BIO47 Database and App Instructions

Contents

How to install the database and app

Step 1: Download and extract BIO47_Data.zip

- Download the BIO47_Data.zip file and save it in a convenient location on your computer. Do not save to your Downloads folder or any other folder with automatic syncing (such as Google Drive or Dropbox).
- Extract the contents of the zipped folder so that you have a folder named BIO47_Data that contains three subfolders (code, data, output) as well as three files: BIO47 Database and App Instructions.pdf, BIO47_Data.Rproj, and Table Descriptions.csv.
- The database (but not the app) is now installed. You can open and view any of the data tables using Excel or any program that can open and display csv files. Read the sections Database Structure and Main Table Descriptions to learn more about the files that are in the BIO47_Data folder. In order to complete the BIO47 Prelab assignments and use the make_analysis_table app you must install R and a few requires R packages. See steps 2 - 5 below.

Step 2: Install or update R and RStudio

RStudio is a user interface program that allows you to use the program R. You must have current versions of both programs installed in order to use the app. Both programs are free.

If you do not have R or R studio on your computer…

1. Download and install the correct version of R for your operating systems from https://cloud.r-project.org/. Use the links for installing R for the first time.
2. Download and install the most recent version of RStudio Desktop for your operating system from https://www.rstudio.com/products/rstudio/download/#download.

If you have R and/or RStudio…

1. Update your R version to the most recent version. If you do not care about losing your existing installed packages, the easiest way to do this is to download and install the most recent version of R from https://cloud.r-project.org/. If you want to keep your installed packages and update them, a quick Google search for “How to update R and R packages” should give you instructions. For example, the post https://www.datasciencecriot.com/r/upgrade-R-packages/ has instructions for Windows and Mac.

2. Download and install the most recent version of RStudio Desktop for your operating system from https://www.rstudio.com/products/rstudio/download/#download.

If you do not update to the most recent versions of R and RStudio, you may not be able to use the app because it depends on recent versions of the tidyr and dplyr packages (see next section).

Step 3: Install or update required R packages

The make_data_table app requires recent versions of the following packages:

- dplyr (version 0.7.4 or higher)
- tidyr (version 0.7.1 or higher)
- shiny (version 1.0.5 or higher)

If you just installed R and RStudio you do not have these packages installed yet. Skip ahead to the numbered instructions.

If you are an experienced R user with a recent version of R, check whether you have recent enough versions of these packages and if not, install or update them. Note that you may also need to update other packages that shiny, dplyr and tidyr depend on.

Instructions for installing required packages

1. Open R Studio.
2. Click Tools > Install Packages…
3. In the window that appears, select the following options:
   - Install from: Repository (CRAN)
   - Packages: shiny, dplyr, tidyr
   - Install to Library: [Default]
   - Install dependencies is checked
4. A pop-up window may ask, “Do you want to install from sources the packages that need compilation?” Select Yes.
5. The installation process may take several minutes. You will know that it is finished when in the upper right Console window you see > followed by a blinking cursor.
6. Check that the three packages installed correctly. To do this click on the tab in the lower right window labeled packages and scroll through the list of installed packages until you see the three packages you just tried to install (shiny, dplyr, tidyr). Check that the version number to the right of each package is greater than the version numbers listed at the top of this “Step 3” section. If one of these packages are not listed or the version number is too low, see Step 4: Troubleshooting. Otherwise, skip to Step 5: Run the app.

Step 4: Troubleshooting

In this section we describe a few issues that users have found when trying to install the three required packages (shiny, dplyr, tidyr). You may come across an issue not listed here, in which case we recommend copying the error message you see in the R Console into a search engine along with “error installing package X on Y”, where X is the name of the package that did not install correctly and Y is the name of your operating system.

