselves and others and treat others more equitably and respectfully. Beyond the obvious benefits to our lab culture, these topics are absolutely critical to recognize for scientific study design (e.g. using human data). Being aware of our implicit biases and systemic issues can help navigate potential pitfalls in data collection and interpretation, which can affect people worldwide.

Links and details to follow.

8.5 Lab Proceedings

8.5.1 Communication

- Communication within the group | Slack | In-person/Zoom co-work

In all meetings and interactions, it is imperative that lab members are constructive in feedback and respectful of each other. Critiques are always welcome and encouraged, but they should be supportive and aimed at building others up and improving our science, not tearing others down. At the same time, we shouldn't shy away from providing critical feedback! As scientists, we all need regular practice critically analyzing and interpreting what we read and hear, and a friendly environment like our lab group is the best place to work on these skills together.

8.5.2 Meetings and formats

Since communication is key to good science, we meet regularly as a lab in a variety of formats. The details about these meetings can be found here.

8.5.3 Remote work & working hours

At a very broad level, the more time you put into your work, the more you'll get accomplished. However, this is also an obvious oversimplification. Some people prefer a typical 9-5, others thrive with a 10 hour/4 day workweek, and others still find their productivity falls off after 6 hours of work. Adding to this, our work can often be asynchronous, and working remotely can be integrated into a routine as well.

Regardless of your working style, taking breaks is important! Stretch, take a brisk walk, hydrate, and enjoy some caffeine (in **moderation**).

If you are working remotely for extended periods, communicate regularly on Slack, and schedule a few co-work times when you and others can log on together and work in a shared virtual space that makes spontaneous interactions possible. Remote work has advantages, but a major drawback is the lack of this spontaneity. Being able to walk to a neighboring desk and ask a quick question, take a shared tea break, and see the faces of the people you work with are real advantages to collaborative science. Some of these aspects can still be done remotely, but they require active effort. Please stay involved like this even if you aren't on campus!

Generally speaking, if you are meeting milestones, participating in project and group discussions, and comfortable with your work-life balance in our lab, the details of your work schedule are flexible. For a diverse group, a schedule that accommodates everyone necessarily requires compromise. Your exact schedule will come down to your role in the lab, your preferences, and the group's availability. There are some basic rules and tenants to be aware of.

8.5.3.1 Work hours

For full-time members (e.g., PRAs, postdocs, grads), expect to put in 40 hours of work per week on average. For part-time members (e.g., undergraduates during the semester, interns, and other volunteers), you might be in the range of 10-30 hours a week typically.

These ranges are estimates and will vary across people and over time. We never want to find ourselves in a crunch, but deadlines sometimes do pile up, and some weeks may require more hours. At the same time, some weeks will be lighter, breaks are encouraged, and working to the clock, or overtime should not be glorified either. Most importantly, maintain a comfortable work-life balance that leaves you wanting to come back for more, and keeps you far from the burnout line! If you ever feel stressed about your schedule/workload, talk to your PI & group to see how we can change things.

Presenteeism isn't a healthy mindset, and health and productivity rank ahead (in that order!) of simply being present. Similarly, preferring not to be at work might also be indicative of potential issues. Talk to your mentor(s) and labmates, in case you have any concerns regarding working in person.

- Working remotely | CU Remote Work Agreement

For any student-employee-related questions, please refer to the University of Colorado Denver student hourly employment handbook or the student employment website. Both handbooks define the student employment rules for the Denver and Anschutz Medical Campus.

8.5.3.2 Time Off

Always be a happy researcher! To be a happy researcher means you need a good balance between work and time off. We encourage you to take time off when needed, but there are a few things to remember. If there are meetings/deadlines involved, please let me (PI) and team members know ahead of time if you can't make it during crunch time.

- It's understandable that things occasionally come up – so it's for a few hours/day(s); feel free to drop me (PI) a quick note and take off as needed.

- If it's more than 1-3 days (or recurring hours/days), let me (PI) know in advance so we can plan ahead.

- If it's >3 days, please include me (PI) in the conversation during the planning phase (before booking your tickets/planning your travel). It's better for all parties involved if this is a conversation rather than an FYI. *Similar for >few days of remote work, unless previously requested.

- Mark your time off on the lab calendar (in a non-disruptive way – darkest color background and NO notifications per event for everyone in the group) and notify your PI and co-workers in advance.

- Let others know if you miss any meetings during your time off. Please do your best to plan your work accordingly when known vacation times (no meeting weeks) are approaching. Don't leave too much undone before going on vacation because you may have too much to do when you return, which can cause stress.

- Also, leave good notes on what you were working on and what you must do when you return. Sometimes, when you are gone for too long, remembering where you left off can be hard, so having these notes can help you get back on track.

- Studies have shown not having a sustainable work-life balance can lead to difficulties in maintaining an efficient, productive, and healthy research experience. Therefore, it is essential to maintain a healthy work-life balance to ensure the best and most productive research experience (Bartlett et al., 2021).

- Be respectful of your labmates and avoid sending them messages when they are on vacation. If you are working closely with them on a project or need their help, notify them of any deadlines that may overlap with their vacation. This way, meetings can be determined before your labmate leaves for vacation.

- If you receive work-related messages during vacation/holidays (incl. weekends), you are not obligated to answer them unless they are critical. If the message is urgent (esp. something you've committed to prior to your vacation and have missed checking during working days), please do your best to answer them as soon as possible. This is more so important when there are deadlines approaching. It is OK if there are occasional instances when you can't make it to a deadline – but this can't be a norm, and you *must* let me (PI) know in advance so we plan ahead.

8.5.3.3 Leave policy

- Because our group is tightly collaborative, please plan non-sick leave ahead of time to best fit all schedules when possible – at least check in with your mentor(s), mentee(s), and closest collaborators in advance.
- For graduate students: Details may vary per your specific graduate program's handbook, but the Graduate School offers basic guidelines herefor vacation and leave.
- Undergraduate and Masters Students: Student Employment Handbook
- Staff (postdocs, research professionals): The University of Colorado's leave policies

8.6 Publication, Authorship & Research Misconduct

8.6.1 Publishing

To quote the Foreign Secretary of the Royal Society, Dr. Mark Walport, "Science isn't finished until it's communicated". If work isn't documented, written up, peer-reviewed, and made publicly available, it's no different than if the work hadn't been performed in the first place. For our group, that might involve writing clearly commented, clean code, submitting it for code reviews, and sharing it through GitHub, or it could involve writing a manuscript and

submitting it to scientific journals. In any case, it is imperative to work towards sharing reproducible, repeatable, and replicable science.

