

# BETYdb Documentation:

## Volume 2: Data Entry

New Citation	New Site	New Treatment	New Yield	New Trait
<b>Author</b> Heaton, Emily	<b>Site name</b> University of Illinois- South Farms / Crop Sci	<b>Name</b> Control	<b>Mean</b> 26.0	<b>Trait Variable</b> Vomax (umol CO2 m-2 s-1) - maximum rubisco carboxylation capacity
<b>Title</b> Meeting US biofuel goals with less l	<b>Elevation (m)</b> 222	<b>Control</b> True	<b>Date</b> 1 2 - February	<b>Mean</b> 10.5
<b>Journal</b> Global Change Biology	<b>City</b> Urbana	<b>New Management</b> <b>Date</b> 2001 May 15	<b>Site</b> 783: University of Illinois- Sou	<b>Stat</b> 1.78
<b>Doi</b> 10.1111/j.1365-2486.2008.01662.	<b>Lat</b> 40.06	<b>Management Type</b> planting (plants / m2)	<b>Date</b> 2009 7 23	<b>Statname</b> SE
	<b>Lon</b> -88.20	<b>Level</b> 1	<b>Date Level of Confidence</b> 5 day	<b>N</b> 4
			<b>New Bulk Upload</b> <b>Data File:</b> Choose File Sorghum_Ma...ormat.csv <a href="#">View List of Recognized Traits</a>	<b>Cultivar</b> 10: Panicum virgatum Cave-In-Rock
			<b>Upload and validate file &gt;&gt;</b>	<b>Add covariates to this trait</b> <b>Variable</b> leafT - degrees C <b>Level</b> 25 <a href="#">remove</a>
				<b>Variable</b> PAR - umol m-2 s-1 <b>Level</b> 2000 <a href="#">remove</a>
				<b>Treatment</b> 743: observational - P

---

# Table of Contents

Introduction	1.1
Table of Contents	1.1.1
Getting Started	1.2
Finding and Preparing Published Data	1.3
Preparing Published Data	1.3.1
Entering Meta-Data	1.4
Adding a citation	1.4.1
Adding a Site	1.4.2
Adding a Treatment	1.4.3
Adding Managements	1.4.4
Adding PFTs, Species, and Cultivars	1.4.5
Adding Trait and Yield Data	1.5
Web Interface	1.5.1
Bulk Upload	1.5.2
Adding Traits via the Beta API	1.5.3
Appendices	1.6
Quality Assurance	1.6.1
Extracting Data From Figures	1.6.2
Estimating SE from Summary Statistics	1.6.3
Common Unit Conversions	1.6.4
Acknowledgements	1.7

# BETYdb Data Entry Workflow

## Introduction

BETYdb is used to manage and distribute agricultural and ecological data. This book provides instruction on entering data through the BETYdb web interface. The web interface provides a sequence of pages that walk through the process of entering meta-data, and then the option of entering trait and yield data through a similar web form or via upload of a text (csv) file. For entering large tables of data there is Bulk Upload Wizard. This is useful when entering more than a few dozen trait or yield data from a single source.

There are typically two categories of data that are entered:

1. Results from previously published research, typically statistical summaries.
2. Primary data, observations at the level of an individual replicate.

BETYdb supports both of these, because it was designed to support new research that *quantitatively* builds on previous research, allowing researchers to develop, test, and evaluate new hypotheses based on what is already known. For users interested in entering their own data, the protocol is much simpler than the dense prose implies. This is because people collecting new data a) have more complete information than is provided in scientific publications and b) the ratio of data to meta-data is much higher compared to extracting statistical summaries from the literature.

Therefore, sections on how to interpret and transform statistics, enter missing dates, extract data from figures and tables, and entering trait and yield data one at a time are helpful for meta-analysis but not necessary for primary data. For primary data enter the relevant metadata (sites, treatments, managements) and then upload a `.csv` file as described in the section '[Bulk Upload](#)' in the chapter '[Adding Trait and Yield Data](#)'.

This document provides a comprehensive protocol for entering previously published data into BETYdb for meta-analysis. In particular, the section 'Finding and Preparing Data' details how to manage the search, review, annotation, and extraction of information from previously published papers to facilitate the collaboration between scientists and data entry technicians.

This document has guided research teams through the process of extracting data from hundreds of published scientific articles. The general approach is to dividing the task of identifying which data to enter, which must be done by a scientist, from other steps that can be done by a data entry technician (typically undergraduate biology majors).

## Authors

David LeBauer, Moein Azimi, David Bettinardi, Rachel Bonet, Emily Cheng, Michael Dietze, Patrick Mulrooney, Scott Rohde, Andy Tu

# Table of Contents

# Overview

This is the user guide for entering data into the BETYdb database. The goal of this guide is to provide a consistent method of data entry that is transparent, reproducible, and well documented. The steps here generally accomplish one of two goals.

The first goal is to provide data that is associated with the experimental methods, species, site, and other factors associated with the original study. The second goal is to provide a record of all the transformations, assumptions, and data extraction steps used to migrate data from the primary literature to the standardized framework of the database.

## Getting Started

### Sign up for BETYdb

At a minimum, you should create an account on the instance of BETYdb that you will be using. For example, [BETYdb.org](https://betydb.org). Other instances of BETYdb include:

Institution	url	Maintainer
University of Illinois	<a href="https://betydb.org">https://betydb.org</a>	David LeBauer
Boston University	<a href="https://psql-pecan.bu.edu/bety">https://psql-pecan.bu.edu/bety</a>	Mike Dietze
Brookhaven National Lab	<a href="http://modex.test.bnl.gov/bety">http://modex.test.bnl.gov/bety</a>	Shawn Serbin
University of Wisconsin	<a href="http://tree.aos.wisc.edu:6480/bety">http://tree.aos.wisc.edu:6480/bety</a>	Ankur Desai
TERRA Ref	<a href="https://terraref.ncsa.illinois.edu/bety">https://terraref.ncsa.illinois.edu/bety</a>	David LeBauer

### Other Accounts

There are multiple research groups running the database. To use the database; request "creator" access during signup to enter data; request "manager" to perform QA/QC.

- [Mendeley](#) to track and annotate citations.
- [Google Docs](#) to prepare and transform data prior to entry.
- [Github](#) to track data that need to be checked and/or corrected.



## Finding data

BETYdb is designed for both previously published data and 'primary' data. Most of this documentation assumes that you have already identified a data set that you want to upload, or have a set of papers from which you would like to extract data and summary statistics.

## Meta-analyses

If you are planning to do a meta-analysis, even if this is not your first time, please read 'Uses and Misuses of Meta-analysis in Ecology' \cite{Koricheva\_2014}. Many texts are available, but the recent "Handbook of Meta-analysis in Ecology and Evolution" is probably the most comprehensive and specific for plant sciences.

For a meta-analysis, the first step is to find papers that contain the target data.

The easiest approach to use a search engine such as [Web of Science](#), [Google Scholar](#), or [Microsoft Academic Search](#). Starting with queries such as "*scientific name + trait*", and allowing these results to guide further queries. Often, the references (particularly of meta-analyses and reviews) and forward citations will point to other studies.

Another starting point for the programmatically inclined - which aids in documenting searches - is to submit queries programmatically. Carl Davidson wrote a [python script](#) to search for citations based on species and trait name. In addition, the rOpenSci project has a [suite of R packages for searching publications](#).



# Preparing Publications for Data Entry

## \label{sec:preparing\_publications}

### Mendeley

Mendeley provides a central location for the collection, annotation, and tracking of the journal articles that we use. Features of Mendeley that are useful to us include:

- Collaborative annotation & notes sharing
  - Text highlighter
  - Sticky notes for comments in the text
  - Notes field for text notes in the reference documentation
- Read/ unread & favorites: Papers can be marked as **read** or **unread**, and may be **starred**.
- Groups
- Tagging

Each project has two groups: "projectname" and "projectname\_out" for the papers with data to be entered and for the papers with data that has been entered, respectively. Papers in the \_out group may contain data for future entry (for example, traits that are not listed in Table \ref{tab:traits}).

Each project manager may have one or more projects and each project should have one group. Group names should refer to plant species, plant functional types, or another project specific name. Please make sure that David LeBauer is invited to join each project folder.

1. Open Mendeley desktop
2. Click `Edit` → `New Group` OR `Ctrl+Shift+M`
3. Create group name following instructions above
4. Enter group name
5. Set `Privacy Settings` → `Private`
6. Click `Create Group`
7. Click `Edit Settings`
8. Under `File Synchronization`, check `Download attached files to group`

### Adding and Annotating Papers

When naming a group, tag folders so that instructions for a technician would include the folder and the tag to look for, e.g. "please enter data from projectx" or "please enter data from papers tagged y from project x". To access the full text and PDF of papers from off campus, use the [UIUC VPN](#) service. If you are managing a Mendeley folder that undergraduates are actively entering data from, please plan to spend between 15 min and 1 hour per week maintaining it - enough to keep up with the work that the undergraduates are doing.

## Adding a reference

- If the DOI number is available (most articles since 2000)
  1. Select project folder
  2. Right click and select `Add entry manually...`
  3. Paste DOI number in *DOI* field
  4. Select the search spyglass icon
  5. Drag and drop PDF onto the record.
- If DOI not available:
  1. Download the paper and save as `citation_key.pdf`
  2. Add using the *Files* field
  3. The citation key should be in `authorYYYYabc` where `YYYY` is the four digit year and `abc` is the acronym for the first three words excluding articles (the, a, an), prepositions (on, in, from, for, to, etc...), and the conjunctions (for, and, nor, but, or, yet, so) with less than three letters.

## Annotating a Reference

Each week, please identify and prepare papers that you would like to be entered next by completing the following steps:

1. Use the star label to identify the papers that you want the student to focus on next.
  - Start by keeping a minimum of 2 and a maximum of 5 highlighted at once so that students can focus on the ones that you want. Students have been entering 1-3 papers per week, once we get closer to 3-5, the min/max should change.
  - Choose papers that are the most data rich.
2. For each paper, use comment bubbles, notes field, and highlighter to indicate:
  - Name(s) of traits to be collected
  - Methods:
    - Site name
    - Location
    - Number of replicates

- Statistics to collect
- Identify treatment(s) and control
- Indicate if study was conducted in greenhouse, pot, or growth chamber
- Data to collect
  - Identify figures number and the symbols to extract data from.
  - Table number and columns with data to collect
- Covariates
- Management data (for yields)
- Units in 'to' and 'from' fields used to convert data
- Esoteric information that other scientists or technicians might not catch and that is not otherwise recorded in the database
- Any data that may be useful at a later date but that can be skipped for now.

