**The Metadata solution implemented in MG-RAST**

*Authors*:
Travis Harrison, teharrison@anl.gov
Jared Bischof, jbischof@mcs.anl.gov
Andreas Wilke, wilke@mcs.anl.gov
Folker Meyer, folker@anl.gov
v: 1.0
date: 3/11/2013

## 1. Introduction

Metadata (data describing data) is in constant flux and while many users struggle to add it to their data, as service providers, we need it to enable users to analyze their data. Think of the ability to "color" an ordination plot using user provided data (e.g. BIOME, sampling location, plot name, replicate name, ..).
Rather than using a **single database with a fixed schema** (that would be too restrictive) or a **complete schema-free approach** (this would be too chaotic and not offer validation), we are using a **hybrid approach**, combining the strength of both.

The hybrid approach enables us to capture arbitrary key value pairs **and** have blessed and controlled subsets of the data ⇒ the best of both worlds. Figure 1 shows the general layout

### *1.1 Scenario points*

- the world for data and metadata consists of files that travel in pairs.
- we believe in minimal checklists driven by the practitioners in the field not heavy handed enforced standards designed by committees
- we represent "truth" as a graph where edges describe some transformations and nodes are project, sample, or data.
- A project may contain many samples, for each sample there may be several data sets. Example: A study of 4 soil samples that each have a 16s amplicon sequence set, a shotgun metagenome, a meta transcriptome and a meta-proteome associated with it.
    - 1 project (including Title, abstract, PI, admin contact, URL, PubMed)
    - 4 samples (GPS coordinates, soil extraction method...)
    - 16 data sets (for sequence data: DNA/mRNA extraction, library creation, sequencing parameters; for non sequence data similar)
- Controlled vocabularies (CV) enable validation (in addition to format or range checks)
    - we use the term CV to mean both controlled vocabularies and some formats (e.g. temperature in Celsius, distances in the metric system, GPS)
- We can implement different levels of metadata stringency for different types / fields.
  - moderately strict: (several required fields with CV terms)
    → enables comparison
    → enables data discovery using CV (e.g. all samples from "marine sediment")

- loose: (required fields, no CV terms)
    → you can compare to some extend
    → very limited data discovery using text search (some samples will not be found)
- very loose (everything is valid)
    → no comparison without human intervention (e.g. meters vs. feet in depth or Fahrenheit vs. Celsius in temp)
    → no data discovery can't find data ⇒ each user defines separate fields, temp vs temperature, or depth in either meters or feet

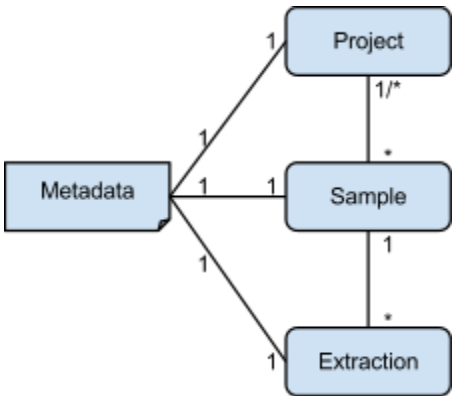



Figure 1: A three tiered hierarchy allows modeling of real life experiments, combining multiple "samples" into a "project" (or study) and allowing for different "extractions" from the sample.

## 2. Workflow

When data is uploaded (or after the fact) a separate metadata file is created. Using a simple spreadsheet (generated by Metazen in most cases) users can capture minimal metadata required for standard compliance (currently GSC MIxS).

Users can submit their metadata spreadsheet for an entire project with many samples and extractions for validation. For later download spreadsheet and XML/GCDML versions can be created.