8.6.2 Authorship

It's often said that publications and citations are the currency of science. How does one become an author on a paper? What even is a paper? When is a project ready to publish? Many of these questions are best answered through conversations with peers and collaborators, but there are resources available to help provide context.

- A paper, a publication, a manuscript, or whatever you might hear it called is a formal write-up of a scientific project. It typically follows a format referred to as IMRAD for Introduction, Methods, Results, and Discussion. These refer to the order of sections of the paper, which roughly presents the introduction to the problem or knowledge gap, the methodological approach or techniques the authors decided to use to address the issue established in the introduction, what the results and data outputs of those methods are, and the interpretation of those findings and how they relate back to the problem established at the start. While the exact formatting varies journal-to-journal, this structure generally captures how scientific results are written and reported.

- Having your name on a paper's author list typically requires a substantial intellectual contribution. Without your work, the story in the manuscript would not have been complete. Typically, the position of the authors in the list reflects the amount of credit they deserve for the work done, and the PI(s) who oversaw the project receive(s) the last author position(s) to signify that it was their lab(s) who performed the work. Many people make minor contributions that do not make the cut for the author list, and they may earn a space in an Acknowledgements section. Discussions about project responsibilities and the positions contributors will earn on a manuscript should happen early and often. The details of this vary from field to field and lab to lab, but a good overview can be found in this article, interviewing many different scientists across disciplines.

- What kinds of contributions are important? Researchers have asked this question enough times to lead to the creation of a framework called CRediT for Contributor Roles Taxonomy. While this isn't the final say on what types of contributions there can be on a manuscript, many journals have adopted the CRediT system to specifically assign which authors contributed in what ways in a paper, so it's definitely worth a look.

- What does it mean if you're an author? No matter your position in the author list, it means you are literally assigning your name to work performed. Even if you only worked on one small part of the analysis, you must be familiar with all aspects of the paper. Your authorship is a statement that you vouch for the content, the methods, and the conclusions made in the paper. While honest mistakes do happen and can slip through to publication, all authors should do their best to ensure everything is conducted to the highest standards for scientific rigor. Mistakes that significantly change

the interpretation or outcome of an experiment, or worse, intentionally misleading or fraudulent data, can seriously affect or even end scientific careers for authors on erroneous papers. Authorship should be taken seriously because contributing to the larger body of scientific knowledge can and does have real impacts on the world and lives in it.

8.6.3 Research Misconduct

- If you are concerned about misconduct by fellow scientists, talk to them first! Science is inherently collaborative, and everybody brings a unique perspective – you may notice something that others have missed, and simply asking about it can course-correct a project. If you don't feel satisfied with the answers, or if a researcher is not acting in good faith, it is your responsibility to report potential misconduct. In cases where you aren't certain but still have concerns, you can speak with a Research Integrity Officer at CU and discuss whether the issue qualifies as misconduct before escalating to a formal report.

9 How to X

9.1 Get started

If you are new to computational biology and R programming, join the `#bioinfo-primers` channel. Use the `#help-x` channels to ask for help/answer others' questions on R/Py programming, shell scripting, version controlling, or anything else that's general and not project-specific. Use the corresponding `#proj-x` channels for all project-related questions.

Check out the Slack tips in the pinned posts in #help-proj_management to get oriented since you are new to Slack (the channel used to be `#howto-slack`)!

To re-familiarize yourself with R/Python/Unix, check out these two resources in addition to Slack | CompBio gists and R-Ladies East Lansing. If you are new to R-Ladies Aurora, R-Ladies East Lansing, and Women+ Data Science, check them out and join Meetup-RLA/Meetup-RLEL and their Discord/Slack to remain apprised of upcoming events and to connect with the local R and data science communities.

If you find other useful resources, please add them to this list (or share them with me).

9.2 Code

- Write, work with, and manage source code and data
- Prepare for and carry out code review
- Share data
- Write an R package
- DBMI software engineering team tips & tricks blog posts

9.2.1 Learn R

To learn R, I would recommend getting started with the pinned posts in #courses-primers.

A couple of helpful GitHub repositories - https://github.com/jananiravi/workshop-tidyverse (Intro to `R's tidyverse package` — handy to learn right away) - Other R workshop materials from our R-Ladies East Lansing chapter: https://github.com/rladies-eastlansing - Interactive tutorials with learnr & swirl. - R for Genomics from Data Carpentry - R on HPC

- Coursera

    - [Intro to R programming and tidyverse](#)
    - [Data visualization and Dashboarding with R specialization](#)
    - [Getting Started with Data Visualization in R](#)
    - ... and more | explore by topic, duration, skill-level

Go-to books (also in pinned posts on Slack)

- [R for Data Science](#) (for tidyverse and such)
- [Hands-On Programming with R](#) (for base R) Intro to Git, GitHub resources

9.2.2 Intro to Machine Learning

- YouTube Videos

    - [Machine Learning from Scratch by AssemblyAI](#)
    - [MIT Deep Learning Course](#)
    - [Deep Learning by Assembly AI](#)
    - [Understanding Tensors by Dan Fleisch](#)
    - [Understanding Neural Networks](#)
    - [Basic Computer Parts Explained](#)

- Articles

    - [Transformers](#)

- Coding Practice Sites

    - [LeetCode](#)

9.2.3 Learn Git/GitHub

- https://happygitwithr.com/ Happy Git and GitHub for the useR (connecting git/GitHub w/ R)
- [Git and GitHub learning resources](#) from GitHub
- [Git 101](#)
- [Coursera Intro to Git and GitHub course](#)

9.3 Access HPC, servers

For computing that requires high performance hardware and computing clusters, we utilize two different remote servers.