Some things to try:

Determine which packages failed to install correctly. Next attempt to re-install the failed packages one at a time by clicking Tools > Install packages… and then only typing the name of one package. Likely this will still not work, but it will allow you to re-generate the errors in the console and read them. Scroll back through the text that appeared in the Console window until you come to a line that reads:

```
> install.packages( "shiny")
```

Then scroll down until you see the first "ERROR:…" This is the first problem you need to fix.

If the error says something along the lines of "Can't update/install XX package", where XX is the name of a package (we've had trouble with sourcetools and htmltools packages), then try typing in the Console window: `install.packages("XX", type = "source")` , replacing XX with the name of the package that did not install. If that doesn't work, try an internet search as described above.

If you are having trouble with shiny and you are working on a Mac, you may need to download or update Xcode. Go to the App Store on your computer and check whether you have Xcode installed (update it!) and if you don't then install it. Then, open the application and make sure that you agree to the "Terms and Conditions" or it will not work correctly. Once XCode is installed, try installing shiny again in R Studio.

Step 5: Run the app

Go to the section How to use the make_analysis_table app and following the instructions for Step 1. If the pop-up window appears, then the app should be working.

Database structure

The BIO47 Database is stored and distributed in a folder named BIO47_Data. This folder contains database documentation (BIO47 Database and App Instructions.pdf, Table Descriptions.csv), an R Project file to allow easy access to files in the database (BIO47_Data.Rproj) and three main folders:

code: a folder with R scripts used by the current BIO47 class and an application for creating summary tables from the database (make_analysis_table)

data: a folder with the main database tables and any derived table created by the make_analysis_table application or distributed to the current BIO47 class (e.g. files for completing assignments). The tables in the main_analysis folder are the primary database and should not be changed by anyone except the database maintainer. Users can place any files they want into the analysis_tables folder.

output: a folder where any figures or other content generated in R should be saved.

For detailed information about how variables were collected and the database is constructed, read the section Main Table Descriptions. For a quick overview of the meanings of column names in each table, read the section Description of Database Columns.

Do not change any of the files that are in this database. Instead, save your own tables for analysis in the data/analysis_tables folder and save any other content in the output folder.

For more information

Database Maintainer: Jesse Miller, jedmill@stanford.edu
Database Creator: Jes Coyle, jes.r.coyle@gmail.com

How to use the database

Start by opening the BIO47_Data.Rproj file (double click on it), which will open a new R Studio session. If you are in R Studio already, click File > Open Project… and then navigate to the BIO47_Data.Rproj file. When the BIO47_Data R Studio project is open, the working directory is automatically set to be the main BIO47_Data folder.

IMPORTANT: If you do not first open the BIO47_Data.Rproj file, the following code and app will not work. This file tells R where the "working directory" is so that it can find the data folder.

You can either read in the main data tables directly to R using:

```
table_name <- read.csv(file.path("data/main_tables", "table_name.csv"))
```

where table_name is one of the tables listed below in the Description of Database Columns section. Or, to create a custom data table for analysis, use the **make_analysis_table** app. This will allow you to aggregate data from multiple years and select only those variables of interest. Read the section How to use the make_analysis_table app for step-by-step instructions.

If you have saved a data table as analysis_table.csv inside the data/analysis_tables folder, you can read the table into R using:

```
analysis_data <- read.csv(file.path("data/analysis_tables", "analysis_table.csv"))
```

Save any figures that you generate to the output folder.

How to use the make_analysis_table app

Step 1: Open the application

1. Open the BIO47_Data R Studio project by double-clicking on the BIO47_Data.Rproj file in the main BIO47_Data folder. This step is critical. You must have the BIO47_Data R Studio project open or the app will not work.

2. When R Studio opens, navigate to the make_analysis_table folder inside the code folder (code/make_analysis_table) and open the file named app.R. ***Never make changes to the app.R file.***

3. When the code for the application opens in R Studio, click the Run App button with the green arrow in the upper right corner of the code window. This should open a pop-up dialog with many options for choosing variables in the database.

