# Final Projects
in BILD 62

Goals for this video

- Describe scope and expectations for final projects
- Introduce means of code collaboration
- **Discussion**: Form your group for the project!
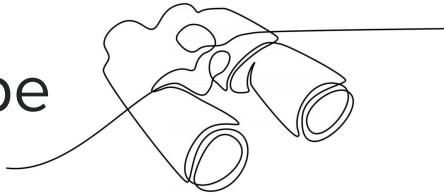
The main goal of the project is to demonstrate that you can design, write, and implement code to fulfill some task that you choose.
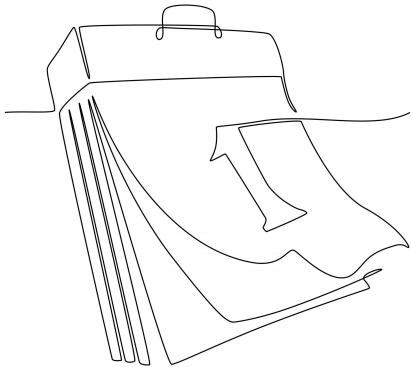
Objectives for the final project

- Choose a topic that you are interested and work on code related to that idea

- Plan out what is required for a computational solution to your chosen topic

- Practice following best practices for coding style, documentation and code testing

- Work with different coding tools and packages

- Collaborate **with a group**

Project Scope

- **There is not a minimum amount of code that you have to write for the project.**
- **Your project must implement some new thing, that you design and write the code for.** To do so, you are expected to write new code that creates or adds some functionality.
- A project that appropriately responds to this call will have organized and documented code that:
  - Includes at least **three new functions (or methods)**
  - Uses code constructs such as loops and conditionals when needed
  - Imports code from available modules when needed

Final project schedule

Week 7

- ❏ Understand expectations for projects
- ❏ *In Discussion:* Form groups
- ❏ Start considering what you'd like to do & working on your proposal

Project topics

For your project, you can choose a topic that extends an application from one of the assignments or in class notebooks, or propose your own topic.

Topics largely fall into two categories:

1. **Write a program to complete a task.** For example:
   ○ Create a chatbot using the framework from the "guessing game" challenge.
   ○ Write a program that will generate plots for various datasets of the same format
   ○ Look for specific gene alleles in complex DNA sequences
2. **Analyze & visualize data of your choosing**
   ○ Growing list of datasets

Sample projects (from BILD 62)

https://github.com/BILD62/StudentProjects

Sample projects (from COGS18)
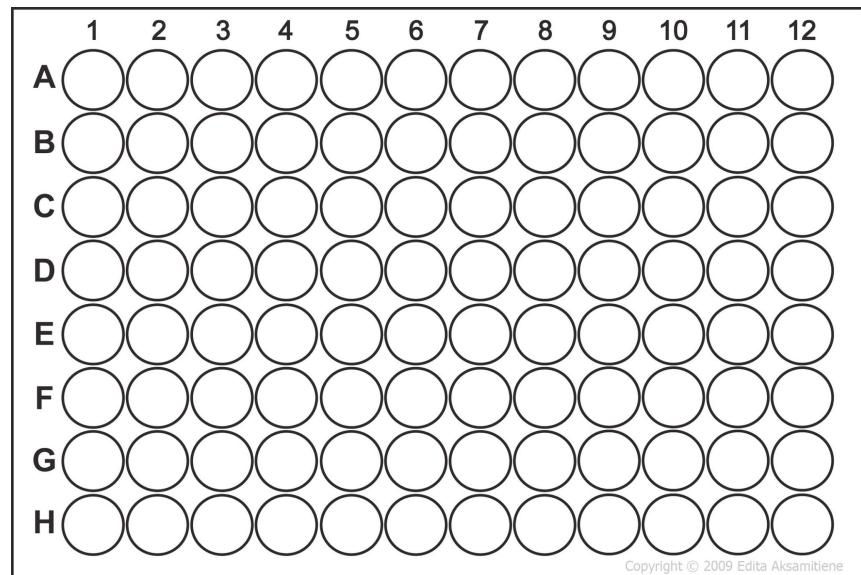
Data Analysis:
https://github.com/bydanielasodre/Project_COGS18_WI21

Biological Data Pipeline:

https://github.com/n1tsai/Lab-Assays

Study flashcards:
https://github.com/vikizzz/Flashcard/blob/main/ProjectNotebook.ipynb

Creating a group on Canvas
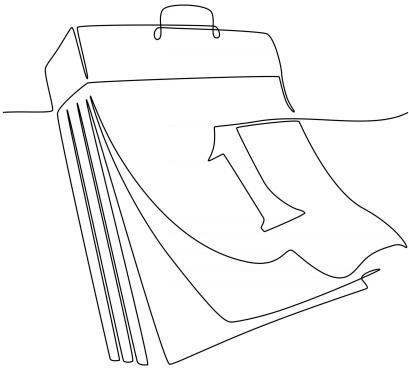
Once you are assigned a group in Discussion, we will create your group on Canvas.