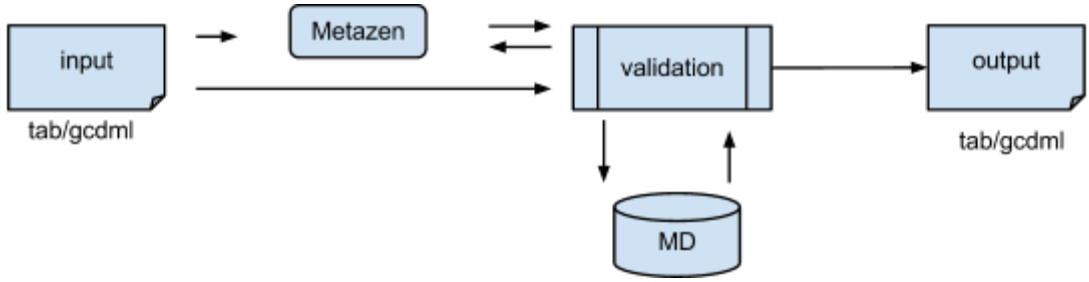



Figure 2: All data files are accompanied by metadata. The tuples of payload file(s) and a metadata file are uploaded together and the metadata file is potentially validated. Separate metadata files describe the data, different “schemata” described Project, sample and data, however the lower tiers can inherit data from the higher tiers i.e. Project and sample.

## 3. Implementation

We use two schemas, a metadata template mechanism, and a validator for controlled vocabulary.
We rely on external controlled vocabularies e.g. BioOntology.org via web-services or downloaded flat files.

### *3.1. Schema 1: Basic data*

1. Generalized metadata storage

This is our “capture all” storage mechanisms, enabling us to capture arbitrary key value pairs. This is important in a field that is currently discovering the need for metadata capturing and the tools for that.

We use a very simple schema: (id | collection | key | value ) to store arbitrary metadata on any data type. By (optionally) organizing metadata into collections we can for each individual collection:

- require certain fields (Keys)
- validate the use of controlled vocabulary terms (when required for certain values) for given fields.

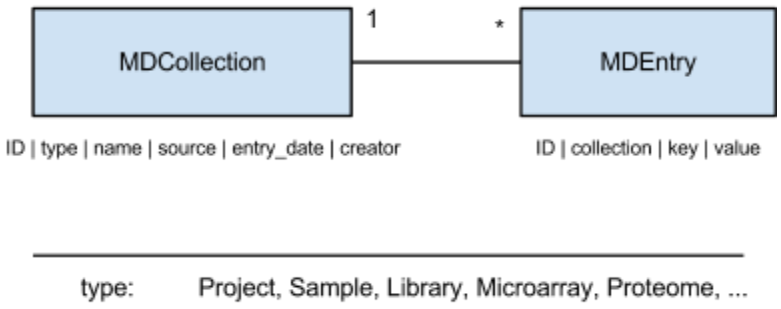


Figure 3: The backend storage mechanism provides two different tables. MDEntry to store all data and MDCollection to group entries into sets.

*3.2 Schema 2:* A meta-metadata database

This database describes the required fields and the controlled vocabularies for each “namespace|domain”. Example domains are GSC MIxS, GSC MIMS.

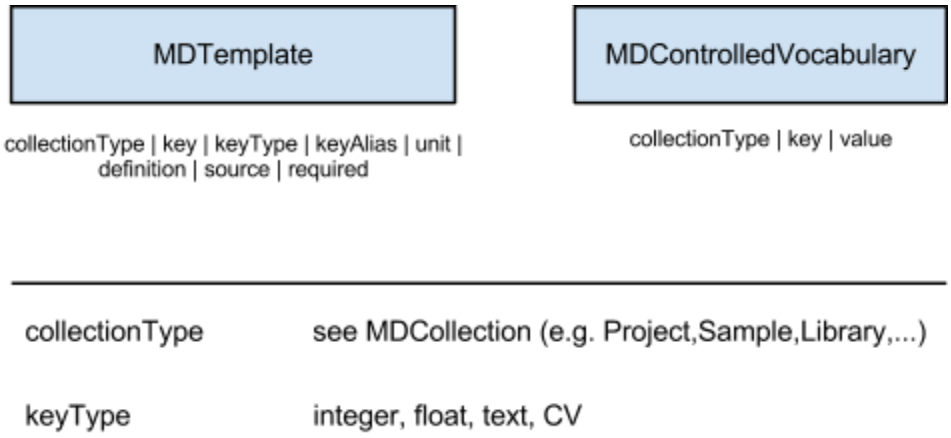



Figure 4: The meta-metadata database provides essentiel two tables describing structure and syntax of certain meta data. For a given metadata collection type the MDTemplate table provides information about metadata categories, value format for the category and source name (e.g. MIAME). MDControlledVocabulary contains valid values for defined keys with type CV in MDTemplate.