**Comment or Highlight** the following information

- Sample size
- Covariates (see table \ref{tab:covariates})
- Treatments
- Managements
- Other information entered into the database, e.g. experimental details

## Finding a citation in Mendeley

To find a citation in Mendeley, go to the project folder. By default, data entry technicians should enter data from papers which have been indicated by a yellow star and in the order that they were added to the list. Information and data to be collected from a paper can be found under the 'Notes' tab and in highlighted sections of the paper.

## Recording extracted data and transformations

Google Spreadsheets are used to keep a record of any data that is not entered directly from the original publication. Please share all spreadsheets with the user betydb@gmail.com in addition to any collaborators.

- Any raw data that is not directly entered into the database but that is used to derive data or stats using equations in Tables \ref{tab:conversions} or \ref{tab:stats}.
- Any data extracted from figures, along with the figure number
- Any calculations that were made. These calculations should be included in the cells.

Each project has a Google document spreadsheet with the title "project\_data". In this spreadsheet, each reference should have a separate worksheet labeled with the citation key ( authorYYYabc format). Do not enter data into excel first as this is prone to errors and information such as equations may be lost when uploading or copy-pasting.

## Data Entry Overview

Before entering data, it is first necessary to add and select the citation that is the source of the data. It is also necessary for each data point to be associated with a Site, Treatment, and Species. Cultivar information is also required when available, but it is only relevant for domesticated species. Fields with an asterisk (\*) are required.

## Adding a Citation

Citation provides information regarding the source of the data. A PDF copy of each paper should be available through Mendeley.

1. Select one of the starred papers from your project's Mendeley folder.
2. The data to be entered should be specified in the notes associated with the paper in Mendeley
3. Identify (highlight or underline) the data (means and statistics) that you will enter
4. Enter citation information
  - [Data entry form](#) for a new site: BETYdb → Citations → new
  - **Author:** Input the first author's last name only
  - **Year:** Input the year the paper was published, not submitted, reviewed, or anything else
  - Fill out Title, Journal, Vol, & Pg. For unknown information, input 'NA'
  - **DOI:** The 'digital object identifier'. If DOI is available, PDF and URL are optional. This can be located in the article or in the article website. Use Ctrl+F 'DOI' to find it. Some older articles do not have a DOI. When entering the DOI, don't include a "doi:" prefix; the DOI should start with "10."
  - **URL:** Web address of the article, preferably from publisher's website. Include the "http://" or "https://" prefix. If no on-line version is available, but some other information about how to obtain the citation is available, you may use a parenthesized note in lieu of a bona fide URL, e.g. "(e-mail Dr. No at no@example.com for a written copy)".
  - **PDF:** URL of the PDF of the article. Include the "http://" or "https://" prefix. (A parenthesized note is allowed here as well.)

Energy Biosciences Institute

BETYdb

Biofuel Ecophysiological Traits and Yields Database

Logged in as: Moein Azimi

Home

Data

Docs

Runs

Model I/O

Logout

New Citation

Author

Year

Title

Journal

Vol

Pg

Doi

Url

Pdf

Back

Create

15

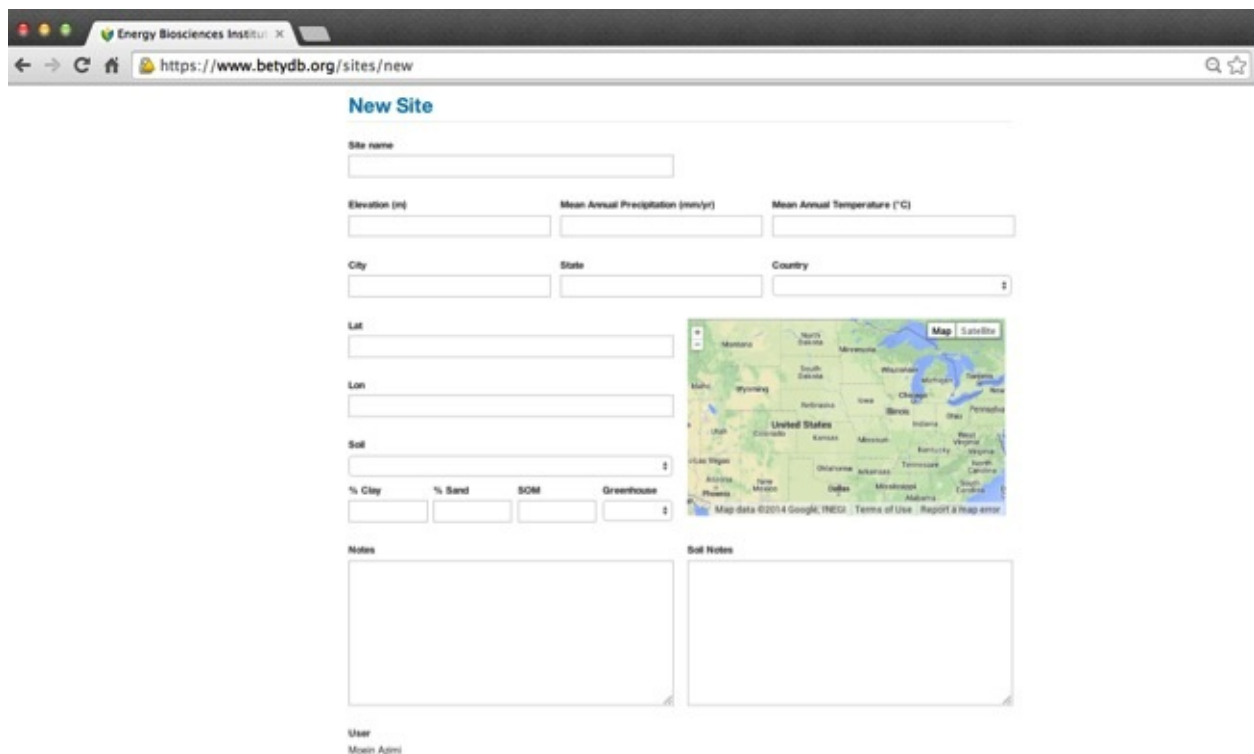
## Adding a Site

Each experiment is conducted at a unique site. In the context of BETY, the term 'site' refers to a specific location and it is common for many sites to be located within the same experimental station. By creating distinct records for multiple sites, it is possible to differentiate among independent studies.

1. Before adding a site, search to make sure that site is not already entered in the database.
2. Search for the site given latitude and longitude
  - If an institution name or city and state are given, try to locate the site on Google Maps
  - If a site name is given, try to locate the site using a combination of Google and Google Maps
  - If latitude and longitude are given in the paper, search by lat and lon, which will return all sites within  $\pm 1$  degree lat and long.
  - If an existing site is plausibly the same site as the one mentioned in the paper, it will be necessary to check other papers linked to the existing site.
    - Use the same site if the previous study uses the *exact same location* and experimental setup.
    - Create a new site if the study was conducted in a different field (i.e., not the exact same location).
    - Create a new site if one study was conducted in a greenhouse and another was conducted in a field.
    - Do not use distinct sites for seed source in a common garden experiment (see 'When not to enter a new site' below)
3. To use an existing site, click `Edit` for the site, and then select current citation under `Add Citation Relationships`
4. If site does not exist, add a new site.

Interface for adding a new site:





**New Site**

Site name

Elevation (m)  Mean Annual Precipitation (mm/yr)  Mean Annual Temperature (°C)

City  State  Country

Lat

Lon

Soil

% Clay  % Sand  SOM  Greenhouse

Notes

Soil Notes

User  
Moeen Admi

## Attributes of a site record

Descriptors	Notes
Site Name	Site identifier, sufficient to uniquely identify the site within the paper
City	Nearest city
State	State, if in the US
Country	Country
Longitude	Decimal Form. For conversion see the equation in table 9
Latitude	Decimal Form. For conversion see the equation in table 9
Greenhouse	TRUE if plants were grown in a greenhouse, growth chamber or pots.
Soil	By percent clay, sand, and silt if given
SOM	Soil organic matter (% by weight)
MAT	Mean Annual Temperature (°C)
MAP	Mean Annual Precipitation (mm)
MASL	Elevation (meters above sea level, m)
Notes	Site Details not included above
Soil Notes	Soil details not included above
Rooting Zone Depth	Measured in Meters (m)
Depth of Water Table	Measured in Meters (m)

1. Do **not** enter a new site When plants (or seeds) are collected from multiple locations and then grown in the same location, this is called 'common garden experiment'. In this case, the location of the study is included as site information. Information about the seed source can be entered as a distinct cultivar.

## Site Location

Points can be added via the web interface; spatial geometries, e.g. a plot, site, or country boundary, must be entered via the PostgreSQL command line.

## Point Locations

If latitude and longitude coordinates are not available, it is often possible to determine the site location based on the site name, city, and other information. One way to do this would be to look up a location name in [Google Maps](#) and then locate it on the embedded map. Google Maps can provide decimal degrees if the LatLng feature is enabled, which can be done [here](#). Google Earth can be particularly useful in locating sites, along with their coordinates and elevation. Alternatively, the site website or address might be found through an internet search (e.g. Google).

Use Table \ref{tab:location\_accuracy} to determine the number of significant digits to indicate the level of precision with which a study location is known.