9.3.1 JRaviLab server

- A single high performance machine used only by our group.
- To request access, contact the system administrator `shaddai.amolitos@cuanschutz.edu` by sending him your CU username (e.g., the CU auto-generated `ravijan` instead of `janani.ravi@cuanschutz.edu`) and a brief message that you will need access to the server. CC janani.ravi@cuanschutz.edu, so they can both verify your access.
- If you are new to remote server work, then it maybe helpful to learn about using an ssh config for managing your remote hosts.

9.3.2 Alpine HPC

- Alpine is the HPC cluster for all the University of Colorado campuses, including Anschutz.
- There is extensive documentation about getting setup
- At the moment, a web shell is the only access option for Anschutz members. This is different from a typical SSH login, and it is best to follow the official, latest Alpine documentation.

Whenever possible, we highly recommend using the VSCode extension which provides an IDE on the remote host without having to worry about server-side installation of other IDEs like RStudio or Jupyter.

9.3.3 Use VPN

https://www.ucdenver.edu/offices/office-of-information-technology/software/how-do-i-use/vpn-and-remote-access

9.4 Read/write/present

- Read papers
- Make a poster
- Write a paper
- Writing in the Sciences on Coursera

9.5 Plan your semester/year

Templates on yearly/semester planning and professional development are available here.

9.6 Others

- Career development week
- Remote work | CU remote work policy | CU DBMI remote work agreement

10 Resources

10.1 Funding

We strongly encourage trainees to apply for scholarships/fellowships or other grants to help support their independent research and to get acquainted with the process of developing competitive research and personal statements.

Here are a few scholarship and funding opportunities:

10.1.1 Grad students

- Graduate student funding opportunities (c/o JHU)
- @CUDenver
- NSF GRFP
- NIH F30, F31
- Graduate tuition for in-state, out-of-state, non-resident, international students
- Immigrants vs resident aliens

10.1.1.1 Training grants

https://www.cuanschutz.edu/graduate-programs/biomedical-sciences-program/resources/grants-and-fellowships

10.1.1.2 AWIS

https://awis.memberclicks.net

10.1.2 Undergrads/postbacs

- [Colorado Biomedical Informatics Summer Training Fellowship for URM](#) Mail Evelin Zumba with questions.

- [Loan repayment program](#)

- [University of Colorado Anschutz Medical Campus Preparation in Interdisciplinary Knowledge to Excel (PIKE) PREP](#) (URM + citizen + BS in biomedical field)

- [For high school, undergraduate, and graduate/professional students](#)

10.1.3 Postdocs

- [Beautiful central resource maintained by JHU](#)

- [Postdoctoral fellowships for international scholars](#)

- [NIH F32, K99/R00](#)

10.1.4 Other

- [Anschutz Funding and Fellowships](#)

- [Ludeman Center for Women's Health Research](#)

10.2 CU resources

10.2.1 Health and Wellness

CU Anschutz: [Mental health](#)| [Health and Wellness Center](#)

MSU: [Mental health](#)| [Student Health & Wellness](#)| [Fitness](#)

10.2.2 Women in STEM

10.2.3 JEDI

Justice, Equity, Diversity, and Inclusion

- Central Office of Diversity, Equity, Inclusion, and Community Engagement
- Office of Disability, Access, and Inclusion
- Office of Diversity and Inclusion
- DEI education, outreach, and recruitment

10.2.4 Writing Center at CU Anschutz

11 Offboarding

We are aware that career plans change, and that is perfectly OK. The Ravi lab always encourages you to pursue your future endeavors and dreams. Ensure you do your best to make Janani aware of any changes in your career plans so the offboarding process can run smoothly.

11.1 Ready to head out?

We are delighted to have been a part of your research life. We wish you great laurels as you move on to the next phase of your career.

Please make sure you have checked these boxes before leaving our group.

☐ Please return any computers, devices, cords, or accessories that belong to the lab.

☐ All your code is on GitHub (jravilab organization), and all your other non-GH docs are on our shared Google Drive folder. Please check to make sure none of the codebase or docs are on private Drive/GitHub repos (that are not part of our lab).

☐ All your processed data files/figures/reports (<100MB) are on GitHub.

☐ Large data files (raw/processed) are on the server (and backed up); check to make sure your project successors and I have read/write access to these files/folders on the server.

☐ All your scripts and data files are well-annotated with appropriate `README` files and `Mermaid` workflows, when relevant.

☐ Added a detailed status report of where things – what was tried, what worked/didn't work, where the scripts/data files are, what remains to be done (within the scope of the project, and clear next steps).

☐ You have reoriented me and at least 1-2 other project members with the precise status (and next steps) of the project and the location of files.

☐ You are not a full member of our GitHub organization, Slack workspace, Google Drive, or Outlook Calendar anymore but will retain access to the project channel/repo until publication (or complete transition to other project members).

☐ If you are switching from being a full member to a collaborator, you will now only have collaborator privileges on Slack, GitHub, and Google Drive, until you are done with the project/your term with the collaborator.

☐ If you have worked with us for a considerable period of time, you will remain a part of `#the-continuum` channel on our Slack.

☐ Please update us of any changes in contact information for authorship purposes (in case we need to reach you and add you to the list of authors for all the work you contributed to the published project) and for us to know where you are now thriving.

We appreciate your contributions to JRaviLab, and we have set you up to succeed in life! We wish you the very best with your career next steps!

11.2 Stay in touch

We would love to hear from our dear alumni – please share your updates/accomplishments on `#the-continuum` or stay tuned for others' news, too! And let us know when you're in Denver next so we catch up for a cup of coffee/tea!

Webpage | PI | Email | Mastodon | Bluesky | Twitter (inactive)

A Meetings

Please update this semester's meeting schedule here.

A.1 Format

A.1.1 Weekly lab meetings

- These provide a space for everyone to share updates and roadblocks, and depending on the needs of the group, these meetings can be in a roundtable format where everybody who has updates can share them briefly (<10min per person), longer form updates or presentations for 1-2 people (<30min per person), or the entire slot can be reserved so one person can give a practice talk or major project summaries (<60min per person). We operate a Google spreadsheet where members can put themselves down for specific topics, times, and dates. In all cases, these weekly lab meetings are required for all members to attend (it is flexible for part-time and junior members such as undergrads with heavy class loads).