Step 2: Select columns for the data table

1. **Choose years:** Select the years that you would like data from first, since this will change the variables that are available. Not all variables were measured in all years.

2. **Summarize by plant or flower:** If you want each row in the table to be a plant, then summarize data by plant. This will aggregate flower variables measured on the same plant. Alternatively, if you want each row in the table to be a flower collected for nectar analysis, then summarize the data by flower.

3. **Choose plant-level variables:** Plant-level variables are properties that were measured on each plant, such as location, plant size, dates when flowers were first observed and weekly flower counts. I you summarize by flower, but include plant-level variables in the table, flowers from the same plant will have the same value for these variables. It is probably more appropriate to conduct a plant-level analysis if you are interested in comparing flower-level variables to plant-level variables.

    - **Flower count variables**, such as flower density and the proportion of flowers with closed stigmas, were measured multiple times on the same plant each year so you will need to select the weeks from which you want data. Flower counts for a week are averaged across all of the teams that collected data that week. You may also choose to summarize flower count variables in April and May by calculating the average or maximum of these flower count variables across weeks.

4. **Choose flower-level variables:** Flower-level variables are properties that were measured on each flower collected for nectar microbe analysis. In most years, flowers were experimentally covered with a cage, a mesh bag, or left exposed. If you choose to include flower-level variables you must specify which treatments you would like data

from (Caged, Bagged, Exposed). How these data are included in the table depends on whether you summarize by plant or by flower.

- **If summarizing by plant,** you can calculate the mean and/or maximum of flower-level variables for each plant. You should specify whether you want separate summaries for each treatment (select "no" for Combine variables across treatments?) or whether the data from multiple treatments within a plant should be combined (select "no" for Combine variables across treatments?).

  Variables that are TRUE/FALSE: If you choose variables that are TRUE/FALSE (e.g. stigma status or whether the nectar pH is < 5), then summarizing using the mean will give the proportion of flowers on a plant within each treatment that are TRUE for that variable (e.g. the proportion of exposed/bagged/caged flowers with closed stigmas on a plant). Summarizing using the maximum will indicate whether any flowers have a TRUE status for that variable (e.g. whether any exposed/bagged/caged flowers on a plant had closed stigmas). For example, a column named "Stigma_mean_Bagged" gives the proportion of bagged flowers that had closed stigmas whereas a column named "Stigma_max_Bagged" will indicate (0 or 1) whether any bagged flowers had closed stigmas.

- **If summarizing by flower,** the table will only contain flowers from the experimental treatments you select (Caged, Bagged, Exposed). If you select multiple treatments, be sure to select the "Treatment" variable so that it is included as a column in the table otherwise you will not know which flowers come from which treatments.

5. **Choose microbial taxa:** These selectors allow you to examine the presence or abundance of specific microbial taxa among flowers or plants. Select the taxa of interest and then select how each taxon should be quantified: presence/absence or abundance (CFU per uL). If you choose to include microbial variables you must specify which flower treatments you would like data from (Caged, Bagged, Exposed). How these data are included in the table depends on whether you summarize by plant or by flower.

   - **If summarizing by plant,** you can calculate the mean, maximum or total of each microbial variable for each plant. Only flowers from the treatments that you select will be reported. You should specify whether you want separate calculations for flowers from each treatment (select "no" for Combine variables across treatments?) or whether the data from multiple treatments within a plant should be combined (select "no" for Combine variables across treatments?).

     Microbial presence / absence: Summarizing presence/absence using the mean will give the proportion of flowers on a plant have each of the selected taxa present. Calculating the total will give the number of flowers in which each taxon

is present. Calculating the maximum will indicate whether or not each taxon is present at all on the plant.

Microbial abundance (CFU per uL): This feature is currently not implemented. When available, summarizing abundance using the mean will give the average CFU per uL of each taxon among flowers on the plant. Calculating the maximum gives the maximum CFU per uL found on the plant for each taxon. Calculating the total adds all of the CFU together and doesn't really make much sense.