**Table \ref{tab:location\_accuracy}** \label{tab:location\_accuracy} Level of accuracy to record in lat and lon fields.

Location Detail	Degree Accuracy
City	0.1
Mile	0.01
Acre	0.001
10 Meters	0.0001

## Boundaries

A vector boundary must be obtained. Here is one way to obtain a site boundary using R:

### A rectangular plot (with bounding box)

Here I set the bounding box for a plot by specifying the plot corners and elevation. Notice that it is necessary to specify the first point twice, once at the beginning and once at the end.

```
UPDATE sites
SET geometry = ST_Geomfromtext('POLYGON((-76.116081 42.794448 415,
                                           -76.116679 42.794448 415,
                                           -76.116679 42.79231 415,
                                           -76.116081 42.79231 415,
                                           -76.116081 42.794448 415))', 4326)

WHERE
  ID = 1123;
```

### A country boundary:

```
library(prevR)# for `create.boundary` function
library(sp)
library(rgeos)

UK_boundary <- create.boundary('United Kingdom')
writeLines(
  paste("insert into sites (country, sitename, geometry) values ('UK', 'United Kingdom
', ST_GEOFromText('",
    writeWKT(UK_boundary), "'",4326)) ;"), con = 'uk.sql')
```

Then import at the command line (can also copy / paste to terminal, but this boundary is long)

```
psql -U bety -d bety < uk.sql
```

## References

- PostGIS `ST_GeomFromText` documentation:  
[http://www.postgis.org/docs/ST\\_GeomFromText.html](http://www.postgis.org/docs/ST_GeomFromText.html)
- gis.stackexchange: <http://gis.stackexchange.com/q/111212/1239>
- Github issues: <https://github.com/PecanProject/pecan/issues/570>

# Adding Treatments and Managements

## Treatments

Treatments provide a description of a study's treatments. Any specific information such as rate of fertilizer application should be recorded in the managements table. In general, managements are recorded when Yield data is collected, but not when only Trait data is collected.

**When not to use treatment:** predictor variables that are not based on distinct managements, or that are distinguished by information already contained in the trait (e.g. site, cultivar, date fields) should not be given distinct treatments. For example, a study that compares two different species, cultivars or genotypes can be assigned the same control treatment; these categories will be distinguished by the species or cultivar field. Another example is when the observation is made at two sites: the site field will include this information.

- A treatment name is used as a categorical (rather than continuous) variable: it should be easy to find the treatment in the paper based on the name in the database. The treatment name does not have to indicate the level of treatment used in a particular treatment - this information will be included in the management table.
- It is essential that a control group is identified with each study. If there is no experimental manipulation, then there is only one treatment. In this case, the treatment should be named 'observational' and listed as control. To determine the control when it is not explicitly stated, first determine if one of the treatments is most like a background condition or how a system would be in its non-experimental state. In the case of crops, this could be how a farmer would be most likely to treat a crop.

**Name:** indicates type of treatment; it should be easy for anyone with the original paper to be able to identify the treatment from its name.

**Control:** make sure to indicate if the treatment is the study 'control' by selecting true or false

**Definition:** indicates the specifics of the treatment. It is useful for identification purposes to use a quantified description of the treatment even though this information can only be used for analysis when entered as a management.

Energy Biosciences Institute

https://www.betydb.org/treatments/new

### New Treatment

Name

Definition

Control

User

Moein Azimi

Back

Create

BETYdb

Homepage

Documentation

Maps & Data

Contact Info

Report a problem

Send us a message

Translate Page

Select Language

Energy Biosciences Institute

BETYdb

David LeBauer, Dan Wang, and Michael Dietze, 2010. Biofuel Ecophysiological Traits and Yields Database Version 1.0. Energy Biosciences Institute, Urbana, IL.

Copyright © 2010-2013 Energy Biosciences Institute

# Adding Managements

There are two ways to add management information, through the web interface or from a spreadsheet. These are discussed in turn, below. Recall that managements can be associated with one or more treatments.

Managements refers to something that occurs at a specific time and has a quantity. Managements include actions that are done to a plant or ecosystem, such as the planting density or rate of fertilization, for example. Managements are distinct from treatments in that a treatment is used to categorically identify an experimental treatment, whereas a management is used to describe what has been done. Managements are the way a treatment becomes quantified. Each treatment is often associated with multiple managements. The combination of managements associated with a particular treatment will distinguish it from other treatments. The management types that can be entered into BETY are described in Table \ref{tab:managements}. Each management may be associated with one or more treatments. For example, in a fertilization experiment, planting, irrigation, and herbicide managements would be applied to all plots but the fertilization will be specific to a treatment. For a multi-year experiment, there may be multiple entries for the same type of management, reflecting, for example, repeated applications of herbicide or fertilizer.

*note:*Managements are not always required - and the level of detail depends on the scope of research. By default managements are recorded for Yields but not for Traits, unless specifically required by the data or project manager.

- **Date:** in format YYYY-MM-DD OF YYYY-MM-DD HH:MM
- **Datoloc:** date level of confidence, explained in Section \ref{sec:traits} and defined in Table \ref{tab:traits}.
- **Mgmttype:** the name of the management being used. A list of standardized management types can be found in Table \ref{tab:managements}
- **Level:** a quantification of mgmttype
- **Units:** refers to the units of the level. Units should be converted to those in Table \ref{tab:managements}

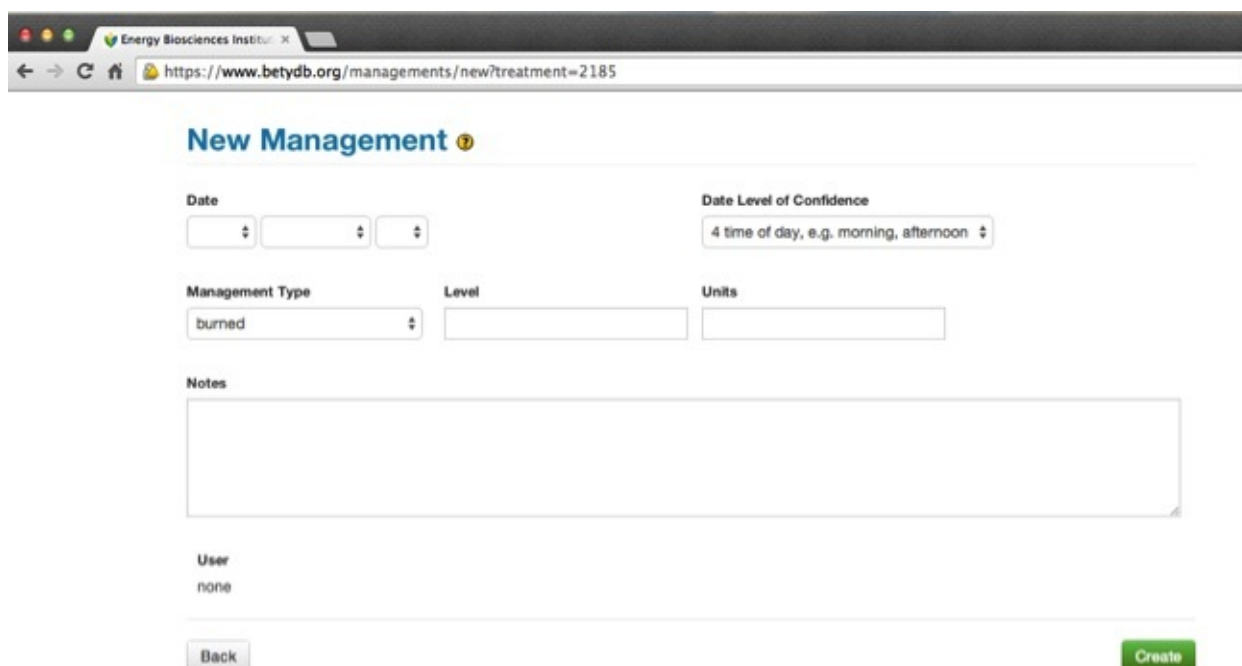
**Managements** This is a list of managements to enter, with the most common management types in bold. It is more important to have management records for Yields than for traits. For greenhouse experiments, it is not necessary to include informaton on fertilizaton, lighting, or greenhouse temperature.

Management Type	Units	Definition	Notes
Burned	aboveground biomass burned		
CO2 fumigation	ppm		
Fertilization_X	kg x ha <sup>{-1}</sup>	fertilization rate, element X	
Fungicide	kg x ha <sup>{-1}</sup>		add type of fungicide to notes
Grazed	years	livestock grazing	pre-experiment land use
Harvest			no units, just date, equivalent to coppice, aboveground biomass removal
Herbicide	kg x ha <sup>{-1}</sup>		add type of herbicide to notes: glyphosate, atrazine, many others
Irrigation	cm		convert volume \ area to depth as required
Light	W m <sup>{-2}</sup>		
O3 fumigation	ppm		
Pesticide	kg x ha <sup>{-1}</sup>		add type of pesticide to notes
Planting	plants m <sup>{-2}</sup>		Convert row spacing to planting density if possible
Seeding	kg seeds x ha <sup>{-1}</sup>		
Tillage			no units, maybe depth; <i>tillage</i> is equivalent to <i>cultivate</i>

## Via Web interface

Managements can be entered via the web interface. First enter the management, and then associate it with one or more treatments. To associate a management with multiple treatments, first create the management, then edit the management and add treatment relationships.





The screenshot shows a web browser window with the URL <https://www.betydb.org/managements/new?treatment=2185>. The page title is "New Management". The form contains the following fields:

- Date:** Three dropdown menus for day, month, and year.
- Date Level of Confidence:** A dropdown menu with the value "4 time of day, e.g. morning, afternoon".
- Management Type:** A dropdown menu with the value "burned".
- Level:** An empty text input field.
- Units:** An empty text input field.
- Notes:** A large text area for notes.
- User:** A dropdown menu with the value "none".

At the bottom of the form, there are two buttons: "Back" and "Create".

## Preparing a managements spreadsheet for Upload

When there is a long list of managements, the `insert_managements` scripts enables users to insert data organized in a text based (csv) file.

Preparing the csv file can be done in any spreadsheet program such as Excel or Google Sheets. The insertion is straightforward, but requires familiarity with the bash shell as well as administrative access to the Postgres database.

### File format

**Required Fields** the spreadsheet or CSV file must contain the following column headings:

```
citation_author
citation_title
citation_year
treatment_name
mgmttype
```

These columns map to fields in the database (in the citations, treatments, and managements field). Each row must have non-empty values in each of these columns. Moreover, the citation columns must match exactly one row in the citations row of the database and the treatment name must match exactly one of the treatment rows associated with the matched citation.

**Optional Fields** The table *may* also contain the following column headings:

```
date
dateloc
level
units
notes
```

Each optional column heading corresponds to an optional field in the database managements table. The column can contain one or more empty rows.

If the table is prepared in a spreadsheet program, use the "save as --> .csv" option to export a single text based .csv file.

## Inserting Management Insertion Script

The `insert_managements.rb` script takes a CSV file describing managements to be added to the database as input and outputs a file containing SQL statements to do the required insertions.