A.1.2 Weekly project-specific meetings

- These give lab members under a broadly shared project/topic umbrella a space for more detailed discussions and to seek specific feedback and help from their peers in the same or similar topics. These meetings are only required for those in the relevant associated projects, but are open to any members curious to sit in, learn, and contribute. Like the weekly lab meetings, lab members can schedule themselves for specific topics or times here. Because of their more focused scope, there may be times where these meetings can be quick updates dropped in Slack instead, and all involved lab members should communicate regularly to ensure each meeting is necessary.

A.1.3 One-on-one meetings or small group meetings

- These more direct meetings can be set up as-needed. To easily schedule a one-on-one with Janani, please use this booking link.

A.1.4 Journal clubs & Primers

[Sign up here](#)

Journal clubs

- Journal club serves as opportunities to present interesting new research papers from other groups. For journal clubs, all members of the lab must read the selected paper ahead of time and be prepared to help their peers explain it. Likewise, to be respectful of the time commitments of others, the presenter must share their selected paper at least a few days before the meeting, and preferably the week before to give everyone a chance to read it thoroughly.

- The format of journal club is largely up to the presenter, but it should walk briefly through the background to the problem, the question or need the paper addresses, how they did the work, what the results were, and how it ties back to the larger problems in the field and future directions. While the presenter can take the lead and walk the group through everything, journal clubs are a lot more engaging and educational with participation, so having the audience explain figures and interpretations is an easy way to ensure people are learning instead of just listening.

Primers

- Technical primers can be great practice on teaching a topic by sharing how-to or tutorial lectures on subjects of your choosing. These are great opportunities to share a skill or approach with the group so we can all benefit from it.

- Additionally, these are spaces where we can invite an outside presenter to share with us (for example, the [Department of Biomedical Informatics' Software Engineering Team](#), or the [CU Office of Diversity, Equity, Inclusion & Community Engagement](#), or a scientist with subject expertise from another lab group). It is up to the group collectively to decide which primer topics are appropriate.

A.1.5 Co-work sessions

- Organized as desired, and can be in-person or virtual. Have a paper or a grant deadline? Need to do some code review? Looking to catch up on recent literature? A co-work session is a great place to get these done in a focused but still social setting. These typically work best if you bring a task or to-do list, but are also good to set up regularly if you're working remotely for any extended periods of time to keep in touch and allow more natural interactions with your colleagues.

A.1.6 Weekly software engg meetings

- Meetings with the DBMI SET on software and UI/UX design and implementation, as well as work on our JRaviLab server, software applications like MolEvolvR, or work on the CU Alpine HPC cluster.

A.1.7 (Bi)weekly/monthly collab meetings

- Less frequent except during sprints, these are meetings with our collaborators on joint projects. They may involve sharing goals, blockers, and milestones, or might involve a collaborator working as a consultant to give advice or feedback from an outside perspective on a project.

A.2 Logistics

- GitHub issues/project for detailed project to-do's
- Shared Outlook Calendar for meetings, deadlines, RLEL & other events!
- Semester planning meeting (once per sem, during the first month)
- Slack for everything else!

A.3 Schedule and times

Always confirm the exact meeting schedule with our shared Outlook Calendar and Sheets document, but below are the usual meeting times.

A.3.1 Monthly/(Bi)Weekly individual check-ins:

Book via Outlook bookings.

A.3.2 Group meetings

- Weekly group meeting (All members) | Fri 10.30a MST (wk)
- Journal clubs/Bioinfo primer series | Thu 11a MST (wk)

A.3.3 Project meetings

- Software engg (MolEvolvR, server, & such) | Tu 11.30a MST (wk) | DBMI-SET (CU)
- EEPID | Tue, 12.30p (wk) | w/ WaleLab (MSU)
- Disease-gene associations + ML | Thu, 9a MST (biw) | w/ KrishnanLab (CU)
- Host response and disease/drug signatures | Thu, 10a MST (wk) | w/ KrishnanLab (CU)
- Microbial G2P (MolEvolvR/CompGenomeR/AMR/MicroGenomeR) | Fri 11.30a MST (wk)

A.3.4 DBMI seminar schedule

Posted by the DBMI seminar committee | 2023-24 schedule

A.3.5 Monthly Group Socials

<TBD | spontaneous>

- Celebrations (w/ coffee/food/snacks)
- Random discussions/catch-up
- Coffee/tea or ice cream for local folks
- Once in a while, we can dedicate the group socials to fun topics close to your heart (e.g., music, sports, books, food, …)!
- Online games: https://jackbox.tv | https://drawbattle.io (was more popular during the pandemic)

B Code Review

B.1 Why do it?

- **Pride**: Lab members (along with co-PIs) sign their code/software via attributable commits. This means we are happy to and should stand behind our code, whether private or public.
- **Licensing**: The lab code/repositories carry a LICENSE file with the PI's labs. Code review helps stand by the licensing agreement. Also, when using dependencies (other published, in-dev packages, codebases, be sure to read and follow their licenses. We can only be as inclusive as the least inclusive cited resource.
- **Attribution**: When using code taken from elsewhere (other repositories, SO, papers, or alternate sources), make sure the code and coder are appropriately acknowledged. Within code attribution is important. If this is substantial, we need to add them to other centralized attribution files, as well.

B.2 Preparation for Code Review

Things to be done by the author *before* code review. Please note that all repos should remain private by default. Students, postdocs, and staff should discuss with Janani about when to make each repo public. By default, the code contributors and the PI will be co-authors on all software and codebases.

- **Adhering to "good enough" practices:** Make sure you code adheres to the checklist of good-enough practices. Enforcing basic style rules is typically not a good reason to do code review.

  - For e.g., this means everyone follows styler/lintr, Google's R Style Guide, or Python PEP8, or the Javascript, HTML/CSS equivalents for basic cleanly formatted code.
  - Use a max line length of 80 (or 100) to make sure the code is readable without wraps.
  - Make sure there are no unnecessary white spaces (and that necessary ones are present).
  - Are your variable and function names descriptive and interpretable?