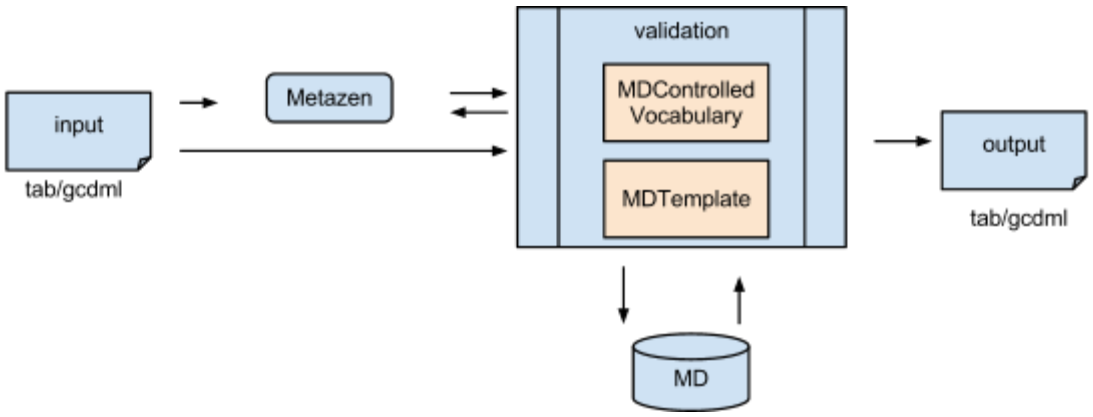


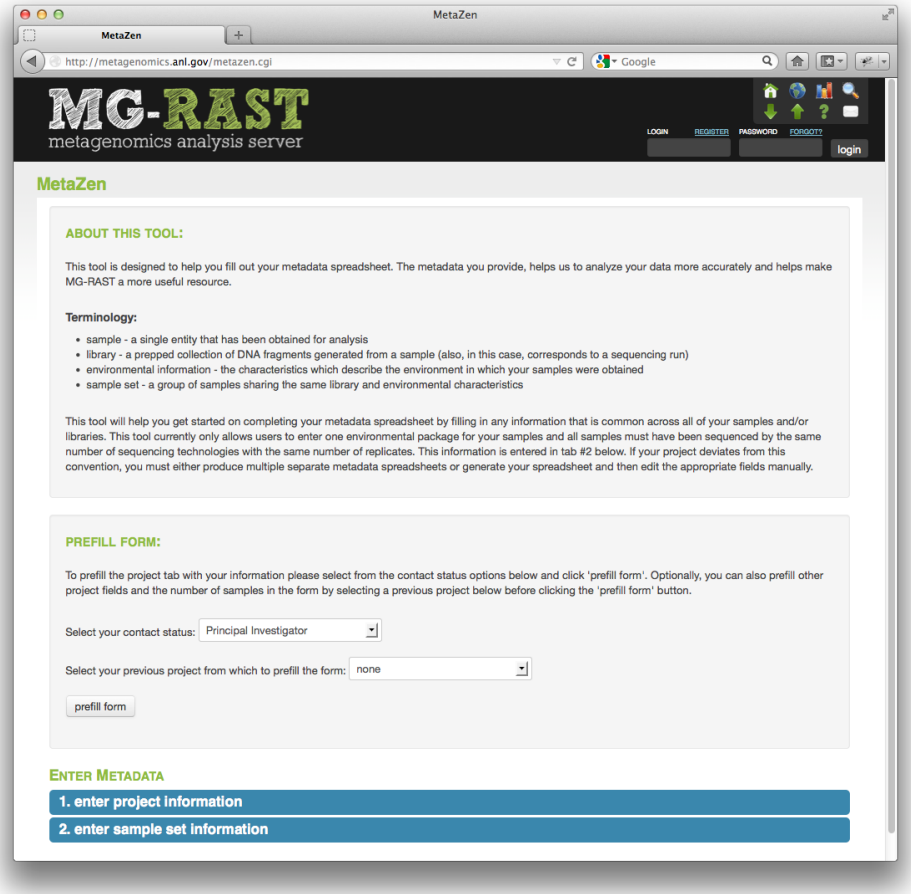

Figure 5: Using MDTemplate and MDCV we enable checking for required keys and controlled

vocabulary terms.

*3.3 Example for a metagenome in cartoon*
sd
*3.4 Example for a metagenome and a 16s data set from the same sample*
sd

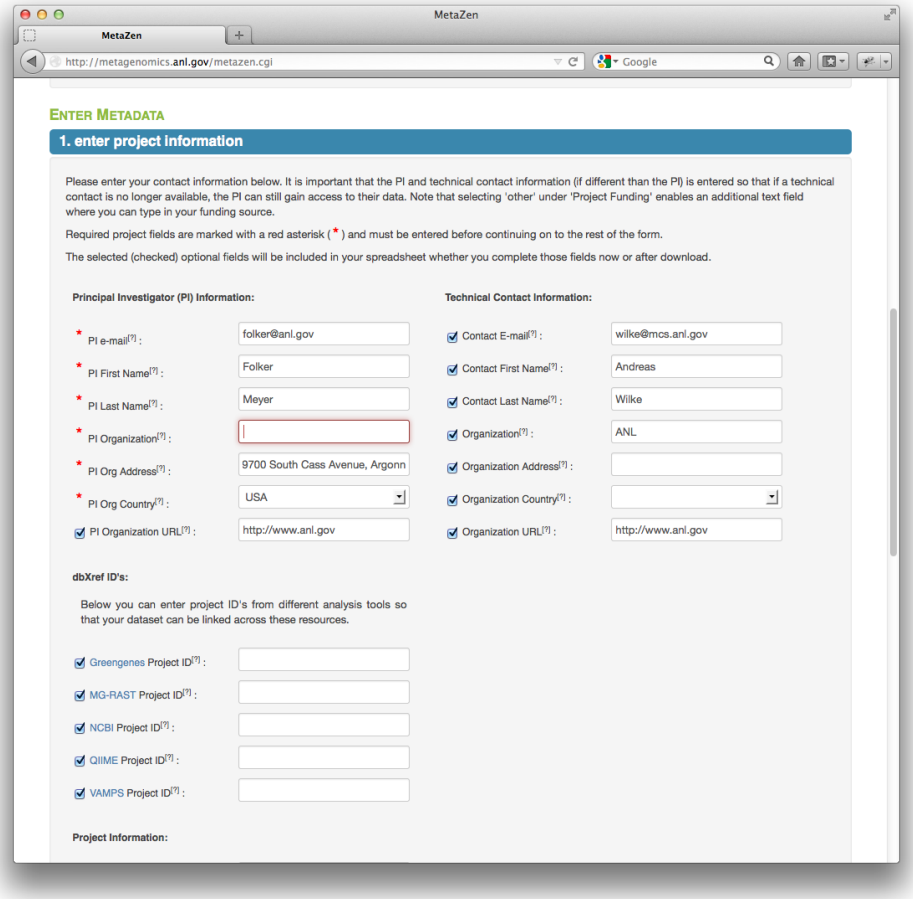
## 4. Metazen a metadata validation tool

*4.1 Recording metadata*
Metazen is an interactive tool that aids users in recording their metadata in a complete and valid format. A defined set of mandatory fields captures vital information while the option to add fields provides flexibility. Project and user level information is stored so users can reload that information without having to enter it repeatedly. The following figure shows what the user sees when they log into Metazen.