The script `insert_managements.rb` is in the directory `RAILS_ROOT/script`. The complete usage instructions (also obtainable by running `./insert_managements --man`) follow. For additional information, see [Github issue #288](#)

### `insert_managements.rb`

Usage:

```
insert_managements [options] <CSV input file>
```

where [options] are:

<code>-u, --login=&lt;s&gt;</code>	The Rails login for the user running the script (required)
<code>-o, --output=&lt;s&gt;</code>	Output file (default: new_managements.sql)
<code>-e, --environment=&lt;s&gt;</code>	Rails environment to run in (default: development)
<code>-m, --man</code>	Show complete usage instructions
<code>-h, --help</code>	Show this message

## DATABASE SPECIFICATION

The database used by the script is determined by the environment specified by the '--environment' option (or 'development' if not specified) and the contents of the configuration file 'config/database.yml'.

(Run 'rake dbconf' to view the contents of this file on the command line.)

## USING THE SCRIPT TO UPDATE THE PRODUCTION DATABASE

There are three options for using this script to update the production database.

**Option A:** Run the script on the production server in the Rails root directory of the production deployment of the BETYdb Rails app.

1. Upload the input CSV file to the production machine.
2. Log in to the production machine and `cd` to the root directory of production deployment of the BETYdb Rails app.
3. Run the script using the '--environment=production' option and with '--login' set to your own BETYdb Rails login for the production deployment. The command-line argument specifying the input CSV file path should match the location you uploaded it to.
4. After examining the resulting output file, apply it to the database with the command

```
psql <production database name> < <output file name>
```

(If your machine login doesn't match a PostgreSQL user name that has insert permissions on the production database, you will have to use the '-U' option to specify a user who does have such permission.)

**Option B:** Run the script on your local machine using an up-to-date copy of the BETYdb database.

To do this:

1. Switch to the root of the copy of the BETYdb Rails app you want to use.
2. For the copy of the BETYdb database connected to this copy of the Rails app, ensure that at least the citations and the treatments tables are up-to-date with the production copy of the BETYdb database. (If you have different databases specified for your development and your production environments, be sure that the environment you specify with the '--environment' option points to the right database.)
3. Run this script.
4. Upload the output file to the production server and apply it to the production database using the `psql` command given above.

**Option C:** Run the script on your local machine using a Rails environment connected to the production database.

1. Go to the copy of the BETYdb Rail app on your local machine that you wish to use.
2. Edit the file `config/database.yml`, adding the following section:

```
ebi:
  adapter: postgres
  encoding: utf8
  reconnect: false
  database: <production database name>
  pool: 5
  username: <user name for connecting to the production database>
  password: <password for the user specified above>
  port: 8000
  host: localhost
```

Most of these values can be copied from the production copy `config/database.yml` if you have access to it. The port and host entries are 'new'.

3. Set up an ssh tunnel to the production server using the command

```
ssh -L 8000:<production server address>:5432 <production server address>
```

This will log you into the production server, but at the same time it will connect port 8000 on your local machine with port 5432 (the PostgreSQL server port) on the production machine. (The choice of 8000 for port number is somewhat arbitrary, but whatever value you use should match the value you specified for the port number in the database.yml file.)

4. Run this script with the environment option '--environment=ebi'. (Again, the name 'ebi' for the environment is somewhat arbitrary, but the option value should match the name in your database.yml file.)
5. Continue as in step 4 under option B.

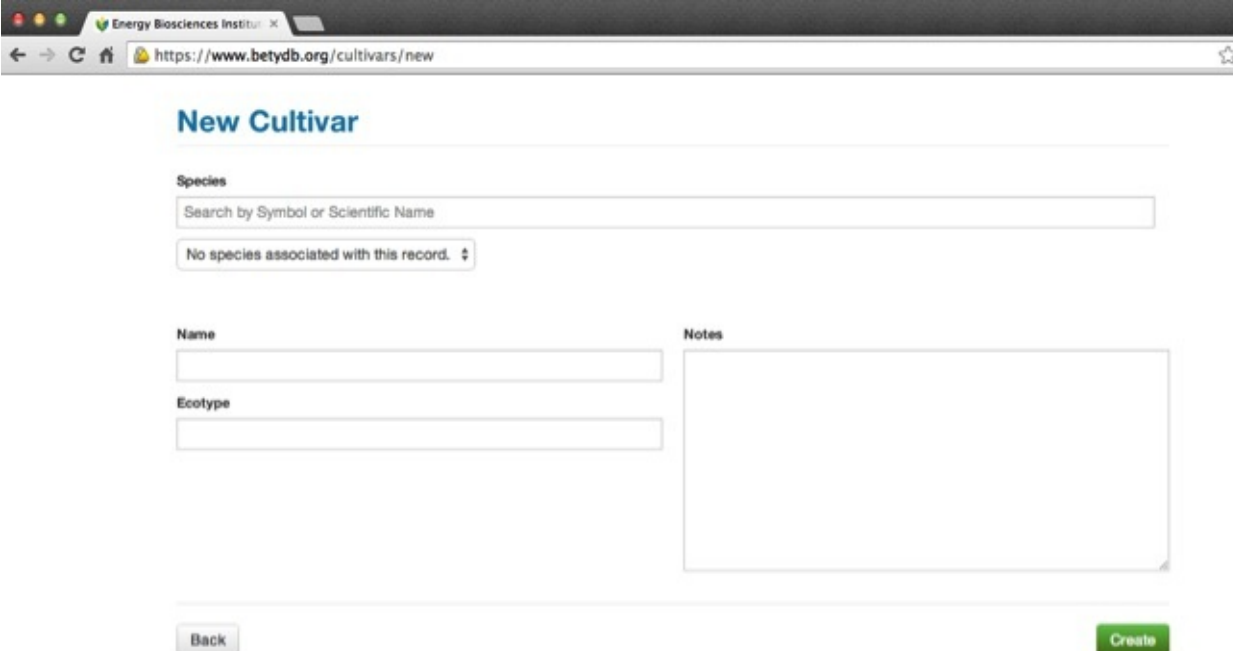
# Adding a PFT, Species, or Cultivar

Plant functional types (PFTs) are used to group plants for statistical modeling and analysis. PFTs are associated with both a specific set of priors, and a subset species for which the traits and yields data will be queried. In many cases, it is appropriate to use default PFTs (e.g. `tempdecid` is temperate deciduous trees)

In other cases, it is necessary to define PFTs for a specific project. For example, to query a specific set of priors or a subset of a species, a new PFT may be defined. For example, Xiaohui Feng defined PFTs for the species found at the EBI Farm prairie. Such project-specific PFTs can be defined as ``projectname`.`pft`` (i.e. `ebifarm.c4grass` instead of `c4grass` ).

Species that are found or cultivated in the United States should be in the Plants table. Look it up there first.

To add a new Cultivar, go to the [new cultivar](#) page: `Cultivar` → `new` .



The screenshot shows a web browser window with the URL `https://www.betydb.org/cultivars/new`. The page title is "New Cultivar". It features a "Species" section with a search bar labeled "Search by Symbol or Scientific Name" and a message "No species associated with this record." Below this are input fields for "Name" and "Ecotype", and a large text area for "Notes". At the bottom, there are "Back" and "Create" buttons.

## Adding Trait and Yield Data

## Adding a Trait

The screenshot shows the 'New Trait' form in the BETYdb web interface. The form is titled 'New Trait' and contains several input fields and sections. At the top, there are fields for 'Mean', 'Std', 'Method', and 'Statname'. Below these are fields for 'Date', 'Date Level of Confidence', 'Time', and 'Time Level of Confidence'. The 'Site' field is populated with '1133: Confluence of Casiquiare River and Rio Negro - San Carlos'. The 'Species' field has a search bar and a note 'No species associated with this record'. The 'Cultivar' field is empty, and the 'Treatment' field is populated with '2186: Succession : growth following fire - LHO 1987'. The 'Trait' field is populated with '542: -', and the 'N' field is empty. The 'Access level' field is populated with '2,558 Researchers'. There is a 'Notes' section with a text area. Below the notes, there is a link 'Add a covariate to this trait' and a section for 'Variable' and 'Level'. At the bottom, there is a 'User' field with the value 'Moore Admin' and a 'Create' button.

In general, a 'trait' is a phenotype (a characteristic that the plant exhibits). The traits that we are primarily interested in collecting data for are listed in Table [Table 1](#). Before adding trait data, it is necessary to have the citation, treatments, and site information already entered. If the correct citation is not identified at the top of the page [Figure 8](#). To add a new Trait, go to the [new trait](#) page: Trait → new .

### Key Traits Stored in BETYdb

Variable	Units	Median (90%CI) or Range	Definition
Vcmax	$\mu\text{mol CO}_2 \text{ m}^{-2} \text{ s}^{-1}$	(44 (12, 125))	maximum rubisco carboxylation capacity
SLA	$\text{m}^2 \text{ kg}^{-1}$	(15(4,27))	Specific Leaf Area area of leaf per unit mass of leaf
LMA	$\text{kg m}^{-2}$	(0.09 (0.03, 0.33))	Leaf Mass Area (LMA = SLM = 1/SLA) mass of leaf per unit area of leaf
leafN	%	(2.2(0.8, 17))	leaf percent nitrogen
c2n leaf	leaf C:N ratio	(39(21,79))	use only if leafN not provided

leaf turnover rate	1/year	(0.28(0.03,1.0))	
J <sub>max</sub>	( $\mu$ ) mol photons m <sup>-2</sup> s <sup>-1</sup> )	(121(30, 262))	maximum rate of electron transport
stomatal slope		(9(1, 20))	
GS			stomatal conductance (= $g_{s(\text{max})}$ )
q*		0.2--5	ratio of fine root to leaf biomass
*grasses	ratio of root:leaf = below:above ground biomass		
aboveground biomass	g m <sup>-2</sup> ) or g plant <sup>-1</sup> )		
root biomass	g m <sup>-2</sup> ) or g plant <sup>-1</sup> )		
*trees	ratio of fine root:leaf biomass		
leaf biomass	g m <sup>-2</sup> ) or g plant <sup>-1</sup> )		
fine root biomass (<2mm)	g m <sup>-2</sup> ) or g plant <sup>-1</sup> )		
root turnover rate	1/year	0.1--10	rate of fine root loss (temperature dependent) year <sup>-1</sup> )
leaf width	mm	22(5,102)	
growth respiration factor	%	0--1	proportion of daily carbon gain lost to growth respiration
R( <sub>dark</sub> )		( $\mu$ ) mol CO <sub>2</sub> m <sup>-2</sup> s <sup>-1</sup> )	dark respiration
quantum efficiency	%	0--1	efficiency of light conversion to carbon fixation, see Farquhar model
dark respiration factor	%	0--1	converts V <sub>m</sub> to leaf respiration
			proportion of seedlings



			that die
r fraction	%	0--1	fraction of storage to seed reproduction
root respiration rate*	CO <sub>2</sub> kg <sup>(-1)</sup> fine roots s <sup>(-1)</sup>	1--100	rate of fine root respiration at reference soil temperature
f labile	%	0--1	fraction of litter that goes into the labile carbon pool
water conductance			