- **Common sense debugging:** Catching bugs is not a good reason to do code review unless you've tried all of the following:

  - Have you created a small toy/example/test dataset, ran your code, and matched the output with hand-calculated results? You need to have this dataset handy during code review.
  - Are there off-by-one errors? Python and R begin their indices differently.
  - Do the loops terminate in the way you expect? Do they terminate at all?
  - Have you declared and initialized all variables properly? For all a, b, c, & d, it is worthwhile peppering the code with a number of print statements to follow along and see if each step checks out.
  - Have you run automated tests, debuggers, etc.?
  - Do you have top-level and within function/code-chunk comments? This includes file comments to include what each file does broadly and how to run them from the command line, docstring comments for functions, and in-line comments for detailed logical steps within the code chunk.
  - Are all dependencies and imports defined in the R package files or at the top of the file?

- **A visual flowchart of the code:** Before you ask anyone else to look at your code, you need to be able to clearly explain to them what your code (or part of it) is meant to do. If it is a long piece of code, you should bring a drawing of which methods call which other methods, or which objects use which other objects.

- **Prioritizing the goals of code review:**

  - If you're trying to use code review to gather thoughts on speeding up some calculation, but your teammates are thinking about catching bugs, neither of you will get much from the process.
  - Good reasons to do code review:
    * To help learn how to think like a coworker so that it's easy to navigate and change their code in the future.
    * To bring up to speed on what files and features have changed recently so that when bugs sneak in, at least two people can help in diagnosing and fixing the problem.
    * To get critical feedback on a specific section that requires improvement in terms of speed-up, unfathomable errors, etc.

B.3 Code Review Process

Involves the author and at least one reviewer (ideally someone related to the project)

- Set up a time and set a time limit (say, half an hour or one hour maximum; no more than 400 lines of code should be reviewed at a time). Then, divide the process into three phases:
    - Phase 1: The author should give a description of the code/feature (using the visual flowchart of code, if necessary) including the list of input files, output files, and usage. This should include clearly stating the primary goals of the code review to spend time more efficiently.
    - Phase 2: Reviewers do a first pass of the code along with the author trying to understand the changes (or the whole thing if it's the first time) and write down questions (you can ask questions for clarification but do not give feedback yet). If you think you can approve the code/change at this point, do it and be done.
    - Phase 3: If you think the code is not ready, list your thoughts or questions to the author and engage in a discussion in the order the code is designed to run.

- Practice a positive code review culture
    - Each bug or clunky piece of code is an opportunity for the team as a whole to improve code quality. Ask questions, acknowledge good practices, politely suggest fixes, and explain your reasoning.
    - The process allows junior team members to learn from senior members. So, seniors, set a good example by making it clear that this whole exercise is about the code, not the person.
    - Practice the art of giving and receiving critical feedback: i) Give mutual respect, ii) Be specific, iii) Follow the keep-discard-improve rule, and iv) Have a "how can I help you?" mindset.

B.4 If you have any questions

Please reach out to me (JR) if any of these aspects are unclear before requesting a code review.

B.4.1 Acknowledgments

C Code/Data Best Practices

Sourced (borrowed unabashedly & modified) from the excellent material from the Greene & Krishnan Labs + the rest of the references below, combined with lessons I have learnt from my own experience through the years!

C.0.1 Pride

We expect lab members to sign their code. To quote from *The Pragmatic Programmer*, "Craftsmen of an earlier age were proud to sign their work. You should be, too… People should see your name on a piece of code and expect it to be solid, well written, tested, and documented." While some code will be proof-of-concept code, it should be of a form that inspires confidence.

And this, I cannot emphasize enough: Pride also means clearly showcasing what is inspired by, a modified/improved version of, and/or directly borrowed from another software code base. This means that when we write code by building on top of what others have done (via inspiration, modification/improvement, or borrowing), we clearly give credit by:

1. Explicitly saying this in the comments section next to the relevant piece/block of your code, including a link to the source and a short write-up on how the source is adapted here.

2. Including these detail in the documentation of the code base that lives as a supporting document outside the code.

3. Including a citation to the source wherever appropriate in our publication. Of all these, [1] is the easiest place to be extremely generous and complete in attribution to everything non-obvious (fast implementation, numerical recipes, methodological/algorithmic ideas) from StackOverflow answers, blog posts, lecture notes, existing open software, etc. We will start impeding readability and hit space constraints with [2] & [3]. Please talk to me or other senior members of the lab about these to decide on proper attributions.

C.0.2 Language

We write code for our analyses in Python and R, which allows everyone in the lab to know two languages and understand analytical code. Code for visualization can be Python, R, or javascript. Webserver interface code uses R Shiny or javascript. Check out this doc for all the material you need to get started with most of these languages and tools.

C.0.3 Licensing

We expect code that we produce to be licensed under a 3-clause BSD license. Unless a funding agency requires something different, we'll use this. If you have questions or concerns about licensing, feel free to raise them in Slack.

C.0.4 Version control

We have a JRaviLab account on GitHub. We expect that lab members will maintain their code in repositories under this team account. When you commit code, add informative messages. Remember: verbose is better. Even if you're just fixing formatting or correcting typos, instead of phrases like "small changes" and "some minor stuff", just say "fixing formatting" or "correcting some typos". We will only publish code that is held in a public JRaviLab repository that has gone through the code review process.

C.0.5 Data management

For publicly available data, scripts used to download and process these data should be preserved, as should the versions of helper files used in processing (e.g. probe-to-gene mapping, gene-symbol mapping). [If the dataset is large (>100 GB), discuss with me about where to download it.] These items – processing scripts, helper files, etc. – should be version controlled. Where possible, intermediate files of reasonable size can be stored to facilitate re-use, but the process to regenerate these files from publicly available data should be preserved. Keep the following things in mind:

1. Do not tamper with original files that you get from me, a collaborator, or from external resources. In the folder where you're downloading these original files, create a `readme.txt` file that contains detailed information on when you downloaded this data and from where. For example:

    1. If you got this from a link to a Google-Drive/DropBox folder from me/collaborator via Slack/Email, along with the date you downloaded, note the link to the GDrive/DropBox folder and the link to the Slack message or Email.

        1. Slack message link: click the . . . , then "Show message actions" next to the Slack message and click "Copy link".
        2. Email message link or ID can be obtained by right-clicking the message or looking under "Details".

    2. If you get a link to the data on our MSU servers, create a symlink of the data in your data directory (do not copy over the data; it's a waste of space).