- **If summarizing by flower,** the table will only contain flowers from the experimental treatments you select (Caged, Bagged, Exposed). If you select multiple treatments, be sure to select the "Treatment" variable so that it is included as a column in the table otherwise you will not know which flowers come from which treatments.

6. **Environmental variables**:
    - **Light variables**: Photosynthetically active radiation (PAR) was estimated from canopy photographs taken in most years and is available for months from the preceding year. Select whichever month(s) is relevant for your analysis.

    - **Temperature variables**: If you select a temperature variable, then you must also select which time period you want daily temperature measurement summarized by. Weeks refer to the week of the spring quarter and differ slightly across years. Monthly temperature summaries are from the same dates every year. For example, a column named "Temp_daily_max_Week7" gives the average daily maximum temperature during the seventh week of the quarter.

Step 3: Save the data table

The application will save your data table to the analysis_data folder inside the data folder in the BIO47_Data folder (BIO47_Data/data/analysis_data).

1. **File name**: type a name for your table. Do not include a file suffix (like .csv). If you have multiple tables in the analysis_tables folder be sure that the name you choose is different from these tables. The application will save over (without warning!) any files with the same name.

2. **How to handle missing data**: Some of the data may be missing for the variables you selected. By default the application will keep all rows even if they have missing data. You can remove rows that have missing data in all variables ("Remove rows with no data.") or remove rows that have missing data for any variable ("Remove rows with any missing data."). Check the table preview below before saving to see how many rows you lose by excluding data with missing values.

3. **Data from different years**: If you have selected multiple years and are summarizing by plant, then you need to decide whether data from different years should be in different rows of the table or in different columns. If you select “rows”, then each row will represent a plant measured in a single year and the table will have a column indicating which year the measurements were taken. If you select “columns”, then each row will represent a plant and there will be no duplicate rows for the same plant. Each column will have a year appended to it (e.g. .2017) to indicate when the variable was measured.

When you are satisfied with you table, click “Save Table”. Nothing will happen, but if you look inside the data / analysis_tables folder you will see a csv file with the file name you specified.

Step 4: Close the application
When you have finished creating data tables, closed the application by clicking the X in the upper right corner of the window. Also close the app.R code file so that you don’t accidentally change it. ***Never make changes to the app.R file.***

To use your new data table, open a new R script (or the one that you were working on) and type:

```
analysis_data <- read.csv(file.path("data/analysis_tables", "filename.csv"))
```

This will read in a table that you named “filename” to an R dataframe named analysis_data.

Description of Database Columns

The following table lists the variables that are stored in the database tables.

Column_name	Description	Table
Flower_density	Mean number of flowers (Num_flowers column) per cubic meter (Volume_m3 column) on a plant in a given week.	flower_counts
Num_flowers	Mean number of flowers on a plant in a given week.	flower_counts
Fraction_closed_stigmas	Proportion of flowers on a plant that had closed stigmas in a given week (Num_closed_stigmas / Num_closed_stigmas + Num_open_stigmas).	flower_counts
Num_closed_stigmas	Mean number of flowers with closed stigmas counted on a plant in a given week.	flower_counts
Num_open_stigmas	Mean number of flowers with open stigmas counted on a plant in a given week.	flower_counts
Treatment	Experimental treatment applied to the flower.	flowers
Bagged	Is TRUE if a flower was enclosed in amesh bag and FALSE if not.	flowers
Caged	Is TRUE if a flower was enclosed in a cage and FALSE if not.	flowers
Stigma	Is CLOSED if the flowers stigma was closed and OPEN if the stigma was open.	flowers
Age_category	Age category of the flower (1 to 4).	flowers
Age_days	Number of days that the flower was open before harvest.	flowers
Nectar_pH	Flower nectar pH.	flowers
Nectar_pH_less5	Is TRUE if the nectar pH was less than 5.	flowers
Nectar_uL	Volumn of the nectar in uL.	flowers
EtOH_Percent	Nectar ethanol percentage.	flowers
Fructose_mg_per_mL	Nectar fructose concentration (mg per mL).	flowers
Glucose_mg_per_mL	Nectar glucose concentration (mg per mL).	flowers
Sucrose_mg_per_mL	Nectar sucrose concentration (mg per mL).	flowers
Bacterial_CFU_per_uL	Number of bacterial colony forming units (CFU) per uL of nectar.	flowers
Bacterial_morphotypes	Number of bacterial colony morphotypes observed on solid media plates.	flowers
Num_BOTU	Number of bacterial operational taxonomic units (OTU) based on sequencing data.	flowers