Presently, we are also using the Trait table to record ecosystem level measurements other than Yield. Such ecosystem level measurements can include leaf area index or net primary productivity, but are only collected when required for a particular project. Most of the fields in the Traits table are also used in the Yields table. Here is a list of the fields with a brief description, followed by more thorough explanations:

- **Species:** Search for species in the database using the search box; if species is not found, then the new species should be added to the database.
- **Cultivar:** primarily used for crops; If the cultivar being used is not found in drop-down box
- **DateLOC:** Date Level of confidence. See for values.
- **TimeLOC:** Time Level of confidence. See for values.
- **Mean:** For yield, mean is in units of tons per hectare per year (t/ha)
- **Stat name:** is the name of the statistical method used (usually one of SE, SD, MSE, CI, LSD, HSD, MSD). See for more details.
- **Statistic:** is the value of the statistic associated with Stat name.
- **N:** Always record N if provided. N is the number of experimental replicates, often referred to as the sample size; N represents the number of independent units within each treatment: in a field setting, this is often the number of plots in each treatment, but in a greenhouse, growth chamber, or pot-study this may be the number of chambers, pots, or individual plants. Sometimes this value is not clearly stated.

## Uncertainty in Date or Time

### DateLOC

The date level of confidence (DateLOC, Table \ref{tab:dateloc}) provides an indication of how accurately the date associated with the trait or yield observation is known. It provides the values that should be entered in this field. If the event occurred at a level of precision not

The date level of confidence (DateLOC, Table \ref{tab:dateloc}) provides an indication of how accurately the date associated with the trait or yield observation is known. It provides the values that should be entered in this field. If the event occurred at a level of precision not defined by an integer in this table, then use fractions. For example, we commonly use 5.5 to indicate a one week level of precision. If the exact year is not known, but the time of year is, then use 91 to 97, with the second digit to indicate the information known within the year.

**Table Date level of confidence (DateLOC) field** Numbering convention for the DateLOC (Date level of confidence) and TimeLOC (Time level of confidence) field, used in managements, traits, and yields table.

Dateloc	Definition
9	no data
8	year
7	season
6	month
5	day
95	unknown year, known day
96	unknown year, known month
...etc	

## TimeLOC

The time level of confidence (TimeLOC) provides an indication of how accurately the time associated with the trait or yield observation is known. It provides the values that should be entered in this field.

Timeloc	Definition
9	no data
4	time of day i.e. morning, afternoon
3	hour
2	minute
1	second

## Statistics

Where available, direct estimates of variance are preferred, including Standard Error (SE), sample Standard Deviation (SD), or Mean Squared Error (MSE). SE is usually presented in the format of mean ( $\pm$ SE). MSE is usually presented in a table. When extracting SE or SD from a figure, measure from the mean to the upper or lower bound. This is different than confidence intervals and range statistics (described below), for which the entire range is collected.

If MSE, SD, or SE are not provided, it is possible that LSD, MSD, HSD, or CI will be provided. These are range statistics and the most frequently found range statistics include a Confidence Interval (95%CI), Fisher's Least Significant Difference (LSD), Tukey's Honestly Significant Difference (HSD), and Minimum Significant Difference (MSD). Fundamentally, these methods calculate a range that indicates whether two means are different or not, and this range uses different approaches to penalize multiple comparisons. The important point is that these are ranges and that we record the entire range.

Another type of statistic is a "test statistic"; most frequently there will be an F-value that can be useful, but this should not be recorded if MSE is available. Only if there is no other information available should you record the P-value.

## Adding a Yield

The protocol for entering yield data is identical to entering data for a trait, with a few exceptions:

1. There are no covariates associated with yield data
2. Yield data is always the dry harvestable biomass; if necessary, moisture content can be added as a trait

Yield is equivalent to aboveground biomass on a per-area basis, and has units of  $\text{Mg ha}^{-1} \text{y}^{-1}$

## Adding a Covariate

Covariates are required for many of the traits. Covariates generally indicate the environmental conditions under which a measurement was made. Without covariate information, the trait data will have limited value.

A complete list of required covariates can be found in Table \ref{tab:covariates}. For all respiration rates and photosynthetic parameters, temperature is recorded as a covariate. Soil moisture, humidity, and other such variables that were measured at the time of the

Covariates are required for many of the traits. Covariates generally indicate the environmental conditions under which a measurement was made. Without covariate information, the trait data will have limited value.

A complete list of required covariates can be found in Table \ref{tab:covariates}. For all respiration rates and photosynthetic parameters, temperature is recorded as a covariate. Soil moisture, humidity, and other such variables that were measured at the time of the measurement may be required in order to standardize across studies.

When root data is recorded, the root size class needs to be entered as a covariate. The term 'fine root' often refers to the ( $\leq$ )2mm size class, and in this case, the covariate `root_maximum_diameter` would be set to 2. If the size class is a range, then the `root_minimum_diameter` can also be used.

**Table \ref{tab:covariates}: Traits with required covariates** \label{tab:covariates} A list of traits and the covariates that must be recorded along with the trait value in order to be converted to a constant scale from across studies. *notes:* stomatal conductance ( `gs` ) is only useful when reported in conjunction with other photosynthetic data, such as `Amax` . Specifically, if we have `Amax` and `gs` , then estimation of `Vcmax` only covaries with `dark_respiration_factor` and atmospheric CO2 concentration.

We also now have information to help constrain `stomatal_slope` . If we have `Amax` but not `gs` , then our estimate of `Vcmax` will covary with: `dark_respiration_factor` , `CO2` , `stomatal_slope` , `cuticular_conductance` , and vapor-pressure deficit `VPD` (which is more difficult to estimate than CO2, but still possible given lat, lon, and date). Most important, there will be a strong covariance between `Vcmax` and `stomatal_slope` .



# BETYdb: Bulk Data Upload

## Overview

There are three phases for a basic bulk upload of data:

1. Use the web interface
  - to enter metadata pertaining to your data set (new sites, species, cultivars, citations, or treatments);
  - to obtain a template appropriate for your data set.
2. Fill in the template with your data. There are four templates to choose from:
  - [yields.csv](#) — Use this template if you are uploading yield data and you wish to specify the citation in the file by author, year, and title.

If your data includes standard error and cultivar information and you do not plan to specify any of the required information interactively, you will be able to use this template “as-is”. Otherwise, you will need to delete one or more columns:

- i. If your data has no standard error information, delete both the `SE` and the `n` column.
  - ii. If your data set has a single uniform value for the site, species, cultivar, treatment, access\_level, or date, then these values may be entered interactively through the web interface; in this case you should delete the corresponding column(s) from the template.
  - iii. Note that cultivar information can’t be specified interactively unless species information is as well; delete the `cultivar` column if and only if you either have no cultivar information or you are specifying both the species and the cultivar interactively.
- [yields\\_by\\_doi.csv](#) — Use this template if you are uploading yield data and you wish to specify the citation in the file by doi.

Again, if you do not have data for all of the columns listed in the template, or if you plan to specify some of the data interactively, you will have to delete one or more columns.

You may also use this template if all of the data in your data set pertains to a single citation and you wish to specify that citation interactively. In this case, you must delete the `citation_doi` column.

- [traits.csv](#) — Use this template if you are uploading trait data and you wish to specify the citation in the file by author, year, and title.

**This template must be modified before it can be used.** In particular, the column headings `[trait variable 1] ... [trait variable n]` must be replaced by actual variable names that *exactly* match names of variables in the database that have been marked to be recognized as trait variables. The number of these trait variable columns may need to be increased or decreased to accomodate the data set.

Some trait variables allow or even require corresponding covariate information to be included. Again, the column headings `[covariate 1] ... [covariate n]` must be changed to actual covariate variable names, and the number of these columns may need to be increased or decreased to match the available information. As with the yield data templates, some columns may also need to be deleted. For a list of recognized trait variable names and their corresponding required and optional covariates, visit the trait variable/covariates list at [www.betydb.org](http://www.betydb.org). [TO-DO: Make this Web page.]

- [traits\\_by\\_doi.csv](#) — As with the corresponding yield data template, use this template if you are uploading trait data and you wish to specify the citation in the file by doi or if you plan to specify the citation interactively (in which case delete the `citation_doi` column). **Again, this template must be modified before it can be used.**

1. Use the web interface to upload your data set and insert it into the database.

*In what follows, the term “field” always refers either to a column name used in the heading of the uploaded CSV file or to an entry in that column in some particular row of the file. On the other hand, and the term “column” may either refer to a column of data in the uploaded CSV file or to an attribute of a trait or yield datum in the traits or yields table of the database.*

## Detailed CSV Data File Specifications

[Example of a template for bulk upload of yield data:](#)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	citation_doi	citation_author	citation_year	citations_title	cultivar	species	site	treatment	date	dateloc	mean	n	SE	notes	access_level
2															
3															
4															
5															
6															
7															
8															
9															
10															
11															
12															
13															
14															
15															
16															
17															
18															
19															
20															
21															
22															
23															
24															
25															
26															
27															
28															
29															
30															
31															

## Required fields

1. For yields uploads, the only required field is a `yield` column.
2. For trait uploads, there must be at least one column whose label exactly matches the variable name for the trait value being specified. (Leading and trailing spaces are permitted, but letter case must exactly match the name of the variable specified in the database.) If this trait variable has any required covariates, columns for these covariates must be included.

## Information that is required but that *may* be specified interactively for the entire dataset.

*Data values may be specified interactively only if there is a single value that pertains to the whole data set.*

### **Information that references existing database entries**

1. Citation
  - If only one citation for the entire dataset exists, it may be specified interactively by choosing a citation on the citations page instead of including citation information in the CSV file.
  - Otherwise, specify the citation in the CSV file, either by doi or by author, year, and title.