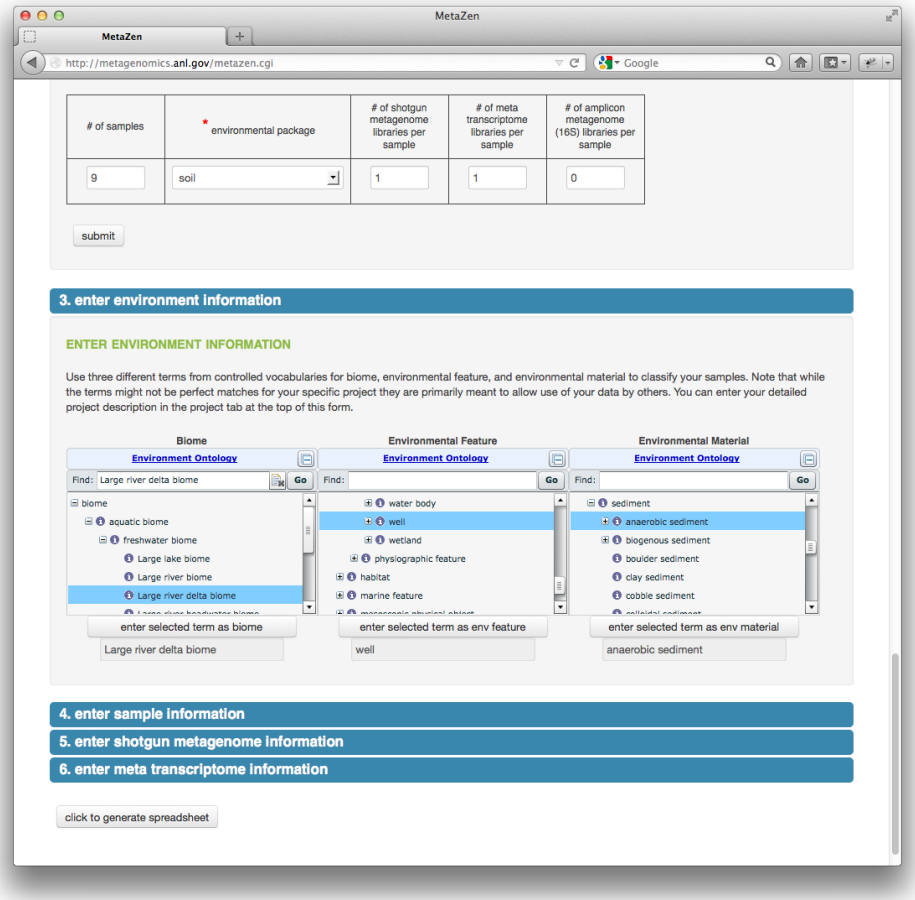



The next figure indicates fields that have been pre-filled by the stored user information, required fields versus optional fields, and fields that are validated (Note: the PI Organization is a required