Fungal_CFU_per_uL	Number of yeast colony forming units (CFU) per uL of nectar.	flowers
Fungal_morphotypes	Number of yeast colony morphotypes observed on solid media plates.	flowers
Num_FOTU	Number of yeast operational taxonomic units (OTU) based on sequencing data.	flowers
Volume_m3	Plant volume in cubic meters (usually calculated as height x width x depth).	plant_measurements
Height_m	Height of the plant in meters.	plant_measurements
Jan_PAR_mol_m2_day	Mean photosynthetically active radiation received by the plant in January (moles per square meters per day).	plant_measurements
Feb_PAR_mol_m2_day	Mean photosynthetically active radiation received by the plant in February (moles per square meters per day).	plant_measurements
Mar_PAR_mol_m2_day	Mean photosynthetically active radiation received by the plant in March (moles per square meters per day).	plant_measurements
Apr_PAR_mol_m2_day	Mean photosynthetically active radiation received by the plant in April (moles per square meters per day).	plant_measurements
May_PAR_mol_m2_day	Mean photosynthetically active radiation received by the plant in May (moles per square meters per day).	plant_measurements
Jun_PAR_mol_m2_day	Mean photosynthetically active radiation received by the plant in June (moles per square meters per day).	plant_measurements
Jul_PAR_mol_m2_day	Mean photosynthetically active radiation received by the plant in July (moles per square meters per day).	plant_measurements
Aug_PAR_mol_m2_day	Mean photosynthetically active radiation received by the plant in August (moles per square meters per day).	plant_measurements
Sep_PAR_mol_m2_day	Mean photosynthetically active radiation received by the plant in September (moles per square meters per day).	plant_measurements
Oct_PAR_mol_m2_day	Mean photosynthetically active radiation received by the plant in October (moles per square meters per day).	plant_measurements
Nov_PAR_mol_m2_day	Mean photosynthetically active radiation received by the plant in November (moles per square meters per day).	plant_measurements
Dec_PAR_mol_m2_day	Mean photosynthetically active radiation received by the plant in December (moles per square meters per day).	plant_measurements

deltaC13	Ratio of C13 to C12 in plant leaf tissue (parts per thousand).	plant_measurements
Foliar_N_percent	Plant leaf nitrogen percentage.	plant_measurements
Foliar_C_percent	Plant leaf carbon percentage.	plant_measurements
Watered	Is TRUE if a plant was watered and FALSE if not.	plant_measurements
BOTU_Richness	Total number of bacterial operational taxonomic units (OTU) on a plant based on sequencing data.	plant_measurements
FOTU_Richness	Total number of byeastoperational taxonomic units (OTU) on a plant based on sequencing data.	plant_measurements
Week_flowers_present	First week that students observed flowers on a plant.	plant_measurements
Date_flowers_present	Date of the first day of the week that students first observed flowers on a plant.	plant_measurements
UTM_N	North UTM coordinate of the plant (zone 10).	plants
UTM_E	East UTM coordinate of the plant (zone 10).	plants
Num_years	Number of years a plant was observed.	plants
First_year	First year that a plant was observed.	plants
Last_year	Most recent year that a plant was observed.	plants
Temp_daily_max	Average daily maximum temperature over a given time period (Celcius).	temperature_summary_long
Temp_daily_mean	Average daily mean temperature over a given time period (Celcius).	temperature_summary_long
Temp_daily_min	Average daily minimum temperature over a given time period (Celcius).	temperature_summary_long
Temp_daily_range	Average daily temperature range over a given time period (Celcius).	temperature_summary_long