- If a DOI is available for all citations in the data set, the citation corresponding to each row may be specified in a `citation_doi` column. In this case, the `citation_author`, `citation_year`, and `citation_title` columns must not be in the column heading list. (If such information is already included in the data set, to keep such columns for purely informational purposes, the string `-ignore` may be appended to each of these headings. One might want to do this, for example, to keep a visual record of the author, year, and title even when it is the citation doi that is being used to determine how the data will be associated with a citation in the database.) Each value in the `citation_doi` column must exactly match the `doi` attribute of some row in the `citations` table except that letter case and leading and trailing spaces are ignored.
- Conversely, if a DOI is not available for all citations in the data set, or if it is preferred to specify the citation by author, year, and title, then the `citation_doi` column should *not* be included and the columns `citation_author`, `citation_year`, and `citation_title` must all be present. (Again, if some DOI information is already included and you wish to retain it for purely informational purposes, simply give the column some descriptive name other than `citation_doi` and it will be ignored by the upload code.)

## 2. Site

- If all of the data in the data set pertains to a single site, that site may be specified interactively.
- Otherwise, a `site` column is required. The value must match an existing `sitename` column value in the `sites` table of the database. (Letter case, leading and trailing spaces, and extra internal spaces are ignored when searching for a match.)

## 3. Species

- If all of the data in the data set pertains to a single species, that species may be specified interactively.
- Otherwise, the `species` column is required. The value must match an existing `scientificname` column value in the `species` table of the database. (Again, letter case, leading and trailing spaces, and extra internal spaces are ignored when searching for a match.)

## 4. Treatment

- If a single treatment and a single citation applies to all of the data in the data set, the treatment may be specified interactively provided that the citation is specified interactively as well.
- Otherwise, a `treatment` column is required. The value must match an existing `name` column value in the `treatments` table of the database; moreover, this matching treatment must be consistent with the specified citation. (Again, letter case, leading and trailing spaces, and extra internal spaces are ignored when searching for a match.)

### ***Other information that may be specified interactively***

#### **1. Date**

- If a single date applies to all of the data in the data set, the date may be specified interactively.
- Otherwise, a `date` column is required.
- Date values must be in the form YYYY-MM-DD. For example, July 25, 2003 must be entered as “2003-07-25”. (Eventually, month and day may become optional, in which case any of the forms “2003-07-25”, “2003-07”, and “2003” would represent dates of varying degrees of specificity. Note that uploading dateloc, time, and timeloc information is not supported.)

#### **2. Rounding**

- The amount of rounding for numerical data can only be specified interactively. Any value from 1 to 4 significant digits may be chosen. The amount of rounding for the standard error SE (if present) may be specified separately from the amount of rounding for yield and for trait variables and their covariates.
- By default, all numerical data is rounded to three significant digits. For example, with this default in place, 999.1 will be rounded to 999 and 1001.1 will be rounded to 1000.

## **Numerical Data (This is *never* specified interactively.)**

### ***Data for Yields***

#### **1. Yield**

Every yield data upload file must have a `yield` column. The data in this column must always be a parsable non-negative number and must never be blank. Scientific notation is not currently supported. As noted above, the number given in the file is subject to rounding before being inserted into the database.

## 2. Sample Size

An `n` column is required if and only if an `se` column is included. The value must always be an integer greater than 1.

## 3. Standard Error

An `se` column is required if and only if an `n` column is included; this datum will be inserted into the `stat` column of the `yields` table, and the `statname` column value will be set to "SE".

### **Data for Traits**

#### 1. Trait variable values

- Every trait data upload file must have at least one column whose heading matches the name of some recognized trait variable. A list of recognized trait variables is listed on the BetyDB web site. If multiple trait variable columns are used, each row in the CSV file will produce one row in the `traits` table for each trait variable column. (These resulting rows will be effectively *grouped* by assigning them a unique entity id. Said another way, there is a one-to-one correspondence between rows in the CSV file and resultant rows in the `entities` table, the table that keeps track of this grouping.) As with yield numbers, the data in this column must always be a parsable number and is subject to rounding before being inserted into the database. In addition, it must conform to any range restrictions on the value of the variable.
- The template for traits uploads provides dummy column headings `[trait variable 1]`, `[trait variable 2]`, etc., which must be changed to actual variable names before data can be uploaded.

#### 2. Covariate values

- If any of the included trait variables has a required covariate, there must be a column corresponding to that covariate.
- For any of the included trait variables that has an optional covariate, a column corresponding to that covariate *may* be included.
- The template for traits uploads provides dummy column headings `[covariate 1]`, `[covariate 2]`, etc., which must be changed to actual variable names before data can be uploaded.

#### 3. Sample Size and Standard Error

An `SE` column is required if and only if an `n` column is included; this datum will be inserted into the `stat` column of the `traits` table, and the `statname` column value will be set to “SE”. ***Note that if you have more than one trait variable column, each trait will get the same values of `n` and `SE`. There is currently no way to use different sample size and standard error values for different trait variables. Also, the `n` and `SE` values for any associated covariates will be set to NULL.***

(Eventually, we may enable associating differing values of `n` and `SE` to different trait variables and covariates. In this case, we might add columns `[trait variable 1] n` and `[trait variable 1] SE`, etc. or `[covariate 1] n` and `[covariate 1] SE`, prefixing the usual column heading with a variable name to indicate which variable the sample size and standard error value is to be associated with.)

Again, values of `n` must be at least 2, and columns for `n` and `SE` must both be present or both be absent.

## Optional data

### 1. Sample Size and Standard Error

As noted above, these are both optional, but if one of these is included, the other must be included as well. In other words, the column heading list must include both of `n` and `SE` (or, in the case of traits, `[trait or covariate variable k] n` and `[trait or covariate variable k] SE`) or neither. Note that if `n` and `SE` are not given fields of the uploaded CSV file, the value of the `n` column of the `traits` or `yields` table will default to 1 and the `stat` and `statname` column values will default to NULL.

### 2. Cultivar

- If a uniform value for the species is provided interactively when uploading the data set, the cultivar may be specified this way as well, provided that it also has a uniform value for the whole data set.
- Otherwise, to include cultivar information in the upload file, both a `species` and a `cultivar` column must be included. The values in the `cultivar` column are allowed to be blank (in which case a value of NULL is inserted into the `cultivar_id` column for the given row); but if provided, the value must match the value of the `name` column in some row of the `cultivars` table, and moreover, this row must be a row associated with the species corresponding to the value given in the `species` column. Again, matching is case insensitive, and leading, trailing, and excess internal whitespace is ignored.

### 3. Notes

To include notes, use a `notes` column. There is no restriction on what can be included in this column, but leading and trailing space will be stripped before insertion into the database. Non-ascii characters entered in the file in UTF-8 encoding are allowed. If there is no `notes` column, each row inserted into the `traits` or `yields` table will use the empty string as the value for the `notes` column.

# Inserting New Traits Via the API

This page provides a general description of how to insert trait data via the (new) beta version of the BETYdb API. For information about *accessing* data via the beta BETYdb API, visit [https://pecan.gitbooks.io/betydb-data-access/content/API/beta\\_API.html](https://pecan.gitbooks.io/betydb-data-access/content/API/beta_API.html). For a list of URLs of API endpoints, visit <https://www.betydb.org/api/docs>.

## Trait Insertion Endpoint.

The path to use for trait insertion is `/api/beta/traits(.EXT)` where `EXT` is `csv`, `xml`, or `json`. These are used to submit data in CSV, XML, and JSON format, respectively. If no extension is given, JSON format is assumed. **A user must have *Creator* status (page access level 3) in order to use the trait insertion API.**

## Format of Data Files

To see some valid sample files in all three of the supported formats, see [Running the Examples](#) below.

## Schema for XML Data Files

XML data files must validate against the schema specified by `app/lib/api/validation/TraitData.xsd`. It is possible to validate files on the command line with `xmllint`:

```
xmllint --schema path/to/TraitData.xsd --noout path/to/xml-data-file
```

Note that data files of other types (CSV and JSON) are converted to XML internally and then validated using this schema.

## General Outline and Semantics of XML Data Files

- The root level elements is named `trait-data-set`.
- `trait` elements appear below the root, either directly, or nested within intervening `trait-group` elements.

- A `trait` element must include a `mean` attribute with a value of type double. An exponent may be used; for example, in place of "0.0123" one may write "1.23E-2".
- A `trait` element *may* have the following attributes:
  - An attribute `utc_datetime` whose value has the form "YYYY-MM-DDZ" or "YYYY-MM-DDTHH:MM:SSZ" which represents the date or date and time the trait measurement was taken. The trailing "Z" emphasizes that the value is in UTC time and is required. If a time of day is given, the symbol "T" must separate the time from the date. If a time is given, fractional seconds may be included.
  - A `local_datetime` attribute may be used in place of `utc_datetime`. This represents the date or date and time of a trait measurement in local time. This attribute may only be used if the trait is associated with a site having a specified time zone. The value format is the same as for `utc_datetime` except there is no trailing "Z".
  - An attribute `access_level` *must* be supplied unless it is supplied using the defaulting mechanism (see below).
- A `trait` element *may* have the following child elements:
  - `site` : This *may* have any of the attributes `id`, `city`, `state`, `country`, and `sitename`. It *must* have enough of these attributes to uniquely identify an existing site. `sitename` *should* be unique and is the preferred attribute to use in identifying a site. Unfortunately, uniqueness is not currently enforced and there are in fact several cases of multiple sites sharing the same site name. ***In general, using the `id` attribute to identify a particular trait association is strongly discouraged and should be used only when necessary.***
  - `species` : Allowed attributes: `id`, `genus`, `species`, `scientificname`, `commonname`, `AcceptedSymbol`. `scientificname` is the preferred attribute for identifying a species. It *should* be unique except in cases where it is left blank. The `scientificname` is mainly left blank only in cases where a species row represents a category of plant; in this case, the `commonname` column is used to describe the category. Even for rows having non-blank `scientificname`, however, uniqueness is not yet enforced.

If a particular cultivar of the species is intended, a child `cultivar` element should be included. This element may use either a `name` attribute (preferred) or an `id` attribute to identify the cultivar. (Cultivar names are guaranteed to be unique within a given species.)

