Course Proposal: Analytical Workflows

(temporarily IB 599)

Mark Novak & Ben Dalziel

Course Motivation

Graduate students are amassing ever larger datasets to ask ever more challenging questions with ever more sophisticated statistical techniques. Many propose dissertation chapters entailing mathematical models and computational simulations. To keep pace requires students to efficiently and effectively straddle empirical, statistical, and theoretical approaches. These large-scale, multi-faceted, and ever more collaborative approaches lie at the heart of our department's *integrative* approach to biology.

Planning and constructing reproducible "*analytical workflows*' is crucial for reproducible science and in many of the career paths our students pursue. These skills are non-trivial, and distinct from learning specific statistical techniques or algorithms. However, our IB graduate curriculum does not currently provide the opportunity to learn these skills [1].

Our course will provide students the tools and experience they need to organize and manage their analyses from inception to publication. Working with their own data and/or models, students will learn and practice: (re)organizing projects into efficient, reproducible, and easily-modified "*analytical workflows*"; writing modular, transparent computer code for biology (in the programming language of their choice); tracking and communicating progress and hurdles (issues) to others using stand-alone and cloud-based version control and collaboration software.

Student-targeted Advertisement

Have you proposed a modeling chapter for your dissertation but need support getting things up and running? Are you sitting on a data set ready for analysis and visualization but don't know how or where to begin? Maybe you're far along in some series of analyses and feel "lost in the trees."

This course will help you with these challenges by practicing the development and implementation of efficient, reproducible *workflows* for your projects. Every project should (and can) be modular and fully automated, hence reproducible, portable and easily modified. Rerunning a model under a different set of parameters should (and can) be as simple as a few keystrokes. Regenerating all analyses, figures and tables after finding a typo in your code or dataset should (and can) be painless.

Efficient workflows start at project conception and end only if the project idea is itself a dead end. Thus, in this course, we'll work to practice (1) refining and articulating project ideas and goals, (2) creating modular and automated analyses, and (3) using best-practices in coding and project management. We'll learn how to use Git, GitHub, LaTeX and Markdown. The instructors will mostly use R within RStudio, but users of other programming languages and text editors are welcome and encouraged. You will need either (1) a large unwieldy dataset and an end goal (e.g., reproducing someone else's analysis or visualization) or (2) a dynamical model or simulation (or sufficiently well-developed ideas for one). The use of other people's data or published models is also encouraged, as needed.

Course Catalog Description

Theory and implementation of efficient, reproducible workflows – including best-practices in scientific programming, project management, and collaboration – for computational, analytical, and data-driven biological research.

Student Testimonials

A preliminary IB599 version of this course was taught by Ben Dalziel (with support from Mark Novak) in the Spring of 2019. Student eSET scores for this course were 6.0 for "the course as a whole" (versus the departmental median of 4.9) and 6.0 for 'the instructor's contribution" (versus the departmental median

[1] See *Related Courses at OSU* below.

of 5.1) out of 6.0 points possible. The following are excerpts of the feedback that was received via email and the eSET evaluations.

> *"I am of the opinion that this Analytical Workflow class is indispensable. Engaging in research comes with a massive deluge of papers, files, data, output, etc, and yet prior to this class I had never before been exposed to organizational "best practices" recommendations."*
>
> *"After taking the class, [...] I feel empowered to solve issues with my model on my own."*
>
> *"I'm so happy this class exists, and it was instrumental to much of the progress I've made this term."*
>
> *"Ben and Mark helped turn a daunting task that I'd been putting off (my modeling chapter) into a well-organized reality! Five stars."*
>
> *"This class has been extremely useful for the conceptual organization and technical execution of my research. Thank you for organizing and leading this class!"*
>
> *"I really appreciate that you shared your experience and learning process with us (metacognitive reflection! best teaching practices!)."*
>
> *"I wasn't experienced in R/Git or savvy about best practices in coding or reproducible workflows. Ben and Mark helped me structure my independent work and provided essential information about programs and best practices in coding."*

Details

Credits:	4
Frequency:	Annually (starting ~~2021~~ 2020)
Quarter:	~~Winter~~ Fall
Course times:	Tuesday & Thursday 10:00-11:50am
Instructors:	Mark Novak & Ben Dalziel (of record alternating annually)
Prerequisites:	Graduate standing, or by instructor permission
Location:	TBD (likely "remote")

Learning Outcomes

After successful completion of this course, students will be able to:

1. Translate a research plan into an explicit analytical workflow;
2. Apply best practices in scientific programming to construct reproducible research;
3. Manage and collaborate on complex research projects using a version control system;
4. Apply analytical workflows to advance their dissertation research.

Schedule

Week 1: Course organization & Philosophy of workflows

Week 2: Student workflow diagrams & Project presentations

Week 3: Project setup & Version control (Git)

Week 4: Team troubleshooting

Week 5: Collaborative projects, management & debugging (GitHub)

Week 6: Coding best practices & grammar

Week 7: Visualization theory & workflows

Week 8: Writing notes, reports & papers (LaTeX, Markdown & reference management)

Week 9: Team troubleshooting

Week 10: Project presentations

Related courses at OSU

OSU's existing *statistics* courses focus exclusivively on data visualization. Existing *omics* courses introduce students to specific *analysis pipelines* in particular programming environments, but do not address issues relating to project management as a whole. Existing courses are also not relevant to the many IB students pursuing non-*omic* research.

ST 537 - Data Visualization *E-campus only*

"Perceptual principles for displaying data; critique and improvement of data visualizations; use of color in visualization; principles of tidy data; strategies for data exploration; select special topics. *Prerequisites: ST 512 or ST 517 or ST 552*"

BB 485/585 - Applied Bioinformatics

Prerequisites: BB 314 or BB 314H" "Fundamental concepts needed to understand the software and methods used in bioinformatics. Includes contemporary techniques such as databases, gene and genome annotations, functional annotations, sequence alignment, motif finding, secondary structure prediction, phylogenetic tree construction, high-throughput sequence data, ChIP-Seq peak identification, transcriptome profiling by RNA-Seq, microRNA discovery and target prediction.