2. Do **not** make changes to any file by hand. Write a shell/Python/R script that reads in the file and generates the desired new file with all the required modifications. If you get a file in formats that cannot be worked with easily using a code (like Excel sheets), export the sheet as a text document and then work with that text file.
3. Automate everything so that you can exactly reproduce everything at a later time. Create a `runlog.sh` file in each folder that has the list of all the commands (shell commands: the full commands to run your Python/R script) you executed in that directory.
4. Many times, we will have to use the scratch space to download large data dumps and process them. Scratch has better I/O but has no backups. So, remember to keep the processing scripts in your project directories, symlink the scripts onto the directories on scratch and use them there. Thus, even if scratch fails, we can rebuild everything seamlessly. Talk to me before moving over huge processed datasets from scratch over to the backed-up research directories.
5. Give all files meaningful, interpretable, and computable names.

C.0.6 Organizing your project

Adhere to the following organization for your project folder/repository. Each project is slightly different but should not depart too much from this proposed organization.

- project_data
  - data
  - src
  - bin
  - doc
  - results

Check out the detailed doc here.

This whole project directory, *except* the big datasets in your data directory, should be synced with your GitHub repo of your project.

C.0.7 Reproducibility

We expect all lab members to maintain code that performs reproducible analyses. This can be in the form of R/Python/Shell scripts to do everything without manual editing and `runlog.sh` scripts that contain the command-line calls of all the scripts with inputs/arguments that allow analyses to be automatically performed. We expect that these scripts, including those to generate figures in papers generated as a consequence of such analyses, will be included in source control repositories and made publicly available before or concurrent with manuscript publication.

C.0.8 Good enough practices for writing code

1. **Pride**: We expect lab members to sign their code.
2. **Using other code:** Code taken from elsewhere is properly acknowledged and should be checked for compatibility with its license.
3. **Style guide:** Python code follows PEP 8. R code follows Google's R Style Guide.
4. **Variable and function names**: Variable names are descriptive and interpretable to someone looking at this code for the first time (e.g. not `a`, `b`, `x`, etc.) They should be full words (`nodeDegree` or `pvalues`) or clearly recognizable acronyms. Function names should begin with a verb (e.g. `parse_expression_dataset`, `shuffle_list_of_genes`, or `get_score_distribution`). In both cases, be as verbose and expressive as you need.
5. **File commenting:** Each file has a comment (a small paragraph) at the top to broadly describe its purpose and how it is expected to be used (e.g. imported, run from command line, both) including details on inputs/outputs and example usage. This is also the place where you make dependencies and requirements explicit.
6. **Function/code commenting:** Each function has a docstring that reports the computation that is intends to implement, its arguments, and its return value(s). E.g., `#' Takes in the gene_length dictionary & a geneset_gene dictionary and returns a random geneset for each real geneset with the random genes having similar lengths as the real member genes.` Docstring documentation is here: Python, R.
7. **In-line commenting:** Each block/section of code contains detailed comments on what it is meant to do.
8. **Imports:** All trivial imports are at the top of the file right after the top comment paragraph.
9. **Code with constants:** Any constants are specified at the beginning of the file following the imports.
10. **Code that uses a random seed [special case of constants]:** Code that uses a random seed is reproducible. This means that the seed can be set and a default value is specified. This needs to be done at the top as well.
11. **Column length:** Lines are 80 characters or fewer. This applies to all text under revision control with the exception of data files that must adhere to a particular file format that does not allow for line "folding" where necessary. This rule is already covered well in PEP 8 but called out here to clarify that we apply it to more than Python code. One reason for this is to aid in readability of `diff` output when performing code reviews.
12. **Whitespace:** There is no unnecessary whitespace.

D References

- The KrishnanLab and GreeneLab Onboarding Docs
- A Guide to Organizing Computational Biology Projects
- Good Enough Practices in Scientific Computing
- Best Practices for Scientific Computing
- Ten Simple Rules for Making Research Software More Robust

E Remote Work

updated during/due to the COVID-19 pandemic.

F Few generic tips

1. Focus on health – physical and mental
2. Setting goals & planning a 'tight' schedule
3. Figure out what's accomplished and what not on a weekly basis & start the process over
4. Find accountability partners
5. Maintain good contact with me and the group
6. Relax and meditate! [my audible/kindle account details are shared via Slack]

G #WorkFromHome tips

~ Most tips may also be applicable to your coursework.

G.1 1. Daily Scrum

At the beginning of each day you're working, think or write down the following:

1. What did you do yesterday? ["Yesterday, I …"]
2. What will you do today? ["Today, I'm going to …"]
3. Are there any impediments in your way?

G.2 2. Casual Co-work

Members are encouraged to setup co-work times to co-write, for code review, or just to work! (based off our most recent When2meet poll)

G.3 3. Finding time to chat with me

Finally, as a substitute for scheduling in-person meetings via Slack, I've made my full calendar available here, where you can easily fix a time to chat.

- No need to ping me ahead.
- No need to have a specific purpose/agenda, although this might help me prepare if it's for a scientific/technical check-in. We can talk about anything – research, courses, work-life, home-life – or just say "Hi!"
- Pick a corresponding slot – 15min, 30min, or 1h.

G.4 Acknowledgments

Our JRaviLab group discussions + ideas from Baucom and Krishnan Labs.

H Make a poster

H.1 Questions to address

In the poster, clearly and concisely explain what you have done in terms of the 4 questions:

1. What is the problem? why is it important/significant? | Background/Overview

2. What are the current approaches to address this problem & what are their limitations? *aka* Gaps in knowledge | Background/Overview

3. What is your approach, what are the results, and what are the take-homes? | Workflow, Results sections, Summary/Conclusions (>60% of your poster)

4. What are the next steps and, in general, who cares? tying it back to the big picture | Future Directions/Significance

H.2 Main elements of the poster

1. Title, authors, affiliations, group, university logo
2. Background/Overview (w/ fig)
3. Approach/Methods w/ your flowchart/cartoon workflow | methods/tools can be spelled out for each logical step of the pipeline
4. Results, Results, Results | in various shapes and forms, but mostly said with clear readable figures, screenshots, tables (if plots can't be made)
5. Summary/Conclusions & Future Directions | separately or together based on how much you have to say for each
6. Acknowledgments & References (could be together/separate)
7. Contact info – group's, yours, … how/whom to reach if they have questions.