- `citation` : Allowed attributes: `id` , `author` , `year` , `title` , `doi` . The preferred method of selecting a citation is by doi or by author, year, and title (often just author and year will suffice).
- `treatment` : Allowed attributes: `id` , `name` , `control` . The preferred method of selection a method is by name. [In the process of implementation: A citation is *required*, either directly on the trait or as a default for a group of traits, if a treatment is to be specified. Moreover, the specified treatment must be associated with the specified citation. This will often make it possible to use the `name` attribute to specify a treatment, since only treatments associated with the given citation will be considered when selecting by name.]
- `variable` : This specifies what the trait measure. This element must be included if it is not specified in a `defaults` element (see below). Allowed attributes: `id` , `name` , `description` . `name` is the preferred attribute to use to specify the variable and *should* be unique, but this isn't yet enforced and there are a few cases of duplicates.
- `method` : Allowed attributes: `name` . This element *must* have a citation child element. (This citation has no ostensive relation to the citation associated with the trait.) Together, the name and the citation should uniquely determine which method is being used. [To do: Constrain the `methods` table to ensure that this is always possible.]
- `covariates` : This element specifies what covariates are associated with a trait measurement. It allows no attributes but must contain one or more `covariate` child elements. Each `covariate` element must contain a `variable` element (specifying what the covariate measures) and have a `level` attribute (specifying the value of that measurement).
- `entity` : Allowed attributes: `name` and `notes` . An entity with the given value for `name` and `notes` will be created if no entity with the given name exists. [To be implemented: It is an error to specify an entity at the trait level having a blank name. It is an error to supply a `notes` attribute if `name` refers to an existing entity.] [To do: Guarantee uniqueness of non-blank names in the entities table.]

The eight elements just mentioned specify how the trait is associated with data in other tables. In addition, a trait may include two additional elements that further describe the trait:

- `stat` : If a trait describes a group of of measurements (as opposed to a single measurement), a `stat` element may be included. It *must* have the following three attributes:



- `sample_size` , a positive integer.
- `name` , the name of the statistic measured. Possible values are "SD", "SE", "MSE" "95%CI", "LSD", and "MSD".
- `value` , a double giving the value of the named statistic.
- `notes` : This is an element having no attributes but containing free-form textual content.
- Using a single entity for the whole data set.

If all of the traits in the data set should share the same entity, it is possible to specify this by placing an `entity` element as the first child of the root `trait-data-set` element. The element has the same form as an `entity` element contained inside a `trait` element except that this *global* entity is allowed to be anonymous, that is, to have no name or notes attribute. If a global `entity` element is used, it must be the only `entity` element in the document, and no `trait-group` elements may be used in the document (see below).

- Trait groups.

If a group of traits share a number of characteristics, it is possible to nest them within a `trait-group` element. This is mainly useful in the following two cases:

- Some (but not all) of the traits in the file should be associated with the same entity.
- Some (but not all) of the traits in the file share the same metadata (site, citation, treatment, variable, date, species, etcetera).

Multiple level of nesting may be used: `trait-group` elements may themselves contain `trait-group` elements.

- Entities for trait groups.

If a `trait-group` element has no `trait-group` child element, then it may contain, as its first child element, an `entity` element. This usage is similar to the data-set entity usage describe above except that the entity will only be used for the traits in the trait group. If a trait group *does* use an `entity` element, then none of the traits in the trait group can have their own `entity` element.

- Specifying metadata for sets of traits.

If many traits have a common citation, site, species, etcetera, it is possible to avoid repeating this information for each individual trait by using a `defaults` element. A `defaults` element may appear as the child of the `trait-data-set` element (in which

case the defaults apply to all of the traits in the document) or as the child of a `trait-group` element (in which case it applies only to the traits within that group).

`defaults` elements have many of the same attributes and child elements as `trait` elements:

- Allowed attributes are `access_level` , `utc_datetime` , and `local_datetime` .  
`local_datetime` is allowed only if a site having a time zone is specified in the `defaults` element or in a `defaults` element at a higher level and if the specified site is not overridden at a lower level (see below).
- Allowed child elements are `site` , `species` , `citation` , `treatment` , `variable` , and `method` .
- As for the other attributes and elements used with `trait` elements, since the `mean` attribute and the `stat` , `notes` , and `covariates` elements are inherently trait-specific, they cannot be used with the `defaults` element. ( `entity` elements applying to groups of traits are direct children of the `trait-data-set` element or a `trait-group` element rather than being nested within a `defaults` element.)

A default specified by a `defaults` element will apply to all traits occurring within the parent of the `defaults` element unless overridden. A default may be overridden either by another `defaults` element appearing at a lower level or by attributes and child elements of an individual trait.

## Schema for JSON Data Files

[To-do]

## Schema for CSV Data Files

The format to use for CSV trait data files is largely the same as that required for the Bulk Upload wizard explained in the [previous section](#). (See the templates `traits.csv` and `traits_by_doi.csv`.) Some significant differences from the bulk-upload case are:

1. The date of a trait measurement must be given in a column with one of the following headings.
  - i. If the heading `"utc_datetime"` is used, the supplied values must conform to one of the following formats: `1918-11-11T10:00:00Z` or `1918-11-11Z` . In particular, the time must be given in UTC time (hence the "Z"), and if the time is specified (first format), the letter "T" must separate the date and the time portions. If the time is specified,

seconds must be included; optionally, fractional seconds may be included as well.

The resulting `date_loc` value will always be `5` (exact date); the `time_loc` value will be `1` (time to the second) if a time is given and `9` (no data) otherwise.

- ii. If the heading "local\_datetime" is used, the supplied values must conform to one of the following formats: `1918-11-11T11:00:00` or `1918-11-11` . In particular, if the time is specified (first format), the letter "T" must separate the date and the time portions. If the time is specified, seconds must be included; optionally, fractional seconds may be included as well. The resulting `date_loc` value will always be `5` (exact date); the `time_loc` value will be `1` (time to the second) if a time is given and `9` (no data) otherwise.

When "local\_datetime" is used to specify the date-time value of a trait measurement, the date and time are assumed to be local (site) time if a site is given and if that site has a time zone value stored. Otherwise, the value given is assumed to be UTC time. (The date-time value is always stored in the database as UTC time. This paragraph has to do with how the supplied date-time value is interpreted when read.)

Note that only one or the other of these columns may occur in the CSV file. Otherwise an error results.

2. Meta-data can not be specified interactively. Thus any associated citation, site, species, cultivar, or treatment must be specified in each row of the CSV file. (This may later change so that repeated metadata specification may be avoided.)
3. Unlike the bulk upload case, matching of metadata entries *is* case sensitive. Thus, if the CSV file specifies the species as "Sorghum Bicolor" but the database entry for the species specifies the scientificname as "Sorghum bicolor", the upload will not be successful.
4. Unlike the bulk upload case, it is not necessary to specify an associated citation, site, species, or treatment. The sample file `SIMPLE_CSV_TEST_DATA` demonstrates the case where a trait value having no associated metadata is inserted.
5. It *is* required, however, to specify an access level for each trait; therefore, the CSV file *must* have a column named `access_level` .
6. When specifying the citation in a CSV file for use with the Bulk Upload wizard, it is necessary to have either a `citation_doi` column, or have all three of the columns `citation_author` , `citation_year` , and `citation_title` . When using the API, however, any combination of these may be used so long as the values specified in each row

determine a unique citation. For example, if there is only one citation with author "Doe", and if that is the value that occurs in the `citation_author` column of every row of the table, then it is unnecessary to have a `citation_year` or `citation_title` column.

7. Just as for bulk uploads, the `trait_covariate_associations` table is consulted to determine which column names correspond to trait variables and which ones correspond to covariate variables, and further, which covariates correspond to which traits. **But failing to specify a required covariate for one or more traits will not result in an error.** (Thus, in essence, *required* covariates are treated just like *optional* covariates; they will be associated if present, but no complaint will be made if they are not.)
8. No rounding is done of floating point values except to the extent required to fit within PostgreSQL's 8-byte float type. (Note that all floating point values *may* be specified with an exponent; for example 3.20E-2 in place of 0.0320.)

## Error Feedback

As mentioned above, all trait insertion API calls generate an HTTP response. The response will use the same format as the format of the file submitted except in the case of CSV files, where the response is given in JSON format.

In the case of unsuccessful API calls, the response will contain information about the types of errors that caused the call to be unsuccessful. These errors can be classified as follows:

### Authorization Errors

If an invalid API key is given, or if the given key is for a user who isn't authorized to perform the given action, an authorization error is returned. (To do: Distinguish between authentication and authorization.)

### Lookup Errors

If a citation, site, species, or treatment is specified that doesn't match exactly one item in the database, a lookup error occurs. This causes the whole data set to be returned in tree form as `annotated_post_data`. The annotations will be the error items next to the data item that caused the error.

### Schema Validation Errors

As mentioned above, data files in CSV and JSON format are converted to XML format and then validated against an XML schema. For CSV files, since the structure of the XML document generated by the converter is generally correct, this error usually arises only when a data value of the wrong type is given (for example, an alphabetical string where a number is expected). But there are other situations that can trigger a validation error: for example, if a sample size column ( `n` ) is given without including a standard error ( `SE` ) column, or vice versa.<sup>1</sup>

## Model Validation Errors

These errors occur when attempting to save a Trait object to the database and may occur if a variable value is found to be out of range or if a required attribute (e.g. `access_level` ) is missing. As in the lookup error case, this causes the whole data set to be returned in tree form as `annotated_post_data` .

## Running the Examples

There are five sample data files in the directory `app/lib/api/test` .

`SIMPLE_XML_TEST_DATA` : A minimal XML data file consisting of a single trait. It provides only the (currently) required values: the trait variable name, the trait value, and an access level to specify who may view this data item.

`SIMPLE_CSV_TEST_DATA` : A minimal CSV data file consisting of a single trait. It provides only a single variable name and value and the access level.

`TEST_XML_DATA` : A full-fledged XML data file making use of all of the features available for XML trait data insertion: specification of defaults for groups of traits, meta-data lookup, and complete latitude for associating specific groups of covariates and specific sample statistics with specific traits.

`TEST_JSON_DATA` : This JSON data file is an exact analogue to `TEST_XML_DATA` ; it should result in exactly the same trait data being inserted.

`TEST_CSV_DATA` : This CSV data file has five column headings corresponding trait variables and two columns headings corresponding to covariate variable. There is a single data row, so when this file is ingested, a single new entity will be created having 5 associated traits and each trait will have 2 associated covariates. Complete metadata is given for the entity (or equivalently, for the traits it comprises).