Main Table Descriptions

Table name: flower_counts.csv

Description: Each row of this table contains data on the open flowers present on a plant in a given year during a particular week of the quarter. Most plants were surveyed by multiple teams of students (on different days of the week) so flower counts and the fraction of flowers with closed stigmas are averaged across teams within a week. To estimate pollinator activity, students were instructed to randomly sample 12 flowers on each plant and determine whether stigmas were open or closed, indicated that a flower has likely been visited by a pollinator. Note that some students may not have followed these instructions or the plant may have had fewer than 12 open flowers on a given day. Therefore, an weekly average of the number of open/closed stigma flowers is likely meaningless (columns 6 and 7).

Column	Name	Description
1	Year	Year the plant was sampled.
2	Plant	Unique number identifying the plant. Corresponds to column in plants.csv.
3	Week	Week of the spring quarter when the flowers were counted.
4	Num_flowers	Number of open (stigma visible) flowers on the plant. Averaged across all teams surveying the plant in a particular week.
5	Fraction_closed_stigmas	Estimated fraction of flowers surveyed with closed stigmas. Averaged across all teams surveying the plant in a particular week.
6	Num_closed_stigmas	Number of flowers with closed stigmas out of a random sample of flowers. Averaged across all teams surveying the plant in a particular week.
7	Num_open_stigmas	Number of flowers with open stigmas out of a random sample of flowers. Averaged across all teams surveying the plant in a particular week.

Table name: flowers.csv

Description: Each row of this data table corresponds to a single flower collected from a plant at Jasper Ridge in a given year. Each flower was either enclosed in a mesh bag, enclosed in a caged to exclude hummingbirds or left exposed.

Nectar microbe communities were assessed by culturing for 7 days at 25 C. Nectar was extracted via capillary tube and pipetted into 0.85% NaCl solution and then dilutions of this nectar solution spread on agar plates. Different nectar dilutions were used in different years, so variables that are not corrected for dilution may not be comparable across years (i.e. number of colony morphotypes and number of OTUs).

2018: Original dilution of nectar in 135 uL saline solution. Yeast plated at the original dilution and bacteria plated at 1:10 dilution.

2017: Original dilution of nectar in 40 uL saline solution. Yeast plated at 1:10 dilution and bacteria plated at 1:100 dilution.

R2A media with cyclohexamide was used to selectively culture bacteria of the original nectar solution, whereas YM media with chloramphenicol was sued to selectively culture yeast. Bacterial and yeast cell densities were then estimated from colony counts on the agar plates. The original nectar solution was used for all nectar chemistry analyses.

Several different methods were used in different years to estimate microbial diversity. In 2012 and 2016 - 2018, students selected colonies from their plates, amplified and sequenced either the ITS region (yeast) or 16S region (bacteria) of the ribosomal RNA gene, then BLASTed resulting sequences against GenBank to determine unique OTUs. In 2014, nectar was sequenced using _____ and microbial OTU richness estimated using the Chao1 richness estimator. In 2013, _____.

Note that several variables were only measured in some years. Flower data are missing for 2015 because that years students collected nectar from a different plant species.

Column	Name	Description
1	Plant	Unique number identifying the plant. Corresponds to column in plants.csv.
2	Year	Year the plant was sampled.
3	Week	Week of the spring quarter when the flower was collected.
4	Flower	Number identifying a flower within a particular year.
5	Bagged	Is TRUE if the flower was enclosed in a mesh bag prior to opening. Is FALSE otherwise.