If you do not have a group by the end of Week 7, you'll automatically be assigned one.

Final project schedule

Week 7

- ❏ Understand expectations for projects
- ❏ *In Discussion:* Form groups
- ❏ Start considering what you'd like to do & working on your proposal

Week 8

- ❏ Write your proposal & submit *(due end of Week 8)*

Project Proposals *(one per group)*

In *no more than one page*, describe your main goal and approach that you will take in your final project.

Your proposal should have the following elements:

I. Group member names

II. Goal of your project: One sentence that describes the goal of your program OR the data analysis question you'll address.

III. Background: In one paragraph, describe the motivation behind your project.

IV. Approach: How will you design your project? What is the collaboration plan? In other words, what tasks will each person handle?

BILD 62: Final Project Proposal

Goal of the project:

The goal of this project is for the user to guess a word based on a given DNA strand.

Background:

As biology majors, we are very familiar with the central dogma. DNA is transcribed into RNA which is then translated into an amino acid sequence. Given our familiarity with this pathway from our coursework, we wanted to take these principles and design a game that would utilize these principles in a novel way that could be fun for users. We both are huge fans of word games, especially the world renowned "Wordle", so we wanted our game to integrate a word guessing challenge.

> **Doesn't *need* references but they might be useful**

Approach:

This will be done via a series of steps to first allow the DNA to get transcribed into RNA. Based on the given codons within the RNA strand that is obtained, the single letter abbreviations that correspond with each codon will be spit out using code. The user will then use the letters from the codon sequence and unscramble the letters into a word where they will have to enter their final guess into a question, if correct, the program will spit out a "prize", if not the user will be prompted to guess again. The end goal is for the user to figure out the correct word. Our design for this project will revolve around three different functions. These three functions are the following:

> **Could be a list of steps or a paragraph**

- Function 1: Transcribe a DNA strand into its complementary RNA strand.
- Function 2: Will take in the RNA strand and give single letter amino acid abbreviations
- Function 3: Examine the entered word and see if it matches the desired word. If not it will prompt the user to guess again. If the word matches, the program will output a "congratulations" message to the user.

Clear function descriptions

II. Goal of your project: The goal of the program is to analyze raw data representing protein expression over time and infer translation elongation times of those genes of interest.

III. Background: Translation elongation has especially been found to be related to translation efficiency and mRNA localization to the mitochondria, the organelle that is significant in energy generation for the cell. It has been found that there is an association between translation elongation rate and the localization behavior of the nuclear-encoded mRNA to the mitochondria. This leads to investigating the translation elongation rates of mitochondrial mRNAs to see if there is a set that translates slower or faster. An in-vivo elongation reporter, luciferase, will be used to govern the initiation of transcription of translation of the gene of interest. The data will track luciferase expression for an hour, starting from the initiation of transcription. The goal is to quantify the translation elongation rate of these nuclear-encoded mitochondrial mRNAs.

IV. Approach: The luciferase induction assays were used to track the dynamics of induction which can be used to infer the translational elongation rates of the genes of interest. The expression of nLuc over 60 minutes then produces a curve, which represents the protein expression of the tested gene. We know that protein expression is proportional to mRNA amounts and time, and mRNA amounts are proportional to DNA amounts and time. Knowing that DNA amounts are constant, nLuc expression can then be proportional to time[2]. We can then take the square root of nLuc expression to produce a Schleif plot, which displays a nice linearization to further analyze the data (Schleif et al., 1973). The x-intercept of a line of best fit from a data selection of a linear portion of the Schleif plot represents the time it takes for the first protein to be produced. Taking the difference between the x-intercepts of the control, which does not contain a tested gene, and the gene of interest, the elongation time of the gene of interest can be determined.