## Using curl to Upload the Data Samples

You can upload the data in these files using `curl`. To try this out, start your Rails server locally with `rails s` and then, from the `/api/lib/api/test` directory, run the command

```
curl -X POST --data-binary @TEST_XML_DATA localhost:3000/api/beta/traits.xml?key=<your API key>
```

(Substitute any of the other sample file names for `TEST_XML_DATA` as desired, changing the `.xml` extension to `.json` or `.csv` where appropriate. **If a CSV file is being uploaded, add the option `-H "Content-Type: text/csv"` to the `curl` command.**)

These API calls all generate a response (in XML format for the XML endpoint and in JSON format for the JSON endpoints). If the call is successful, the response will contain a list of the ids of the new traits that were inserted. Note that new entities and possibly new covariates will also be inserted, but the information about these is not (currently) contained in the response.

## Known Bugs

1. It's too easy to make a mistake without realizing it. Examples: a. If you misspell a trait variable name in the heading, that column will simply be ignored; no error will occur if there exists at least one valid trait variable in the heading. b. If you include the same heading twice, the value in one column will overwrite those in the other.
2. Some error messages are obscure and seemingly unrelated to the error that triggered them.

<sup>1</sup> These errors should really be detected during CSV file parsing before attempting to convert to a valid XML file.

# Appendices

# QA/QC with the Web Interface

## \label{sec:qaqc}

Quality assurance and quality control (QA/QC) is a critical step that is used to ensure the validity of data in the database and of the analyses that use these data. When conducting QA/QC, your data access level needs to be elevated to “manager”.

1. Open citation in Mendeley
2. Locate citation in BETYdb
  - Select [Use](#)
  - Select [Show](#)
  - Check that author, year, title, journal, volume, and page information is correct
  - Check that links to URL and PDF are correct, using DOI if available
  - If any information is incorrect, click 'edit' to correct
3. Check that site(s) at bottom of citation record match site(s) in paper
  - Check that latitude and longitude are consistent with manuscript, are in decimals not degrees, and have appropriate level of precision
  - Click on site name to verify any additional information site information that is present
  - Enter any additional site level information that is found
4. Select [treatments](#) from menu bar
  - Check that there is a control treatment
  - Ensure that treatment name and definition are consistent with information in the manuscript
  - Under “treatments from all citations associated with associated sites”, ensure that there is no redundancy (i.e. if another citation uses the same treatments, it should not be listed separately)
  - If managements are listed, make sure that management-treatment associations are correct
5. Check [managements](#) if there are any listed on the treatments page.
  - If yield data has been collected, ensure that required managements have been entered
  - If managements have been entered, ensure that they are associated with the correct treatments
6. Click [Yields](#) or [Traits](#) to check data.
  - Check that means, sample size, and statistics have been entered correctly
  - If data has been transformed, check that transformation was correct in the associated google spreadsheet (or create a new google spreadsheet following



instructions)

- For any trait data that requires a covariate

# Extracting information from figures

## \label{sec:extracting-data}

The easiest program to use for extracting points is [WebPlotDigitizer](#). WebPlotDigitizer is free, browser-based, and cross-platform. Extracts data from images. Demo [here](#).

1. Identify the data that is associated with each treatment *note*: If the experiment has many factors, the paper may not report the mean and statistics for each treatment. Often, the reported data will reflect the results of more than one treatment (for example, if there was no effect of the treatment on the quantity of interest). In some cases it will be possible to obtain the values for each treatment, e.g. if there are  $n-1$  values and  $n$  treatments. If this is not the case, the treatment names and definitions should be changed to indicate the data reflect the results of more than one experimental treatment.
2. Enter the mean value of the trait
3. Enter the `statname` , `stat` , and number of replicates, `n` associated with the mean
  - `stat` is the value of the `statname` (i.e. `statname` might be 'standard deviation' (SD) and the `stat` is the numerical value of the statistic)
  - Always measure size of error bar from the mean to the end of an error bar. This is the value when presented as (  $X \pm SE$  ) or  $X(SE)$  and may be found in a table or on a graph.
  - Sometimes CI and LSD are presented as the entire range from the lower to the upper end of the confidence interval. In this case, take 1/2 of the interval representing the distance from the mean to the upper or lower bound.

### For more information:

- ["Extracting Data From Graphs" \\* related question on Stats.stackexchange](#)

## Estimating Standard Error from other summary statistics (*P*, *LSD*, *MSD*)

When conducting a meta-analysis that includes previously published data, differences between treatments reported with P-values, least significant differences (LSD), and other statistics provide no direct estimate of the variance.

In the context of the statistical meta-analysis models that we use, overestimates of variance are okay, because this effectively reduces the weight of a study in the overall analysis relative to an exact estimate, but provides more information than either excluding the study or excluding any estimate of uncertainty (though there are limits to this assumption).

Where available, direct estimates of variance are preferred, including Standard Error (SE), sample Standard Deviation (SD), or Mean Squared Error (MSE). SE is usually presented in the format of mean ( $\pm$ SE). MSE is usually presented in a table. When extracting SE or SD from a figure, measure from the mean to the upper or lower bound. This is different than confidence intervals and range statistics (described below), for which the entire range is collected.

If MSE, SD, or SE are not provided, it is possible that LSD, MSD, HSD, or CI will be provided. These are range statistics and the most frequently found range statistics include a Confidence Interval (95%CI), Fisher's Least Significant Difference (LSD), Tukey's Honestly Significant Difference (HSD), and Minimum Significant Difference (MSD). Fundamentally, these methods calculate a range that indicates whether two means are different or not, and this range uses different approaches to penalize multiple comparisons. The important point is that these are ranges and that we record the entire range.

Another type of statistic is a "test statistic"; most frequently there will be an F-value that can be useful, but this should not be recorded if MSE is available. Only if there is no other information available should you record the P-value.

## Further Reading

Many statistical transformations are implemented in the `transformstats` function within the PEcAn.utils package. However, these transformations make conservative (variance inflating) assumptions about study-specific experimental design (especially degrees of freedom) that is not captured in the BETYdb schema, for example HSD, LSD, P.

More accurate estimates of SE can be obtained at time of data entry using the formulas in ["Transforming ANOVA and Regression statistics for Meta-analysis"](#).

# Converting Units and Adjustment to Temperature

For many transformations, particularly when automated, please use the `udunits2` software where possible. For example, in R, you can use

```
library(udunits2)
## transform meters to mm
ud.convert(10, "m", "mm")
## equivalently, via the udunits synonym database
ud.convert(10, "meters", "millimeters")
## it can also handle more complex units
ud.convert(10, "m/s", "mm/d")
```

NB: Many of these conversions have been automated within [PEcAn](#).

## Useful conversions for entering site, management, yield, and trait data

`\label{tab:conversions}`

From ((X))	to ((Y))	Conversion	Notes
(X <sub>2</sub> =)root production	(X <sub>1</sub> =)root biomass & root turnover rate	$(Y = X_2/X_1)$	Gill [2000]
DD(^{\circ}) MM'SS	XX.ZZZZ	$(\text{term}\{XX.ZZZZ\} = \text{term}\{XX\} + \text{term}\{MM\}/60 + \text{term}\{SS\}/60)$	to convert latitude or longitude minutes, seconds to decimal
lb	kg	$(Y = X \times 2.2)$	
mm/s	(\mu) mol CO(_2) m(^{2}) s(^{-1})	$(Y = X \times 0.04)$	
m(^2)	ha	$(Y = X/10^6)$	
g/m(^2)	kg/ha	$(Y = X \times 10)$	
US ton/acre	Mg/ha	$(Y = X \times 2.24)$	
m(^3)/ha	cm	$(Y = X/100)$	units used for irrigation a
% roots	root:shoot (q)	$(Y = \frac{X}{1-X})$	$(\% \text{ roots} = \frac{\text{root}}{\text{total biomass}})$

$(\mu)\text{ mol cm}^{-2}\text{ s}^{-1}$	$\text{mmol m}^{-2}\text{ s}^{-1}$	$(Y = X/10)$	
$\text{mol m}^{-2}\text{ s}^{-1}$	$\text{mmol m}^{-2}\text{ s}^{-1}$	$(Y = X/10^6)$	
$\text{mol m}^{-2}\text{ s}^{-1}$	$(\mu)\text{ mol cm}^{-2}\text{ s}^{-1}$	$(Y = X/10^5)$	
$\text{mm s}^{-2}$	$\text{mmol m}^{-3}\text{ s}^{-1}$	$(Y=X/41)$	Korner et al. [1988]
$\text{mg CO}_2\text{ g}^{-1}\text{ h}^{-1}$	$(\mu)\text{ mol kg}^{-1}\text{ s}^{-1}$	$(Y = X \times 6.31)$	used for root_respiration
$(\mu)\text{ mol}$	$\text{mol}$	$(Y= X \times 10^6)$	
julian day (1--365)	date		see ref: <a href="http://disc.gsfc.nasa.gov/">http://disc.gsfc.nasa.gov/</a> (NASA Julian Calendar)
spacing (m)	density (plants $\text{m}^2$ )	$(Y=\frac{1}{\text{row spacing}} \times \text{plant spacing})$	
$\text{kg ha}^{-1}\text{ y}^{-1}$	$\text{Mg ha}^{-1}\text{ y}^{-1}$	$(Y= X/1000)$	
$\text{g m}^{-2}\text{ y}^{-1}$	$\text{Mg ha}^{-1}\text{ y}^{-1}$	$(Y= X/100)$	
kg	mg	$(Y=X \times 10^6)$	
$\text{cm}^2$	$\text{m}^2$	$(Y=X \times 10^4)$	

# Acknowledgements

## Developers

Patrick Mulroony originally implemented the data entry interface, and it is currently maintained by Scott Rohde, with Rob Kooper playing a key role in the development of functionality required to model ecosystems within PEcAn. Andrew Shirk and Carl Crott have contributed (see [visualization on GitHub](#)).

## Contributors

Many data entry technicians (undergrads) have contributed to the implementation and development of the interface and documentation. These include Moein Azimi, David Bettinardi, Nick Brady, Emily Cheng, Anjali Patel, along with other members of the EBI Feedstock Productivity and Ecosystem Services modeling group.

## Funding Sources

- Energy Biosciences Institute
- Department of Energy ARPA-E TERRA
- National Science Foundation