6	Caged	Is TRUE if the flower was enclosed in a mesh bag prior to opening. Is FALSE otherwise.
7	Stigma	Is OPEN or CLOSED depending on whether the stigma was observed as open or closed prior to harvest.
8	Age_category	Age of the flower at harvest based on the color and appearance of the anthers. 1 = both sets of anthers yellow 2 = one set of anthers orange 3 = both sets of anthers orange 4 = anthers dark and wilted
9	Age_days	Number of days after opening when the flower was harvested .
10	Nectar_pH	pH of the flower nectar after dilution in 40 uL of 0.85% NaCl. Measured using a pH strip.
11	Nectar_uL	Volume (uL) of nectar extracted from the flower. Measured via microcapillary tube.
12	EtOH_Percent	Ethanol percentage of the nectar.
13	Fructose_mg_per_mL	Fructose concentration of the nectar (mg / mL).
14	Glucose_mg_per_mL	Glucose concentration of the nectar (mg / mL).
15	Sucrose_mg_per_mL	Sucrose concentration of the nectar (mg / mL).
16	Bacterial_CFU_per_uL	Density of bacterial cells in the nectar (CFU / uL).
17	Bacterial_morphotypes	Number of different bacterial colony morphotypes observed on a plate. Not corrected for dilution.
18	Num_BOTU	Number of bacterial OTUs present. See details above for methods.
19	Fungal_CFU_per_uL	Density of yeast cells in the nectar (CFU / uL).
20	Fungal_morphotypes	Number of different yeast colony morphotypes observed on a plate. Obvious contaminants (molds) were excluded. Not corrected for dilution.
21	Num_FOTU	Number of yeast OTUs present. See details above for methods.

Table name: nectar_community.csv

Description: Each row in this table corresponds to an OTU observed in the nectar of a flower collected from a plant in a given year. OTU identities were determined by different methods in different years.

In 2012, 2016 - 2018, students BLASTed DNA sequences against GenBank and chose the top result to assign an OTU identity to a sequence.

In 2014,

In 2013,

Column	Name	Description
1	Plant	Unique number identifying the plant. Corresponds to column in plants.csv.
2	Year	Year the plant was sampled.
3	Flower	Number identifying a flower within a particular year. Corresponds to column in flowers.csv.
4	OTU_ID	Number identifying a unique OTU. Corresponds to column in otu_names.csv.
5	CFU_per_uL	Number of CFUs per uL of a given OTU counted for a given flower. Calculated by matching morphotypes to identificed DNA sequences. Data not currently available.

Table name: otu_names.csv

Description: This table gives the taxonomic identities of OTUs observed in the nectar of flowers. OTU_IDs are unique across years to enable a comparison of microbial community composition across years.

Column	Name	Description
1	OTU_ID	Unique code identifying an OTU. Matches code in nectar_community table.
2	Binomial_name	Latin binomial name identifying the taxon to which the OTU belongs. Usually uses the lowest known taxon to which the species belongs.
3	Group	Is YEAST or BACTERIA depending on whether the OTU is a yeast or bacteria.
4	Class	Class to which the OTU belongs (if known).
5	Order	Order to which the OTU belongs (if known).
6	Family	Family to which the OTU belongs (if known).
7	Genus	Genus to which the OTU belongs (if known).
8	Species	Species to which the OTU belongs (if known).

Table name: plant_measurements.csv

Description: Each row in this table corresponds to measurements of a particular plant in a particular year. Students measured plant size using measuring tapes during the first week of the quarter. In most years, staff collected fresh leaves from each plant within one month prior to the start of the spring quarter for nutrient and carbon isotope analysis.

Note: Plant measurement data are missing from 2013, but rows are included for these plants as a placeholder if the data are found.