In my comments to this group, I noted that this sounds like a function

To determine the transcription elongation rate more efficiently, we can introduce coding to perform all the calculations based on the logic explained above. The data set we have contains five different columns, including time (in sec), translated protein products with/without tetracycline (ATC) in control group (without the tested gene), and translated protein products with/without ATC in HEM1 gene (a mitochondrial DNA that regulates transcription). We will use the HEM1 gene as a template to train the script to learn how to calculate the rate of transcription elongation and the difference between the rate of transcription elongations of various genes. Based on the data from the luciferase induction assays, the protein expression level is shown across time, which can be plotted as a function of protein expression with the tested gene using the matplotlib.pyplot module. Then, we will use the numpy package to create the Schleif plot.

We will be working on the CoLab platform to do all the codings in a cooperative manner. ▮▮▮▮▮▮ will provide dataset background explanation, code plots, and determine the portion of data used for linear regression. ▮▮▮▮▮▮ will reorganize data with pandas and predict the transcription elongation rate with linear regression and intercept determination. ▮▮▮▮▮▮ will perform dataset calculations including the Schleif plot and perform application of the code to luciferase induction assays data of other genes.

Really clear group member roles!
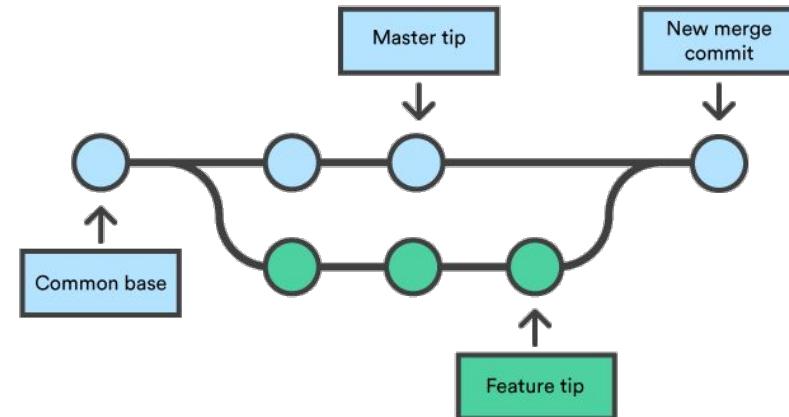
Options for collaboration

- **DataHub Container** — automatically set up once group is on Canvas
- **Google Colab** (https://research.google.com/colaboratory/)
  - Notebook can be shared with multiple people and viewed simultaneously
  - *However* multiple people cannot edit simultaneously (unlike Google docs)
- **Git & Github**
  - Github is a cloud storage service for the files that you maintain with Git.
  - Professional way to share code & implement version control!
  - First step: set up a GitHub account and *one repository for your project*

Version Control

- **Version control** is conceptually similar to track changes in Word — it keeps track of who makes changes, and what they did.
- A really common way to do this with code is a program called **git**.
  - GitHub is an online code hosting service that uses git and has a friendly user interface and a couple other features
- In theory, git can be used to track any type of text.
- Git can be used locally and/or with a remote repository of code

Master tip

New merge commit

Common base

Feature tip

Using git

git helps you collaborate on code & track changes

In other words, it implements **version control**.
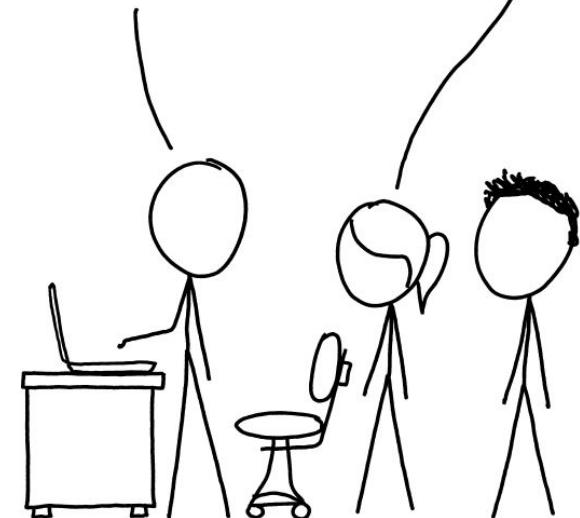
More information: [Git/Github Introduction - VOYTEKlab](https://...)