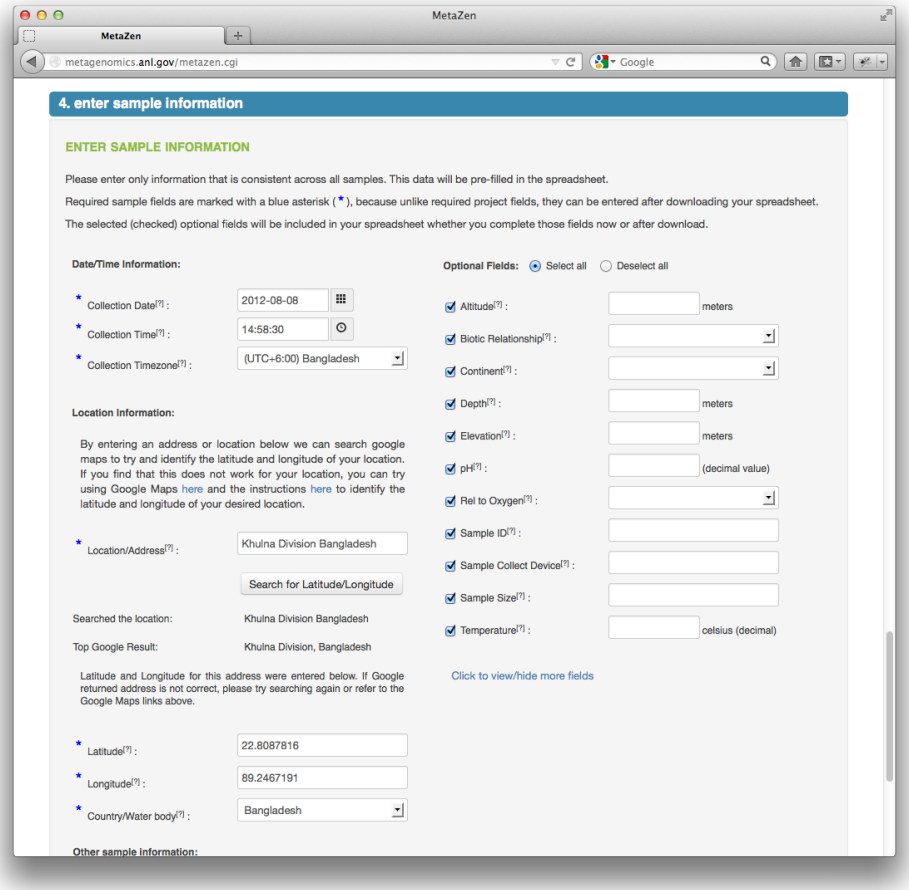
field that was left blank).




Complex, controlled vocabularies are more easily navigated and chosen through Bioportal widgets seen in the following figure.




Users can search for the latitudinal and longitudinal coordinates of where their samples were obtained and enter valid metadata into strictly formatted fields (e.g. date, time, timezone, decimal versus integer values, and other controlled vocabularies). Some of these options are depicted in the next figure.




*4.2 Editing and validating metadata*

Once users are finished entering their metadata into Metazen, a spreadsheet can be generated for download and further editing. Often times users will want to edit their spreadsheet manually if they have many samples and/or sequencing runs for which they would like to enter specific metadata on a per sample or per run basis. Having both the web and spreadsheet to interact with, provides users with guidance to follow standards and flexibility to extend and modify their metadata. Once they are finished editing their metadata, we have a validation tool to help users identify metadata errors, and to ensure that the submitted spreadsheet contains the vital (required) metadata fields and completely valid data.

## 5.0 Extensions to the current state

*5.1 Model for adding new data types*

e.g. GWAS, metaproteomes or microarrays

We believe that adding other data types should be easy, the fields in the spreadsheets might look fundamentally different, but the general approach will translate just fine.

*5.2 Model for addition of new domain specific metadata*

The GSC metadata mechanism is plug-in enabled. Domain scientists can create new required subsets of terms (and the corresponding CV etc). to capture more data on their field of study. Several of these “environmental packages” have been created already and voted into standards by the GSC board.

5.3 Extending existing CVs (locally)

The slowness of the curators of the various CVs leads to user unhappiness in many areas, why are NIH researchers required to list Terrestrial Biome for their Human microbiome samples? Adding a local “ontologies on the fly” extension capability could fix that. This capability basically amounts to adding the “other” field that users can type in IFF their sample/data does not fit into the existing categories.

A number of conditions need to be met in the implementation:

- We’d need to ensure that users only do this after trying to find a good CV term for their data. The default user action here is to select the path of least resistance and select other and type in “blah”. Users are forced to classify their samples and their instinct (as evidenced by hundreds of users) is to provide the most specific classification more akin to a sample name than a sample classification. E.g. “camel” gut vs. “other mammalian” gut.
- Other users should be presented with the new “candidate CV” terms before they can attempt to create their own. This way we can avoid creating too many candidate terms
- We need a procedure for interating on candidate terms with the owners of the CVs. The alternative here is to become CV curators ourselves which we should try to avoid at all cost.
- Validation and export need to recognize the candidate CV terms.