Schiller Research Group

Expectations for Graduate Research Assistants

Hours

As stated in the Graduate Handbook, the normal half-time (50%) graduate assistantship workload is 20 hours per week. Students are sometimes hired for 25% (10 hrs) or 37.5% (15 hrs) of full-time work, under appropriate circumstances. The number of hours per week associated solely with duties for the assistantship is indicated in your contract. In some cases, you may need to put in more hours in a given week in order to meet deadlines. There may be overlap between your assistantship duties, your coursework, and your dissertation research and writing. Students should be aware of both their academic and work obligations and are encouraged to discuss any problems with faculty.

Vacations

In addition to days off when the university is closed, students are allowed up to two weeks of vacation time each year. These days should be scheduled with approval of their advisor and these vacations must not interfere with TA/RA responsibilities. In some cases, deadlines may require that graduate research assistants work on university holidays. In such cases, we will arrange a make-up vacation day at a mutually agreeable time. Graduate students are expected to verify all vacation days with the advisor prior to making travel arrangements.

Flex Time

There may be times during the year when assistantship duties must, by necessity, occupy more or fewer than the contractual number of assistantship hours allocated to a single week. Examples of such necessities include but are not limited to finalizing a paper or grant submission prior to a hard deadline and travel for fieldwork, conferences, or job interviews. Assistantship hours that are missed to accommodate other projects must be made up as soon as feasible, ideally in advance. Assistantship hours that exceed the weekly expectation will be banked for use as flex time in a mutually agreed-upon future week. It is the obligation of both the advisor and the research assistant to anticipate and communicate the need for flex time as far in advance as possible.

Goal-Setting and Performance Reviews

At the start of the assistantship, and at the start of each semester subsequently, we will have individual and/or team meetings to set task-based goals for the upcoming semester. At the end of each semester, we will have individual meetings to review your performance of assistantship duties. We will also have quarterly individual meetings (at the start of January, April, July, and October) to set personal goals for performance or skill improvement in your development as a researcher and to review growth relative to those goals.

Laboratory Notebooks

Laboratory Notebooks (paper or electronic) are required to record and document all research activities. This is not limited to wet or dry lab experiments, as some people think, but applies to simulations ("computer experiments") as well as to theoretical research [3]. The lab notebook serves as a legal record of ownership of ideas and results. It can serve to determine authorship in scientific papers or to establish copyright or patent rights. You need to keep a record of how every result was produced, including details of the mathematical and computational models, all parameters of the simulation setup, and all steps performed for data analysis. Record every step and every detail in order to ensure that your results can be replicated, both by other scientists and by you at a later time.

As a general guideline, I recommend to organize your lab notebook in "spikes". This term is borrowed from agile software development and refers to a period of research with a demonstrable output. Rather than partitioning the lab notebook by days or weeks, spikes allow some flexibility in timing but should always represent a coherent unit of research activity. The duration of a spike should not be longer than a week in order to keep the associated information manageable. The spikes are recorded in the lab notebook in chronological order. Every entry must be recorded with

- Start and end date;
- Subject of the spike;
- Protocol to reproduce results:
  - pointers (links) to all data files that are created or processed;
  - pointers (links) to all input files and parameters;
  - pointers (links) to the exact versions of all analysis and plotting routines;
  - pointers (links) to the exact versions of all software used;
- Thoughts and ideas related to the research problem;
- Notes from seminars, meetings, discussions;
- Literature references.

Create and maintain a table of contents for your lab notebook.

The lab notebook belongs to the university, so protect your lab notebook. Paper notebooks should not be taken home. Electronic notebooks must be backed up so they can be recovered in case of data incidents. Notebooks must be turned in upon graduation, or when the student takes a leave of absence or withdraws from the university.

Data Reproducibility

Scientific claims must always be verifiable by independent researchers. Therefore, it is of utmost importance that all your results can be reproduced. I cannot stress this enough. Reproducibility has to be the overarching objective of all your research activities. One requirement is that you organize and document your work properly. Since most of our data is in electronic form, you need to develop a systematic approach to storing your data [1].