Steps for creating your project repository

1. Make an account on GitHub
2. **Team Member #1** creates a new **public** repository
3. **Team Member #1** adds initial code to the repository (by uploading on Github)
4. **Other team members** **clone** repository via terminal in DataHub
5. **Other team members** **add, commit, and push** their code (either as separate files, when that makes sense, or to scripts/notebooks)
6. **Team Member #1** **pulls** changes
7. BEFORE ANYONE EDITS CODE use **git pull** to get changes

Typical git workflow

1. **clone** code from a remote source to your local computer
   (or in our case, to your server space)

terminal prompt name

```
ajuavine@dsmlp-jupyter-ajuavine:~$ git clone http://www.github.com/ajuavinett/CellTypesLesson.git
```
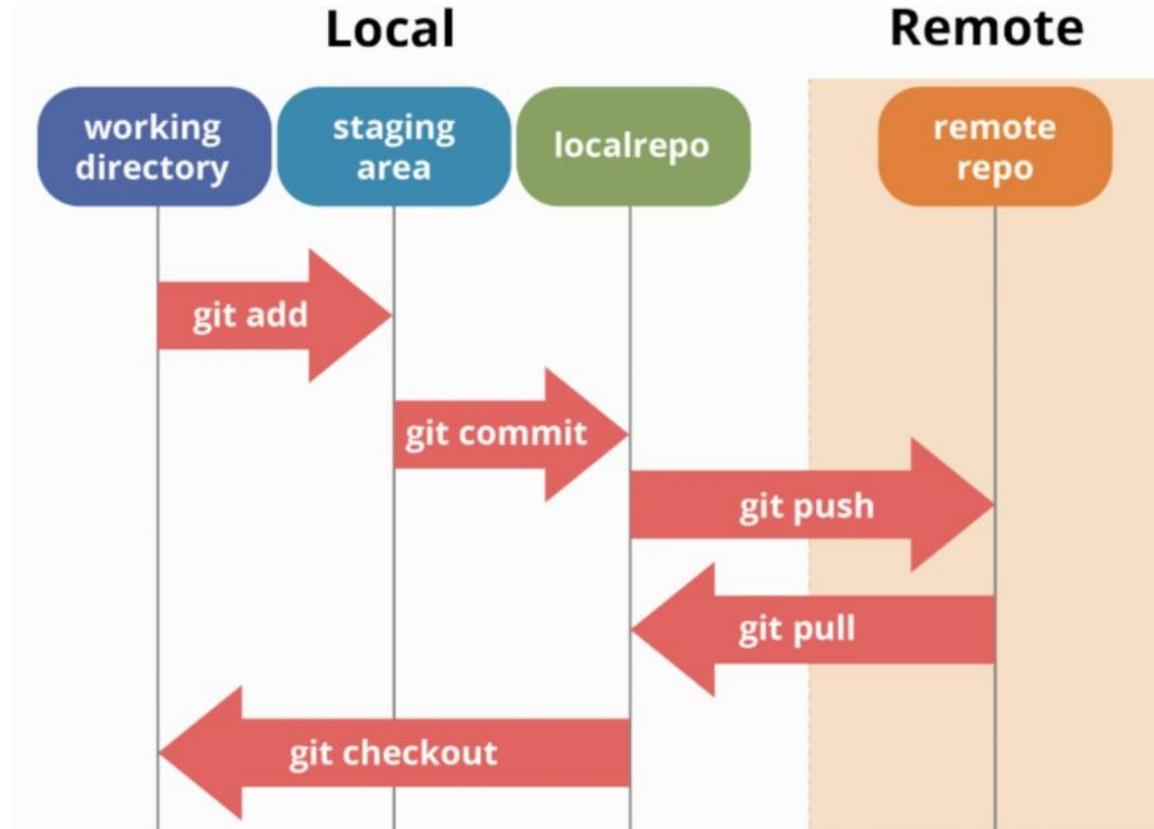
2. Edit code locally on your machine (**add** & **commit**)
3. **push** code back to a remote source

```
git status
```
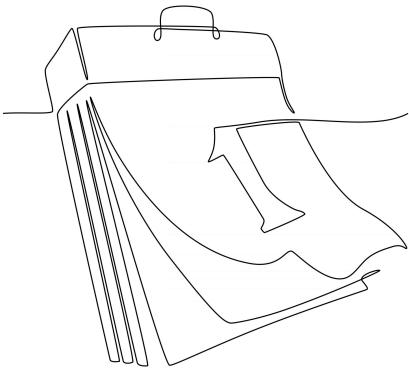
Tells you whether you're up to date

```
git checkout <filename>
```

Discard your changes (only use this if necessary!)

Git flow for a simple project (from [here](#))

Final project schedule

Week 8

❏ Write your proposal & submit *(due end of Week 8)*

Week 9

❏ Review your feedback & start writing your code

Week 10

❏ Write your code!

Finals week

❏ Turn in your final code