H.2.1 Other Tips

- Font sizes readable at 100% zoom from a 2-3ft distance
- Clear BIG section headings
- Use colors, but 'wisely!'
- Figures > flowcharts > bullet points »> blocks of text (avoid)

- Where to make? Google Slides (when collaborative + preferred), Keynote/PowerPoint

H.3 References

- [How to Create a Research Poster: Poster Basics | NYU Libraries](#)
- [Ten Simple Rules for a Good Poster Presentation | PLoS CompBio](#)
- [Simple Rules to Make an Effective Scientific Presentation | my slides](#)
- [Poster/Lightning Talk Guidelines from Women+ Data Science Symposium](#)
- [A guide to poster making for scientific conferences | Sina Rueger](#)
- [Tips for presenting a scientific poster at a conference | Scientifica Neurowire](#)
- [How to make an effective poster | UC Davis undergraduate research center](#)

H.3.1 Example posters

To be added.

I Writing papers

I.1 How to write a scientific paper once we decide that you have one or more interesting results

Start by discussing with me what the paper is about!

I.1.1 Figures and Tables

- Create a draft of your ideal set of *Figures* & *Tables* (in addition to supplements) that can tell an interesting, coherent story.

  - Write detailed figure/table legends to help understand what piece of the story each figure/table will convey.
  - Get feedback on the entire story draft from colleagues / collaborators / committee.
  - Based on these interactions, keep revisiting & revising this draft.

I.1.2 Introduction & Discussion

- Read widely and deeply. Read often.

  - Remember that this exercise is going to entirely feed your *Introduction and Discussion* sections.
  - Record all the papers in a *Zotero* library, starting with Slack shares and periodic PubMed searches.
  - Create a single *Google Doc and make notes about each paper* along with its title & link. The purpose of the notes is to make the points you will write in your manuscript to cite each paper.
  - Based on these readings and discussions during project meetings, *pick a target journal.*

I.1.3 Supplementary data

- Make all main and supplemental *Figures and Tables* along with declarative titles & detailed legends.

    - Run many controls and sanity checks.
    - Rapidly iterate with me & colleagues and do additional analyses needed to fill gaps & round-out the story.
    - Prepare for & do code review.

I.1.4 Methods & Code+Data

- Write a very detailed *Methods* section and prepare *Code & Data* to be released.

    - Decide (w/ me) what *Code & Data* need to be released with this manuscript.
    - As you write each sub-section of *Methods*, organize and document the pertinent code, data, and results.

I.1.5 Results

- Write the *Results* section and rapidly iterate with me & colleagues to make changes/improvements.

I.1.6 Tying them together

- Start with the story line.
- Make the Figures/Tables (you will keep iterating over these until submission).
- Write the Methods since that's the easiest one to tackle.
- Then, write the Results describing your Figs/Tables.
- Finally, write the *Introduction and Discussion* sections, and add *References* (*e.g.,* Zotero + Google Docs integration).

    - The Introduction section should lead up to the main questions and results of the manuscript.
    - The Discussion should put the new results in the context of existing work, describe novelty & potential impact, and conclude with opportunities for future work.
    - Perform any additional analyses/comparisons needed to make these sections airtight.

I.1.7 Title & Abstract

- Write *Title and Abstract*, and get feedback from me/colleagues/collaborators on this full document.

I.1.8 Last Steps

- While others are critically reviewing the manuscript:
  - Create a well-organized & documented companion *GitHub repo* for the manuscript.
    * This should contain all the *Code & Data* (from Step 4) that will be publicly released.
  - Write the *Cover Letter*.
- Submit on bioRxiv, share on social media, and submit to journal!

I.2 Resources

Much has already been written on how to think about and go about writing good papers. This doc contains pointers to great resources on these aspects.

I.2.0.1 Getting Started – General Thoughts and Principles

1. How to write a first-class paper – Six experts offer advice on producing a manuscript that will get published and pull in readers.
2. Ten Simple Rules for Writing Research Papers – Principles and attitude that can help guide the process of writing in particular and research in general.

I.2.0.2 Writing the Paper – Sections and Structure

1. The CommKit is a collection of guides to successful communication in the biological sciences. It contains an entire section on writing Journal Articles with annotated real-world examples. Highly recommended!
2. Ten Simple Rules for Structuring Papers – Clearly laid out structure for papers.
3. Writing Scientific Papers — Structuring, Drafting, and Revising

I.2.0.3 Nailing Key Sections and Fine-tuning

1. The 5 pivotal paragraphs in a paper – There are five really pivotal paragraphs in any paper. If you get those five paragraphs right, you are likely to have and communicate the story arc in a way that grabs attention.
2. How to write paragraphs in research texts – Essay on crafting solid paragraphs.
3. How small changes to a paper can help to smooth the review process – Aspects of presentation that make it as easy as possible for editors and reviewers to focus on the work's scientific content by avoiding confusion or misinterpretation (generate substantial delays in the review process).
4. How to write an effective abstract – An excellent annotated abstract from *Nature* detailing all the components & flow of a good abstract.

I.2.1 Reading, Writing, Reviewing, Sharing Data, Creating an R pkg | from Jeff Leek

- How to read papers
- How to review papers
- How to write your first paper
- How to share data
- How to write an R package

I.2.2 Acknowledgments

Cited sources above + KrishnanLab.

J Career/Professional Development

J.0.1 Define and Measure Immediate, Tangible Growth Opportunities

J.1 Writing

☐ Update CV & submit it for feedback
☐ Perform and submit a self-evaluation: strengths, weakness, and career goals (w/ time-line)?
☐ Create a blog or submit a blog post about your project or lab experience to be posted on the lab's blog
☐ Read and review an article and post your comments on PubPeer or bioRxiv

J.2 Presenting

☐ Prepare for an elevator pitch
☐ Volunteer to give a talk (campus event, seminar, or elsewhere for outreach)
☐ Sign-up for mentoring and outreach

J.3 Networking

☐ Join Mastodon/BlueSky/Twitter and post a message
☐ Make a simple website (or LinkedIn profile) promoting yourself and your science/technology
☐ Initiate contact with a scientist/professional outside the lab/institution (whom you don't know before) – get career advice, scientific advice, or a letter

J.4 Other

☐ Join a professional society

J.4.1 Credits

Avasthi and Krishnan labs.