As a general guideline, I recommend a multi-modal system that uses a **hierarchical directory structure** [2]. Each project will get its own root directory for data on our group storage. The project directory will commonly contain the following subdirectories

- `doc/` Manuscripts and documentation (project specific)
- `src/` Source code (project specific)
- `bin/` Executable binaries (project specific)
- `data/` Data, including scripts to re-run simulations and to replicate analysis

The `data` directory is organized chronologically with subdirectories following a YYYY-MM-DD naming scheme. Each of these subdirectories will correspond to a record in your lab notebook. The `data` directory can be further structured to separate input data, raw output data, processed data, final results, and graphical representations. All intermediate results should be recorded in standardized formats. In particular, keep the data for all graphical representations. It is mandatory to include all scripts that are needed to automatically reproduce each intermediate result.

Jupyter notebooks should be included that document every result with tangible textual explanations.

A **spreadsheet** serves as a table of contents. For every data set that you produce or process, the spreadsheet will indicate, at a minimum

- date;
- corresponding record in lab notebook;
- full path to data directory;
- link to Jupyter notebook / master script;
- Random seeds used to initialize RNGs;
- Git hashes/version numbers of software used;
- DNS name of computer the data was generated/processed on.

You may include other information that is useful for the project.

Avoid manual data manipulation and try to automate everything! Ideally, the whole process of generating the data (running simulations), processing the results, and creating graphical representations can be replicated by executing one or two driver scripts.

Every publication from our group will be accompanied by a reproducibility package that includes the data sets and all the code that is needed to reproduce the tables and figures of the paper.

Software Sustainability

While pursuing computational research, you will be changing your models, workflows, and programs/scripts frequently. Any such change can have intended or unintended impact on the results, and failing to record every change to the software and all input parameters will make it difficult if not impossible to replicate your results. Version control systems allow you to track the continuous evolution of your software.

Every custom file that is required to reproduce a result has to be under version control. This includes but is not limited to

- Source code (C/C++, FORTRAN, etc.)
- Input files
- Batch scripts (Bash, PBS, etc.)
- Analysis scripts (Python, R, Matlab, etc.)

Automatically generated files and binary files should not be checked into the repository. In some cases, exceptions from these rules may be warranted. These should be discussed with all team members.

The version control system generates a unique identifier (Hash) at every state of the code. Every result has to be associated with the identifier of the version used to produce it. This identifier has to be recorded in your lab notebook.

For collaboration and working simultaneously on repositories, we will be following Git(hub) Flow https://guides.github.com/introduction/flow/index.html. The principle guidelines are

- the master branch must always be in a working state;
- development takes place on descriptively named branches;
- commit to that branch locally and regularly push to the upstream repository;
- when the branch is ready for merging, open a pull request;
- branches will only be merged after code review.

Tools

As a research group, we will share data and documents on a regular basis, so we need to have uniformity. If no project specific arrangements have been made, we will use the following standards for research group projects.

Task	Standards
Project management	Wrike, Basecamp, Asana
Bibliography database	BibTeX, Mendeley, Zotero
Document storage / File sharing	Box
Quantitative data storage	*define per project*
Quantitative data analysis	Python (Jupyter notebook)
Version control system	Git (GitHub, GitLab, Bitbucket)
Abstracts, reports, papers, proposals	LaTeX (Overleaf, ShareLaTeX, Authorea)
Presentations / Slides	LaTeX Beamer, Google Slides, Prezi
Dissertation	LaTeX
Internal reports	*share on project page*
Day to day communications	Slack, Email

Acknowledgments

I am indebted to Eliza Gallagher, Department of Engineering and Science Education, for sharing her GRA mentoring and supervision files. Parts of this document are closely adopted from her guidelines. The Computational Science Manifesto is inspired by the Science Code Manifesto [5] and Lorena A. Barba's Reproducibility PI Manifesto [4].

References

1. Sandve, G. K., Nekrutenko, A., Taylor, J. & Hovig, E. Ten Simple Rules for Reproducible Computational Research. *PLOS Comput. Biol.* **9,** e1003285 (2013).
2. Noble, W. S. A Quick Guide to Organizing Computational Biology Projects. *PLOS Comput. Biol.* **5,** e1000424 (2009).
3. Schnell, S. Ten Simple Rules for a Computational Biologist's Laboratory Notebook. *PLOS Comput. Biol.* **11,** e1004385 (2015).
4. Barba, L. A. Reproducibility PI Manifesto. 10.6084/m9.figshare.104539 (2012). Presentation for a talk given at the ICERM workshop "Reproducibility in Computational and Experimental Mathematics". Published on figshare under CC-BY.
5. Barnes, N. Science Code Manifesto. Climate Code Foundation (2011).