Column	Name	Description
1	Plant	Unique number identifying the plant. Corresponds to column in plants.csv.
2	Year	Year the plant was measured.
3	Volume_m3	Volume of the plant in cubic meters. Calculated as the volume of a rectangular prism enclosing the entire plant based on height, width and depth measurements.
4	Height_m	Height of the plant in meters.
5-17	[Jan-Dec]_PAR_mol_m2_day	Estimated total photosynthetically active radiation (PAR) received by the plant in a month prior to the spring quarter. PAR was estimated from hemispherical digital photographs capturing canopy cover above the plant using the Gap Light Analyzer software. Units are moles per square meter per day. Data from 2018 are identical and are from photos taken in 2017. Data from 2016 and 2015 are identical and are from photos taken in 2015.
18	deltaC13	Ratio of C13 to C12 in plant leaf tissue (parts per thousand).
19	Foliar_N_percent	Percentage of leaf tissue comprised of nitrogen.
20	Foliar_C_percent	Percentage of leaf tissue comprised of carbon.
21	Watered	Is YES if a plant was watered during the spring quarter. Is NO otherwise.
22	BOTU_Richness	Total number of bacterial operational taxonomic units (OTU) on a plant based on sequencing data.
23	FOTU_Richness	Total number of yeast operational taxonomic units (OTU) on a plant based on sequencing data.

24	Week_flowers_present	First week that students observed flowers on a plant. These dates were taken from the original plants data for years 2016 - 2017 and calculated for years 2012 - 2015 from the flower_counts.csv table. Because flower counts usually did not start until week 4, flowers could have been observed earlier in 2016 - 2017 than in previous years. In a few cases, the week when flowers were first observed in the flower counts data was earlier than recorded in the plants data. When this occurred, the earliest week of first flowering was used.
25	Date_flowers_present	Date of the first day of the week that students first observed flowers on a plant.

Table name: plants.csv

Description: Each row of this table corresponds to a plant in the database. The table contains information about each plant's location (UTM coordinates) and that it was measured.

Column	Name	Description
1	Plant	Unique number identifying the plant.
2	First_year	First yeas that a plant was measured.
3	Last_year	Most recent year that a plant was measured.
4	Num_years	Number of years that a plant was measured.
5	UTM_N	North UTM coordinate of the plant (zone 10).
6	UTM_E	East UTM coordinate of the plant (zone 10).

Table name: temperature_summary_long.csv

Description: This table contains temperature data collected by iButtons located at each plant. iButtons recorded temperature in Celcius every 10-20 minutes and these data were then summarized into daily temperature statistics (mean, max, min, range). Each row in this table contains a daily temperature summary at plant averaged over a particular time period in a particular year. For example, if the Time_period is “Week3”, then the Temp_daily_max column gives the average maximum daily temperature during the third week of the quarter. See the week_dates.csv table for the specific calendar dates that corresponds to spring quarter weeks in each year.

Raw iButtons files were missing from 2014 and 2015. Data from these years are based on summary data provided.

Column	Name	Description
1	Year	Year that temperature was measured.
2	Plant	Unique number identifying the plant. Corresponds to column in plants.csv.
3	Time_period	Period of time over which daily temperature summaries were averaged. See week_dates.csv for the calendar dates of each spring quarter week each year. Monthly time periods correspond to the actual calendar month.
4	Temp_daily_max	Average daily maximum temperature (C) over the given time period in the given year.
5	Temp_daily_mean	Average daily mean temperature (C) over the given time period in the given year.
6	Temp_daily_min	Average daily minimum temperature (C) over the given time period in the given year.
7	Temp_daily_range	Average daily temperature range (C) over the given time period in the given year.

Table name: temperature_summary_wide.csv

Description: Data in this table are identical to those in temperature_summary_long.csv. The only difference is that each row corresponds to a plant in a particular year and each column is a daily temperature summary averaged over a given time period. For example, the column Week3_Temp_daily_max gives the average maximum daily temperature