# *Course Proposal:* Analytical Workflows (IB 5XX)

Mark Novak & Ben Dalziel

## Course Motivation

Graduate students are amassing ever larger datasets to ask ever more challenging questions with ever more sophisticated analytical techniques. To keep pace requires students to straddle empirical, statistical, and theoretical approaches. Many propose dissertation chapters entailing mathematical models and computational simulations. Many others implicitly work with computational models by using complex datasets which are derived from a mixture of high-throughput automatic observation and statistical modeling, such as climate datasets. These large-scale, multi-faceted, and ever more collaborative approaches lie at the heart of our department's *integrative* approach to biology.

And yet, nowhere in our curriculum are students provided the basic tools and best practices with which to manage this complexity*. Moreover, these basic tools are standard practice in many of the jobs in which our students aim to work.

Our course will provide students the tools they need for reproducible research with modern data. 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.

> *Existing *omics* courses introduce students to specific *analysis pipelines* in particular programming environments, but these courses do not prepare students to contrast and synthesize pipelines into *workflows*, which is required in order to perform reproducible research with complex data. Existing courses are also not of relevant to the many IB students pursuing non-*omic* research.

## 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 giant 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 class 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 class, 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.

## 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 (vs. the departmental mean of 4.9)

and 6.0 (vs. the departmental mean of 5.1) out of 6.0 points possible for "the course as a whole" and "the instructor's contribution" respectively. The following are excerpts of the feedback that was received via email and the eSET evaluations.

> *"Ben and Mark helped turn a daunting task that I'd been putting off (my modeling chapter) into a well-organized reality! Five stars."*
>
> *"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."*
>
> *"After taking the class, [...] I feel empowered to solve issues with my model on my own."*
>
> *"I really appreciate that you shared your experience and learning process with us (metacognitive reflection! best teaching practices!)."*
>
> *"I'm so happy this class exists, and it was instrumental to much of the progress I've made this term."*
>
> *"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."*
>
> *"This class has been extremely useful for the conceptual organization and technical execution of my research. Thank you for organizing and leading this class!"*

## Details

| | |
|---|---|
| *Credits:* | 4 |
| *Quarter:* | Winter |
| *Course times:* | Tuesday & Thursday 10:00-11:50am |
| *Frequency:* | Annually (starting 2021) |
| *Instructors:* | Mark Novak & Ben Dalziel (of record alternating annually) |
| *Prerequisites*: | Graduate standing, or by instructor permission |

## Course Catalog Description

Theory and implementation of efficient, reproducible workflows – including version control using Git, collaborative issue tracking using GitHub, and research dissemination using LaTeX – for the management of computational, analytical, and data visualization projects.

## 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