K LLM use guidelines for research trainees

Enhancing learning without compromising training potential.

A brief, meta compilation of LLM guidelines using an LLM!

> **Note:** At CU as of December 2025, only MS Copilot (and potentially GitHub Copilot) is approved for use. Do not use other public LLM systems with university research data. Inspiration: KrishnanLab LLM guidelines.

K.1 Core principle

Your primary goal in training is to develop independent scientific thinking, not to maximize efficiency. LLMs are tools that should **amplify your capabilities**, not replace the intellectual work that builds expertise.

K.2 Strategic uses (enhance learning)

K.2.1 Documentation & organization

- Clean up (not draft) READMEs, code comments, and inline documentation
- Organize project directories and file structures
- Create documentation templates for GitHub repos, datasets
- Generate boilerplate code structure (after understanding fundamentals)

K.2.2 Learning & skill development

- **After independent attempts:** Get explanations of complex concepts
- Generate analogies to understand difficult topics
- Brainstorm approaches to problems (but verify with literature/experts)
- Use tools like Perplexity to generate reading lists (always check for predatory journals, hallucinated citations)
- Ask "what topics do I need to know to understand this paper/method?"

K.2.3 Critique & gap analysis

- Get feedback on drafts you've already written (after 2 revision rounds with colleagues)
- Identify logical gaps, clarity issues, or missing considerations
- Check for completeness in project proposals or documentation
- Request alternative perspectives on your interpretations

K.2.4 Code assistance (after mastery)

- Debug assistance when you've already diagnosed the problem area
- Syntax help for languages you already understand
- Code refactoring suggestions (when you understand the tradeoffs)
- Standard visualization templates (after learning plotting fundamentals)

K.2.5 Communication practice

- **Voice mode for talk practice:** Deliver presentations, get feedback on flow, narrative, pacing, clarity
- Practice for comprehensive exams or conference talks
- Get suggestions for improving scientific communication style
- Polish grammar and style (like an advanced Grammarly)

K.2.6 Literature search

- Generate lists of related papers to explore (verify all exist)
- Find connections between research areas
- Identify key terminology and concepts in new fields

K.3   Avoid (compromises training)

K.3.1 Writing & thinking

- Having LLMs write any first draft (manuscripts, proposals, abstracts, reports)
- Generating content from bullet points without writing yourself
- Summarizing your own results or data interpretations
- Writing discussion/conclusion sections
- Any writing task you've done <10-20 times independently

K.3.2 Code & analysis

- Generating analysis code for methods you don't understand
- Writing entire scripts/pipelines without knowing each component
- Using AI for statistical approaches you can't verify
- Debugging without first attempting to understand the error yourself
- Any coding task you've done <5-10 times independently

K.3.3 Data & results

- Uploading raw research data to public LLM systems
- Having AI analyze, visualize, or interpret your experimental/computational data
- Using AI for any task involving sensitive, unpublished, or controlled-access data

K.3.4 Core scientific skills

- Bypassing reading original papers (especially in the first 1-3 years)
- Using AI instead of asking labmates/mentors for help
- Generating hypotheses or research questions
- Tasks where discussion with colleagues provides more learning value

---

K.4 Critical requirements

K.4.1 1. Accountability & verification

- You are fully responsible for ALL AI-generated content
- **Verification requires expertise** - if you can't verify output correctness, don't use AI for that task
- For code: Understand every line, test thoroughly
- For writing: Fact-check every claim, verify every citation exists
- For analysis: Verify statistical approaches, check assumptions

K.4.2 2. Documentation

When you use AI, document:

- Tool name and version
- Date of use
- Prompts used
- Output generated
- How you verified/modified the LLM output
- Errors found and corrected

K.4.3 3. Communication

- Inform your PI within 1 week of AI use for research tasks
- Be transparent with collaborators before/during work
- Never present AI outputs as your own understanding

K.4.4 4. Protect sensitive information

Never input into public AI systems:

- Unpublished results or data
- Proprietary datasets or code
- Novel research ideas or hypotheses
- Patient data or controlled-access information
- Grant proposals or manuscript drafts in development

K.5 Career stage guidance

K.5.1 Early stage (years 1-3, undergrads)

- Focus on building foundational skills **without** AI for core competencies
- Use AI primarily for learning/documentation, not execution
- Default to asking your colleagues (or PI) first

K.5.2 Mid-career (advanced grad students, postdocs)

- Use AI augmentatively for tasks you've mastered
- Still avoid AI for novel methods or approaches you're learning
- Emphasize critique/feedback uses over generation

K.5.3 Wet-lab specific

- Use AI for experimental design documentation
- Literature mining for method optimization
- Protocol organization and note-taking
- Never for data analysis/interpretation without computational expertise

---

K.6 The decision framework

K.6.1 Before using AI/LLM, ask yourself

1. **Is this a skill I need to develop?** If yes → do it yourself
2. **Have I mastered this through 10+ independent attempts?** If no → do it yourself
3. **Will using AI prevent intellectual struggle that builds understanding?** If yes → do it yourself
4. **Could discussing with a colleague provide more learning value?** If yes → talk to colleagues
5. **Am I using AI because the task is hard** (bad reason) **or because I've mastered it** (acceptable)?

K.7 Remember

- **Speed ≠ learning.** Efficiency now can mean skill gaps later
- **AI outputs look polished but may be wrong.** Confidence ≠ correctness
- **What distinguishes you:** Deep thinking, original perspectives, robust foundational skills
- **AI often introduces subtle errors that a human would not: these can lead to profound consequences**
- **Hallucinations are real:** Always verify citations, facts, and technical claims
- **When in doubt, ask your PI first**
- Check LLM's *Settings → Privacy/Data controls* and turn off chat history, memory, personalization, and training-related data usage for maximum privacy and no data retention.

The goal isn't to avoid AI entirely — it's to use it strategically so you emerge from training as an independent scientist with distinctive capabilities, not someone dependent on tools they can't verify or correct. You will have ample opportunities as a PI or a scientist in an industry, to learn and use LLMs quickly and efficiently — there's no need to rush into it now at the cost of your successful training.